
Shoop Documentation

Release 4.0.0

Shoop Ltd.

2018-07-06

1	Contents	1
1.1	Getting Started with Shoop	1
1.2	Change History	2
1.3	Glossary	18
1.4	Basic Administration Tasks	19
1.5	Prices and Taxes in Shoop	19
2	Developer documentation	21
2.1	Getting Started with Shoop Development	21
2.2	Development Tools	23
2.3	Data model	23
2.4	The Provides system	24
2.5	Addons	27
2.6	Implementation of Prices and Taxes in Shoop	28
2.7	Services API	30
2.8	Notification Framework	31
2.9	Template Design	34
2.10	Shoop API Documentation	37
2.11	Shoop Web APIs	378
3	Indices and tables	381
	Python Module Index	383

Contents

Getting Started with Shoop

Note: If you are planning on developing Shoop, read the *other Getting Started guide* instead.

Installation

Todo

Update this when Shoop is published to PyPI.

This guide assumes familiarity with the PyPA tools for Python packaging, including `pip` and `virtualenv`.

1. Set up a new `virtualenv` for your Shoop project.
2. Grab a Git clone of the Shoop sources. For this guide, we'll assume the checkout of the clone lives in `/stuff/shoop`.
3. Activate the `virtualenv`. Within the `virtualenv`, run

```
pip install /stuff/shoop
```

This will install Shoop and its dependencies into your `virtualenv`.

After this, you can begin setting up a Django project using whichever standards you fancy.

Shoop Packages

Shoop is a constellation of Django apps, with many delivered in the single “Shoop Base” distribution, and with additional apps available as separate downloads.

The core package all Shoop installations will require is `shoop.core`. It contains the core business logic for e-commerce, and all of the database models required. However, it contains no frontend or admin dashboard, as different projects may wish to replace them with other components or even elide them altogether.

A default frontend, a basic but fully featured storefront, is included, as the application `shoop.front`. It itself has several sub-applications that may be used to toggle functionality on and off.

Todo

Describe the sub-apps.

A fully featured administration dashboard is also included as the application `shoop.admin`.

Change History

Shoop Release Notes

Shoop 4.0.0 Release Notes

Released on 2016-06-29 21:30 +0300.

This release includes many new features, bug fixes and improvements in over 400 non-merge commits since 3.0.0. Highlights of the changes are listed here, but for complete list of changes see [Shoop Change Log](#) or [Git commit log](#).

This is also the last release of Shoop, since the product name will change to Shuup. The same code base with “shoop” replaced by “shuup” will be released as Shuup 0.4 shortly after Shoop 4.0 release.

There’s also update instructions for [updating from 3.0](#).

New Features

- Campaigns
 - Campaigns and coupons can now be managed in Admin
- Order printouts
 - Allows creating PDF printouts of orders from Admin
 - Requires WeasyPrint to be installed (available via pip)
- Stock management
 - Stock counts can now be managed with Simple Supplier module
- Payments and refunds
 - Payments and refunds for orders may now be created in Admin
- Order editing
 - Non-delivered orders without payments may now be edited in Admin
- Package products
 - Package products can now be created in Admin
- Shipping and payment behavior components
 - Shipping and payment methods management is reformed to allow versatile extension with addons. One of the changes introduces a new concept called service behavior components. These components allow merchants to customize behavior of the payment and shipping methods in more detail than before.
- Price display options

- It is now possible to define how prices are rendered for a customer by a contact group setting. Choices are: show prices including taxes, excluding taxes or hide prices.
- Guide
 - Guide app integrates search results from Shoop Guide documentation into Admin search

Miscellaneous

- Brazilian Portuguese translations (pt_BR)
 - Thanks to Christian and Jonathan Hess
- Initialize checkout addresses from customer data
 - Thanks to Jason Sujjon

Updating from 3.0

Shoop 4.0 introduces many API changes which could affect projects or addons based on Shoop 3.0. Here is a list of the API changes and instructions how to update your code.

- Database changes

Database migration from 3.0.0 should go smoothly with MySQL and SQLite databases. Unfortunately we don't test with PostgreSQL currently and found out just before the release that our migrations don't work for PostgreSQL. Setting `connection.features.can_rollback_ddl` to `False` (e.g. in your `manage.py`) may help in that case.
- Payment and shipping service API is refactored
 - This affects payment and shipping method addons.
 - See [Services API](#) document for introduction to the new API.
 - See this `pull request` as an example how to refactor the addon code.
- Simple Pricing is renamed to Customer Group Pricing.
 - Prices have to be migrated manually.
- Enum value `OrderLineType.CAMPAIGN` is renamed to `DISCOUNT`
 - Replace usages of `CAMPAIGN` line types with `DISCOUNT`
- Discount Pricing is removed.
 - Use the new Campaigns features instead.
- `order_creator_finished` signal is moved from Front to Core
 - Update your imports.

Shoop 3.0.0 Release Notes

Released on 2016-01-21 11:15 +0200.

Here's a few highlights of new features, improvements and fixes in Shoop 3.0.0 since 2.0.0. For complete list of changes see [Shoop Change Log](#) or [Git commit log](#).

There's also update instructions for [updating from 2.0](#).

New Features

- Creating orders from the Admin
- Localization
 - Provide tools for collecting and compiling translated messages.
 - Add language chooser to Classic Gray theme.
 - Implement JavaScript translations.
 - Xtheme plugins are now translatable.
 - More messages have been marked for translation.
 - Included translations: English, Finnish, Chinese and Japanese.

Improvements

- Allow addons to inject resources to Xtheme templates
- Enable email login and password recovery with username
- Show product media at order history and product detail pages
- Show Shoop version number in Admin
- Xtheme editor improvements
- Tax system improvements
 - Document the tax system (see *Prices and Taxes in Shoop*)
 - Implement override groups for Default Tax
 - Clean-up internal tax/price related APIs
- And many more

Fixes

- Xtheme: Don't crash when trying to revert unsaved configuration
- Default Tax: Fix calculation of added and compounded taxes
- And many more

Miscellaneous

- Test are now ran also on [Travis](#)
- Python package dependencies are updated and not so strict anymore
- Node package dependencies are locked down with npm-shrinkwrap

Updating from 2.0

Shoop 3.0 introduces some API changes which could affect projects or addons based on Shoop 2.0. Here is a list of the API changes and instructions how to update your code.

- `Priceful.total_price` has been removed. This affects e.g. `OrderLine`, `BasketLine` and `PriceInfo` objects.
 - Use `Priceful.price` instead.
- `PriceTaxContext` is removed.
 - You should not need it. Use `PricingContext` or `TaxingContext` where appropriate.
- Default Theme is removed.
 - It is available as a separate package from <https://github.com/shoopio/shoop-simple-theme>
- Address is split to `MutableAddress` and `ImmutableAddress`. Address is converted to abstract base class.
 - Database changes should be handled by migrations.
 - Usually usages of Address should be converted to `MutableAddress`, but they should be converted to immutable with `Address.to_immutable` for e.g. Order addresses.
- Submodules of `shoop.core.pricing` are now private.
 - Use the API exposed by the `shoop.core.pricing` module's `__init__.py`. E.g. use `shoop.core.pricing.PriceInfo` instead of `shoop.core.pricing.price_info.PriceInfo`.
- Submodules of `shoop.core.order_creator` are now private.
 - Use the API exposed by the module's `__init__.py`.
- `shoop.core.utils.reference` is removed.
- Some submodules of `shoop.xtheme` are now private.
 - Use symbols directly from `shoop.xtheme`, e.g. `shoop.xtheme.Theme` instead of `shoop.xtheme.theme.Theme` or `shoop.xtheme.TemplatedPlugin` instead of `shoop.xtheme.plugins.TemplatedPlugin`.
- Submodules of `shoop.core.models` are now private.
 - Use the models or enums directly from the main package.
- `shoop.core.models.product_variation` is removed.
 - Relevant functions are now available as `Product` methods.

Shoop 2.0.0 Release Notes

Released on 2015-10-05 16:45 +0300.

The highlights of this release (over 500 commits since 1.2!) are:

- Revamp of the pricing and taxation systems for flexible international commerce
- A new pluggable frontend theming system called Xtheme
- Usability improvements for the admin, including much better media management
- A brand new, slick frontend theme “Classic Gray”

- Lots and lots of other fixes and improvements!

For more detailed list of changes see [Shoop Change Log](#) or [Git commit log](#).

Shoop Change Log

Version 4.0.0

Released on 2016-06-29 21:30 +0300.

Core

- Fix bug: Category model is missing MPTT ordering options
- Add staff only behavior component
- Add refund-related methods to `Order`
- Add `OrderLineType.REFUND` enum type
- Update order phone max length to 64 characters
- Add cash payment method
- Add rounding behavior component
- Add contact group availability behavior component
- Fix bug: Do not display decimal values in scientific notation
- Fix bug: Taxes of child order lines are filled incorrectly
- Fix bug: Order line parent lines are not linked
- Add modified on for `Order`
- Add modified by for `Order` and `OrderSource`
- Add `get_company_contact` to `shoop.core.models`
- Implement taxing of lines without tax class
- Add new abstract method `get_taxed_price` to `TaxModule`
- Add `ShopProduct.is_visible`
- Add `Order.can_edit`
- Update shipping status correctly in `Order.create_shipment`
- Add `OrderModifier` for modifying orders with `OrderSource`
- Add option use `Order.create_shipment` with unsaved `Shipment`
- Add identifier for `Shipment`
- Fix bug: Fix max length for service method names for `Order`
- Make `Order.create_payment` also do status update for partially paid
- Add `get_total_unpaid_amount` method to `Order`
- Add `is_canceled` and `can_set_canceled` to `Order`
- Fix bug: Convert `Shipment` weight to kilograms

- Make `create_shipment` for order atomic
- Add `shipment_created` signal
- Add `get_tracking_codes` to `shoop.core.models.Order`
- Add weight based pricing behavior component
- Add `total_gross_weight` property for `Source`
- Fix bug: Order line text is not set for package products
- Add new Service API and implement shipping and payments with it
- Remove `BaseMethodModule` based API (`shoop.core.methods`)
- Add support for custom user model (`AUTH_USER_MODEL`)
- Add management command to generate bought with relations
- Add `merchant_notes` text-field to `Contact`
- Add `first_name` and `last_name` fields to `PersonContact`
- Add bought with relation to `ProductCrossSellType`
- Set customer marketing permission while creating order
- Disable delete for default contact groups
- Allow storing price display options to contact groups
- Add template tags for rendering prices with context specific price display options (pretax or with taxes, or hide prices totally)
- Fix bug: `OrderSource.tax_amount` always returned zero price
- Add contacts automatically to type specific default groups
- `OrderCreator` no longer requires a request
- Add `order_creator_finished` signal under `order_creator`
- Move `calculate_taxes_automatically` from `OrderSource` to `TaxModule`
- Add shop product validation for `OrderSource`
- Add option to create payments with REST API
- Add contact address for `Shop`
- Add `update_stock` calls for `SimpleSupplierModule`
- Rename `CAMPAIGN` in `OrderLineType` enum to `DISCOUNT`
- Add `OrderSourceModifierModule` interface for modifying order source
- Add `DiscountModule` interface for pricing
- Simplify `PricingContext` and require shop and customer for it
- Add `get_price_info` etc. functions to `shoop.core.pricing`
- Add `get_suppliable_products` to `shoop.core.models.Supplier`
- Add “codes” API to `OrderSource` and `BaseBasket`

Localization

- Add Brazilian Portuguese translations (pt_BR)

Admin

- Add refund creator to order admin
- Add customer detail view to order creator
- Define module-level permissions for all admin modules
- Enable adding of permission groups from users admin
- Add admin module for managing Django permission groups
- Add `get_required_permission` to `AdminModule`
- Add support for module-level permissions
- Use `Select2MultipleField` for handling contact members
- Add `Select2MultipleField`
- Add account manager for contact
- Add `PersonContact` choice widget
- Add barcode filter to product list view
- Show stock in order creator
- Fix bug: Fix decimal precision issues on order creation
- Enable order creation for contact from contact detail view
- Fix bug: Refresh order lines when customer is changed during order creation
- Add default `is_active` filter to `Contact` and `User` admins
- Enable default values for `ChoiceFilter`
- Enable contact activation/deactivation
- Allow contact adding and removing for company
- Show companies in person contact edit page
- Add target option to `SearchResult`
- Filter contacts with group in contact list view
- Hide group members from contact group edit view
- Remove support for select pickers
- Enable multiselect dropdown with `Select2`
- Enable adding log entries to orders
- Enable order editing
- Allow shipment creation form extensions
- Add payment creation view to `Order` admin
- Enable order cancelation in `Order` edit view

- Hide invalid choices for package products
- Fix bug: Fix convert to parent menu items in `EditProductToolbar`
- Show tracking codes in order detail
- Fix bug: Show package siblings for variation children
- Fix bug: Detail page of contact with multiple groups fails on Python 3
- Enable add/edit for weight based behavior component through service admin
- Add `admin_contact_group_form_part` provider for `ContactGroup` admin
- Redo shipping and payment method management
- Add service provider management
- Add package mode for products
- Enable merchant notes editing for contacts
- Add option to add extra form parts to Shop edit view
- Enable delete for contact groups
- Make all enabled shipping and payment methods available in order creator
- Check product quantities in order creation
- Add option to add action buttons to Order edit view

Addons

- Enable upgrade, migrations and collectstatic from admin
 - Has a known problem in reloading application server.

Front

- Initialize checkout addresses from customer data
- Logout users linked to inactive contact
- Allow user to change password from account settings
- Add login check for inactive contacts
- Remove “Ordering for company” from checkout if logged in
- Allow user to link company contact from account page
- Log-in as company if user is a member of a company
- Make `get_product_cross_sells` faster
- Make `get_best_selling_products` faster
- Make `get_visible_products` faster
- Fix bug with multiple service checkout phases
- Update UI for package products
- Add `SHOOP_FRONT_ADDRESS_FIELD_PROPERTIES` setting
- Support also `django-registration-redux` 1.4

- Enable description and logo for methods in checkout
- Add admin view for monitoring customer carts
- Remove `get_method_validation_errors` signal
- Fix bug at `get_visible_products` filter when `orderable_only` is False
- Set template price display options from the customer
- Fix bug: `BasketStorage.finalize()` never called `delete()` correctly
- Check product quantity already in basket while adding
- Move `order_creator_finished` signal under core
- Add “next” parameter support for registration
- Process given coupon codes in basket
- Add `get_visible_products` template helper

Xtheme

- Add custom `XThemeModelChoiceField` to show admin URLs
- Enhance default text plugin editor to remarkable markdown editor
- Add support for global/multi-view placeholders
- Add generic snippets plugin for doing simple integrations
- Add a plugin for displaying category links on shop front
- Add a linkable image plugin

Classic Gray Theme

- Add Shoop Wishlist addon support for logged in users
- Hide product order section when prices are hidden
- Hide cart when prices are hidden
- Show tracking codes in order detail
- Remove `ProductCrossSellType.COMPUTED` from cross-sells plugin
- Update cross-sells plugin to use `ProductCrossSellType.BUGHT_WITH`
- Render prices with the new price rendering template tags
- Show error messages while adding products to basket
- Add “next” parameter to register links
- Add Coupon use possibility to basket page
- Add option to only show orderable products to highlights plugin
- Add Xtheme plugin to display social media links on shop front

Simple Supplier

- Add admin modules for updating stock
- Add support for stock counts and values

Order Printouts

- Add app for creating PDF printouts of orders
 - Currently supports printing of Order confirmation and Delivery slips

Campaigns

- Add campaigns app with following features:
 - Campaigns management
 - Coupon management
 - Contact group sales ranges

Customer Group Pricing

- Rename Simple Pricing to Customer Group Pricing
- Fix pricing for `AnonymousContact`

Discount Pricing

- Removed

Simple CMS

- Add option to list children on page
- Add possibility to set parent on page

Default Tax

- Filter tax rules by postal codes to gain better performance
- Add minimum and maximum postal code values to `TaxRule`

Guide

- Add guide app that integrates help documentation into admin search

General/miscellaneous

- Rename `UnitedDecimal` to `UnittedDecimal`
- Add a way to find out min and max values from pattern
- Reword `doc/provides.rst`

Version 3.0.0

Released on 2016-01-21 11:15 +0200.

Core

- Document Shoop tax system
- API and documentation clean-up
- Remove `AddressManager`
- Split address into mutable and immutable address
- Add `Product.get_public_media` helper function
- Ensure `TranslatableShoopModel.__str__` doesn't return lazy objects
- Deny price unit changes of in-use Shop
- Assign created customers to `CustomerTaxGroups` on creation
- Fix couple tax related terms
- Remove `PriceTaxContext`
- Add `PricingContextable` interface and fix related docstrings
- Remove `Priceful.total_price`
- Add dynamic configuration API: `shoop.configuration`
- Fix tax calculations and implement override groups
- Add autoexpiring versioned cache API: `shoop.core.cache`
- Enable email login and password recovery with username

Localization

- Add translations for Finnish, Chinese and Japanese
- Add translation extraction tools (`shoop_makemessages`)
- Mark more messages for translation in templates
- Enable JavaScript translations

Admin

- Bump bootstrap-datetimepicker version to 2.3.8
- Show Shoop version number in Admin
- Fix order list sorting and filtering by total price
- Fix CMS page list sorting by title
- JavaScript compilation fixes: Turn our ES6 to ES5 with Babel
- Fix URL encoding in `redirect_to_login`
- Add view for creating orders from the Admin
- Enable markdown editor for product and category description
- Fix SKU and name initialization when creating a product by search
- Add new template macros
- Refactor templates to use template macros
- Admin form styling and UX updates

Front

- Set default country in checkout address forms
- Fix `SHOOP_FRONT_INSTALL_ERROR_HANDLERS` setting being not respected
- Change password recover error message

Xtheme

- Make Xtheme plugins translatable
- Allow addons to inject resources
- Editor improvements
- Fix a crash when trying to revert unsaved configuration

Classic Gray Theme

- Basket: Hide line base price when it's not positive
- Show product media at order history and product detail pages
- Add language changer to navigation
- Add possibility for other future brand colors
- Add carousel styles for Bootstrap carousel
- Unvendor fonts
- Show maintenance mode for super user
- Fix logo text line height
- Add new placeholders

- Footer CMS Pages field are no longer required
- Update label for footer links to avoid confusion

Default Theme

- Remove Default theme from Shoop Base. Moved to <https://github.com/shoopio/shoop-simple-theme>

Campaigns

- Fix admin list view sorting

General/miscellaneous

- Add Transifex configuration for the `tx` command
- Add verbose names to all model and form fields
- Do unit testing from doctests too
- Update Python package dependencies
- Lock down JavaScript dependencies
- Code style improvements
- Add configuration for Travis CI
- Include JS and CSS source maps to the Python package
- Cleanup tax TODOs
- Move `apply_request_middleware` to testing
- Documentation: Tune Sphinx settings and ignore migrations in API docs
- Fix ESLint complaints
- Make sure that bower is ran non-interactively

Version 2.0.0

Released on 2015-10-05 16:45 +0300.

Admin

- Add basic Manufacturer views
- Basic Supplier management
- Add image field for category
- Fix issue with price not being saved
- Optionally disable creating shops
- Add “Product Media” tab to product editing
- Tabify translated fields

- Variation UI styles
- Media Browser Rehaul
- Add `ImageChoiceWidget`
- Actually send `data-filter` to media browser from browse widgets
- Admin form error indicators
- Add keyboard shortcuts to megasearch
- Show first language tab with errors
- Update styles for admin form error indicators
- Show errors more clearly
- Make folder clicks work at media browser
- Fix paths of generated source maps
- Add styles for bootstrap input-group
- Time interval attribute now renders as `DecimalField`
- Notify: Add better error handling for “step edit”-popup
- Better variation error handling
- Fix issue where visibility errors caused an error
- Update admin category view
- Admin datetimepicker

Core

- Rework `SimplePricing` and pricing in general
- Add non-ASCII support for supplier name
- Fix checking of duplicate settings
- Add new fields to shop core
- Add `DiscountPricingModule`
- Fix `PriceInfo` usage with non-one quantities
- Fix `shoop.core.migrations.0006`
- Tax clean-up and refactoring
- Prevent `Shop` being deleted when image was deleted

Front

- An all-new dynamic theming system, `Xtheme`
- Classic Gray: A new slick theme built on the `Xtheme` system
- Add ordering for cross sells template helper
- Fix `get_root_categories` performance
- Maintenance mode

- `template_helpers`: Fix `get_pagination_variables`
- Ensure user is logged in after activating account
- Customer URL now requires login
- Add support for Complex variations
- Add Default ErrorHandling
- Fix issue with variation children being listed for admin user
- Front: Fix issue with variation children visible in search results

General/miscellaneous

- Run ESLint on all the things!
- Prunes, manifests
- PEP8ify
- Various fixes
- Tests: Make `test_user_detail_contact_seed` not fail randomly
- Miscellaneous tiny fixes
- Fixes
- Cms duplicate
- Embetter patterns
- Saner sanity tools
- Workbench: Allow overriding couple settings from env

Version 1.2.0

Released on 2015-08-24 17:30 +0300.

- Admin: Polyfill forms to ensure IE support
- Fix uniqueness of some `InternalIdentifierFields`
 - Namely identifier field of `Attribute`, `OrderStatus`, `ProductVariationVariable` and `ProductVariationVariable-Value`
- Admin: Show payment details in order views
- Coding Style: Clean-up and sort all imports
- Fix usages of too-direct imports of models
- Fix some unicode/bytes issues by adding “`unicode_literals`” imports
- Admin layout fixes
 - Update telemetry admin layout and add translations tags
 - Change the attributes icon from product edit to the right one
 - Move attributes in product type edit to it’s own tab
 - Hide browser native horizontal scrollbar from main menu

- Admin: Product image management
- Admin: Product Variation management
- Front: Add cross-sells to product detail page in default template
- Admin: Fix menu scrolling
- Upgrade Python and npm dependencies
- Admin: Shop management
- Front: Add link to admin panel in default template
- Admin: Fix product attributes getting cleared unless they were edited
- Admin: Product Sales Unit management
- Admin: Add ProductChoiceWidget for selecting Products
- Admin: Product cross-sell management
- Admin: Styling: Add borders to bootstrap select
- Admin: Fix showing details of a CompanyContact
- Admin: Fix showing current addresses in contact details

Version 1.1.0

Released on 2015-07-03 12:30 +0300.

- Improve “Getting Started with Shoop Development” documentation
- Add a basic REST API for reading/writing products and reading orders
- Use the database to store shopping baskets by default
- Implement pluggable shopping basket storage backends
- Implement basic contact group admin
- Add telemetry (usage statistics) system
- Add Dockerfile
- Improve admin login flow
- Document settings; make documentation builds available on ReadTheDocs
- Make release packaging much more robust
- Generate order keys in a secure manner
- Trim admin search strings
- Embetter admin order layouts
- Create the Shop as active with `shoop_init` management command
- Fix usages of `Category.get_ancestors()` in templates
- Remove Stripe integration (`shoop.stripe`)
 - It now lives in <https://github.com/shoopio/shoop-stripe>
- Core: Declare correct `required_installed_apps` in `AppConfig`
- Fix handling of tuple-format `required_installed_apps`

- Fix Money class to not read settings at instance creation
- Fix management command `shoop_show_settings` for Python 3
- Add Addon documentation (`doc/addons.rst`)

Version 1.0.0

Released on 2015-06-04 16:30 +0300.

- The first Open Source version of Shoop.

Glossary

Software Components

Shoop Shoop is a framework for building web shops or ordering portals.

Shoop Core Shoop Core is the kernel of the Shoop. It is part of every Shoop installation.

Shoop Base Shoop Base contains Shoop Core and optional *modules*.

Module

Todo

Define module

Todo

Define all plugin terms

Entities

user user is general definition of any user

admin admin is one kind of user

customer Customer is one kind of user or one kind of company. Customer can be an admin and vice versa.

company Company is one kind of organization. Organization is general definition of any group of companies and/or users.

shop

Todo

Define shop

Basic Administration Tasks

Todo

Revise this when user management lands in the admin.

You can use the built-in Django management commands `createsuperuser` and `changepassword` to manage superusers and change user passwords.

See also [Creating superusers](#) from Django’s documentation.

Prices and Taxes in Shoop

This document gives an overview of Shoop’s pricing and tax mechanics. For deeper view about the implementation – for example, if you’re implementing a price/tax related addon – also read the [Implementation of Prices and Taxes in Shoop](#) document.

Price Unit

Prices in Shoop have an unit that is combination of a currency and an includes/excludes taxes flag. That is, prices may be specified pretax or with taxes included. Which taxation type and currency is used is usually decided by the [Shop](#), which has `currency` and `prices_include_tax` fields. In general, it is also possible that the active [PricingModule](#) uses a different price unit that is specified by the shop. Currently there is no such pricing module in the Shoop Base distribution.

Different price units cannot be mixed: Adding a pretax price and a price including taxes together would be an error, as would be adding USDs to EURs.

The price unit of a [Shop](#) can be changed as long as there are no [Orders](#) created for the shop.

The price unit of an [Order](#) is stored in its `currency` and `prices_include_tax` fields. The line prices of an order are stored in that unit, but the total price of order is stored with and without taxes in the `taxful_price` and `taxless_price` fields.

Calculation of Taxes

How Taxes Are Determined

Taxes in Shoop are implemented by a [tax module](#). The Shoop Base distribution ships a tax module called [Default Tax](#), but it is possible to plug in another tax module via [addons](#) or to implement a new one.

The responsibilities of a tax module are to calculate taxes for an order or for separate items (e.g. product, shipping or some other taxable item). The most important function of a tax module is to take an order source (such as a basket), which has lines (with pre-tax prices or prices with tax included) and fill in the taxes for each line in the source.

When Taxes Are Determined

There are two modes of operation for calculating the taxes: on-demand and on-checkout. If current tax module declares that tax calculation is “cheap” (does not cost a transaction fee and is fast to compute) and `SHOOP_CALCULATE_TAXES_AUTOMATICALLY_IF_POSSIBLE` setting is true, then on-demand calculation will be used. Otherwise on-checkout calculation will be used.

With on-checkout tax calculation mode Shoop calculates taxes for a basket in the confirmation phase of the checkout process or in the confirmation phase of the order creating UI in the Shop Admin. This means that taxes are not known for items in the basket, product listings or on the detail page of a product. The reason for not calculating taxes before the confirm phase is that the active tax module might query tax information from an external source which might be prohibitively slow or cost a transaction fee.

With on-demand tax calculation mode calculation happens when showing prices in the shop, if prices returned by the current pricing module are pretax and current price display options demand prices including taxes or the other way around.

Taxes in Orders

Taxes are stored in order lines. Each order line can have several taxes applied and each of them is stored to a separate *line tax* object linked to the order line. These line tax objects contain references to *Tax* objects, the name of the tax, the applied amount and the base amount the tax is calculated off of.

The Default Tax Module

Shoop Default Tax is a tax module that calculates taxes based on a set of static rules stored in the database. A tax rule applies a tax for an order line or any other taxable item (e.g. product or shipping method). An item can be taxed with several taxes, which will be either added together or compounded over each other.

Defining Tax Rules for The Default Tax Module

The tax rules of Default Tax can be managed in the Shoop Shop Admin (*Menu* → *Taxes* → *Tax Rules*).

Most fields of the tax rule determine the conditions when the rule applies. All non-empty fields must match for the rule to apply. Empty fields are not considered, e.g. if the “Customer tax groups” field is left empty, all customer tax groups will match. You may use these conditions to apply tax rules e.g. only for a specific country or area.

Area specific matching criteria fields are specified with a pattern that is able to match multiple values. See the help text in the admin view for details on how to write those patterns.

If all conditions of a tax rule match, the rule will be applied. That means that the tax specified in the rule will be added for the item. If there are several rules to be applied for an item, the total tax is determined by the priority field. Rules with same priorities are calculated as added (which would be the case for taxes in the United States) while rules with different priorities define compounding taxes (for example the PST taxes in Canada’s Quebec province).

Tax rules may also define override group numbers. If several rules match, only the rules with the highest override group number will be effective. This can be used, for example, to implement tax exemption by adding a rule with very high override group number that sets a zero tax.

Developer documentation

Getting Started with Shoop Development

Note: If you are planning on using Shoop for developing your own shop, read the *other Getting Started guide* instead.

Installation for Shoop Development

To start developing Shoop, you'll need a Git checkout of Shoop and a Github fork of Shoop for creating pull requests. Github pull requests are used to get your changes into Shoop Base.

1. If you haven't done so already, create a fork of Shoop in Github by clicking the “Fork” button at <https://github.com/shoopio/shoop> and clone the fork to your computer as usual. See [Github Help about forking repos](#) for details.
2. Setup a virtualenv and activate it. You may use the traditional `virtualenv` command, or the newer `python -m venv` if you're using Python 3. See [Virtualenv User Guide](#), if you don't know virtualenv already. For example, following commands create and activate a virtualenv in Linux:

```
virtualenv shoop-venv
. shoop-venv/bin/activate
```

3. Finally, you'll need to install Shoop in the activated virtualenv in development mode. To do that, run the following commands in the root of the checkout (within the activated virtualenv):

```
pip install -e .
python setup.py build_resources
```

Workbench, the built-in test project

The Workbench project in the repository is a self-contained Django project set up to use an SQLite database. It is used by the test suite and is also useful for development on its own.

Practically the only difference to a normal Django project is that instead of `python manage.py`, one uses `python -m shoop_workbench`.

To get started with Workbench, invoke the following in the Shoop working copy root.

```
# Migrate database.
python -m shoop_workbench migrate

# Import some basic data.
python -m shoop_workbench shoop_populate_mock --with-superuser=admin

# Run the Django development server (on port 8000 by default).
python -m shoop_workbench runserver
```

You can use the credentials `admin/admin`, that is username `admin` and password `admin` to log in as a superuser on `http://127.0.0.1:8000/`.

Building resources

Shoop uses JavaScript and CSS resources that are compiled using various Node.js packages. These resources are compiled automatically by `setup.py` when installing Shoop with `pip`, but if you make changes to the source files (e.g. under `shoop/admin/static_src`), the resources have to be rebuilt.

This can be done with

```
python setup.py build_resources
```

The command also accepts couple arguments, see its help for more details:

```
python setup.py build_resources --help
```

Running tests

To run tests in the active virtualenv:

```
py.test -v shoop_tests
# Or with coverage
py.test -vvv --cov shoop --cov-report html shoop_tests
```

To run tests for all supported Python versions run:

```
pip install tox # To install tox, needed just once
tox
```

Collecting translatable messages

To update the PO catalog files which contain translatable (and translated) messages, issue `shoop_makemessages` management command in the `shoop` directory:

```
cd shoop && python -m shoop_workbench shoop_makemessages
```

Docstring coverage

The DocCov script is included for calculating some documentation coverage metrics.

```
python _misc/doccov.py shoop/core -o doccov.html
```

Development Tools

Sanity Checker

The `check_sanity.py` tool does basic text file sanity checking on the codebase.

```
python _misc/check_sanity.py
```

DocCov

The `doccov.py` tool checks for docstring coverage and docstring style.

```
python _misc/doccov.py -o report.html shoop
```

Ensure Code Style

The `ensure_code_style.py` tool has a couple checkers:

- one that checks that all `ForeignKey`'s are declared with `on_delete` clauses
- one that checks that model and form fields have properly translatable verbose names

```
python _misc/ensure_code_style.py shoop
```

Ensure License Headers

The `ensure_license_headers.py` tool checks that all source files have the requisite license header. It can also optionally write the headers to the files by itself.

```
python _misc/ensure_license_headers.py shoop
# or to write changes:
python _misc/ensure_license_headers.py -w shoop
```

Ensure Unicode Literals

The `ensure_unicode_literals.py` tool checks that all Python files have the `from __future__ import unicode_literals` statement, to improve Python 2 compatibility.

It can also optionally add these statements itself.

```
python _misc/ensure_unicode_literals.py shoop
# or to write changes:
python _misc/ensure_unicode_literals.py -w shoop
```

Data model

Data in Shoop is stored into database using regular Django `models` and it is accessed with Django's normal query API. See `shoop.core.models` for list of models in *Shoop Core*.

Extending models

Non-polymorphic models

Basic models (like *Product*, *Category* or *Order*) cannot be replaced. To extend them, create a new model for your extensions and link that to the original model with a `OneToOneField`.

For example:

```
from django.core import models
from shoop.core import models as shoop_models

class MyProduct(models.Model):
    product = models.OneToOneField(shoop_models.Product)

    # fields of the extension...
    my_field = models.CharField(max_length=10)
    ...
```

Todo

Check [Multi-table inheritance](#) for extending models

Note: Even though basic models cannot be replaced, it is possible to replace the `User` model. See [Specifying a custom User model](#).

Polymorphic models

Polymorphic models (like *Contact*) can be extended by inheritance. The polymorphic base class has a `model manager` that makes sure that the returned objects are correct type. For example, when getting all *Contacts* with a query like `Contact.objects.all()`, the returned `QuerySet` may have instances of *PersonContact*, *CompanyContact* and your custom class.

See [django-polymorphic's documentation](#) for details.

The Provides system

The Provides system is Shoop's mechanism for discovering and loading components, both first-party and third-party. Shoop apps use the provides system in various ways.

- The core itself uses Provides for discovering method and supplier modules.
- `shoop.admin` uses Provides to load admin modules, form customizations etc.
- `shoop.front` uses it for URLconf overrides etc.

The provide categories used by Shoop are listed in [Provide Categories](#) but you can also define your own categories as you wish.

Todo

Document the various ways better.

Provides are grouped under different categories, such as `admin_module`, `xtheme_plugin`, `front_urls`, etc.

Declaring Provides

Shoop uses the Django 1.7+ AppConfig system to declare provides.

Quite simply, a developer needs only include a dict with provide categories as the keys and lists of loading specs as values for new provides to be discovered.

```
class PigeonAppConfig(AppConfig):

    provides = {
        'service_provider_admin_form': [
            'pigeon.admin_forms:PigeonShippingAdminForm',
        ],
    }
```

Note: Some provides also require the class named by the spec string to include an `identifier` field. Refer to the implementation guides for particular functionalities for details.

Using Provides

Provide management functions are found in the `shoop.apps.provides` module.

In general, the `shoop.apps.provides.get_provide_objects` method is your most useful entry point.

Provide Categories

Core

admin_category_form_part Additional `FormPart` classes for Category editing.

admin_contact_form_part Additional `FormPart` classes for Contact editing.

admin_contact_group_form_part Additional `FormPart` classes for ContactGroup editing

admin_extend_create_shipment_form Allows providing extension for shipment creation in admin. Should implement the `ShipmentFormModifier` interface.

admin_product_form_part Additional `FormPart` classes for Product editing. (This is used by pricing modules, for instance.)

admin_shop_form_part Additional `FormPart` classes for Shop editing.

admin_module Admin module classes. Practically all of the functionality in the admin is built via admin modules.

discount_module `DiscountModule` for pricing system.

front_service_checkout_phase_provider Allows providing a custom checkout phase for a service (e.g. payment method or shipping method). Should implement the `ServiceCheckoutPhaseProvider` interface.

front_template_helper_namespace Additional namespaces to install in the shoop “package” within template contexts. .. seealso:: [Custom Template Helper Functions](#)

admin_order_toolbar_button Additional `BaseActionButton` subclasses for Order detail. Subclass init should take current order as a parameter.

admin_order_section Additional `OrderSection` subclasses for Order detail sections.

front_urls Lists of frontend URLs to be appended to the usual frontend URLs.

front_urls_post Lists of frontend URLs to be appended to the usual frontend URLs, even after `front_urls`. Most of the time, `front_urls` should do.

front_urls_pre Lists of frontend URLs to be prepended to the usual frontend URLs. Most of the time, `front_urls` should do.

notify_action Notification framework *Action* classes.

notify_condition Notification framework *Condition* classes.

notify_event Notification framework *Event* classes.

order_source_modifier_module *OrderSourceModifierModule* for modifying order source, e.g. in its *get_final_lines*.

pricing_module Pricing module classes; the pricing module in use is set with the `SHOOP_PRICING_MODULE` setting.

service_behavior_component_form Forms for creating service behavior components in Shop Admin. When creating a custom *service behavior component*, provide a form for it via this provide.

service_provider_admin_form Forms for creating service providers in Shop Admin. When creating a custom *service provider* (e.g. *carrier* or *payment processor*), provide a form for it via this provide.

supplier_module Supplier module classes (deriving from *BaseSupplierModule*), as used by *Supplier*.

tax_module Tax module classes; the tax module in use is set with the `SHOOP_TAX_MODULE` setting.

xtheme XTheme themes (full theme sets).

xtheme_plugin XTheme plugins (that are placed into placeholders within themes).

xtheme_resource_injection XTheme resources injection function that takes current context and content as parameters.

Campaigns Provide Categories

campaign_catalog_filter Filters that filter product catalog queryset to find the matching campaigns.

campaign_context_condition Context Conditions that matches against the current context in shop to see if campaign matches.

campaign_product_discount_effect_form Form for handling product discount effects of a catalog campaign. Should be a `ModelForm` with its model being a subclass of *ProductDiscountEffect*.

campaign_basket_condition Conditions that matches against the order source or source lines in basket.

campaign_basket_discount_effect_form Form for handling discount effects of a basket campaign. Should be a `ModelForm` with its model being a subclass of *BasketDiscountEffect*.

campaign_basket_line_effect_form Form for handling line effects of a basket campaign. Should be a `ModelForm` with its model being a subclass of *BasketLineEffect*.

Addons

Shoop contains facilities for installing, detecting, loading and configuring additional functionality with little or no system administration knowledge needed. Packages that can be loaded in this way are called Addons. Addons aren't very special, though: under the surface they are nothing more than standard Django applications that are discovered using the [Setuptools Entry Points mechanism](#). Functionality registration after this occurs via the Shoop *Provides* subsystem.

Configuring your project to load addons

The Shoop addon manager handles adding addons into Django's `INSTALLED_APPS` list during project initialization time.

It's easy to convert a standard Django configuration to be addons enabled.

For instance, take a bare-bones Shoop core installation.

```
INSTALLED_APPS = (
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'shoop.core',
    'shoop.customer_group_pricing',
    'shoop.simple_supplier',
    'shoop.default_tax',
    'shoop.admin',
)
```

The management interface for the addon loader requires one additional configuration key, `SHOOP_ENABLED_ADDONS_FILE`, to name a path to a configuration file that is writable by the application server.

The `shoop.addons.add_enabled_addons()` method manages reading this file, cross-referencing them with the entry points published by Setuptools and adding them into the installed apps list.

Putting this all together,

```
from shoop.addons import add_enabled_addons

# *snip*

# This varies depending on how your particular project arranges writable files.
SHOOP_ENABLED_ADDONS_FILE = os.path.join(BASE_DIR, "enabled_addons")

INSTALLED_APPS = add_enabled_addons(SHOOP_ENABLED_ADDONS_FILE, (
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'shoop.core',
    'shoop.customer_group_pricing',
    'shoop.simple_supplier',
    'shoop.default_tax',
```

```
'shoop.admin',
'shoop.addons',
))

# *snip*
```

will enable your project to load Shoop addons.

Installing and configuring addons

Once `shoop.addons` is in your `INSTALLED_APPS` list, a section for managing addons appears in the administration panel.

Developing addons

As discussed before, addons are simply Django applications with a `Setuptools` `entry_points` stanza in `setup.py`. This means addon development doesn't require any special steps; just adding the new application to a test project's (such as `Workbench`'s) `INSTALLED_APPS` is enough to get you running.

Preparing addons for distribution

When the time comes to actually distribute your new addon, [configure your package according to the PyPUG guidelines](#) and within the `entry_points` section add a `shoop.addon` entry point stanza, such as this (example taken from the `shoop-pugme` addon):

```
setuptools.setup(
    # ...
    entry_points={'shoop.addon': 'shoop_pugme=shoop_pugme'}
)
```

Note: It's recommended you follow the `name=name` format for the entry point definition. Further iterations of addon discovery may change the format.

With this in your `setup.py`, you can now

- Use `python setup.py sdist` to create a source distribution for your addon and install it via the administration panel as you would for any old addon.
- Or run `pip install -e .` to install the addon in your shop's virtualenv in [editable mode](#), then enable the addon via the administration panel.

(If you had manually added the application into your `INSTALLED_APPS` as discussed before, this would be a good time to take it out of there, as otherwise Django will complain about duplicates.)

Implementation of Prices and Taxes in Shoop

This document describes deeper details about price and tax implementation in Shoop from a developer's point of view. To understand the basics, please read [Prices and Taxes in Shoop](#) first.

Types Used for Prices and Taxes

Money

Used to represent money amounts (that are not prices). It is basically a `Decimal` number with a currency.

Price

Used to represent prices. `Price` is a `Money` with an `includes_tax` property. It has two subclasses: `TaxfulPrice` and `TaxlessPrice`.

There should usually be no need to create prices directly with these classes; see *Creating Prices*.

Priceful

An interface for accessing the price information of a product, order line, basket line, or whatever. See *Accessing Prices of Product or Line*.

PriceInfo

A class for describing an item's price information.

PricingModule

An interface for querying prices of products.

PricingContext

A container for variables that affect pricing. Pricing modules may subclass this.

PricingContextable

An interface for objects that can be converted to a pricing context. Instances of `PricingContext` or `HttpRequest` satisfy this interface.

LineTax

An interface for describing a calculated tax of a line in order or basket. Has a reference to the line and to the applied tax and the calculated amount of tax. One line could have several taxes applied, each is presented with a separate `LineTax`.

SourceLineTax

A container for a calculated tax of a `SourceLine` (or `BasketLine`). Implements the `LineTax` interface.

OrderLineTax

A Django model for persistently storing the calculated tax of an `OrderLine`. Implements the `LineTax` interface.

Tax

A Django model for a tax with name, code, and percentage rate or fixed amount. Fixed amounts are not yet supported.

Todo

Fix this when fixed amounts are supported.

TaxableItem

An interface for items that can be taxed. Implemented by `Product`, `ShippingMethod`, `PaymentMethod` and `SourceLine`.

TaxClass

A Django model for a tax class. Taxable items (e.g. products, methods or lines) are grouped to tax classes to make it possible to have different taxation rules for different groups of items.

CustomerTaxGroup

A Django model for grouping customers to make it possible to have different taxation rules for different groups of customers. Shoop assigns separate *CustomerTaxGroup*'s for a `~shoop.core.models.PersonContact` and a *CompanyContact* by default.

TaxModule

An interface for calculating the taxes of an *OrderSource* or any *TaxableItem*. The Shoop Base distribution ships a concrete implementation of a *TaxModule* called *DefaultTaxModule*. It is based on a table of tax rules (saved with *TaxRule* model). See *The Default Tax Module*. Used *TaxModule* can be changed with *SHOOP_TAX_MODULE* setting.

TaxedPrice

A type to represent the return value of tax calculation. Contains a pair of prices, *TaxfulPrice* and *TaxlessPrice*, of which one is the original price before the calculation and the other is the calculated price. Also contains a list of the applied taxes. *TaxedPrice* is the return type of *get_taxed_price_for* method in the *TaxModule* interface.

TaxingContext

A container for variables that affect taxing, such as customer tax group, customer tax number, location (country, postal code, etc.). Used in the *TaxModule* interface. Note: This is *not* usually subclassed.

Creating Prices

When implementing a *PricingModule* or another module that has to create prices, use the *Shop.create_price* method. It makes sure that all prices have the same *price unit*.

Accessing Prices of Product or Line

There is a *Priceful* interface for accessing prices. It is implemented by *OrderLine* and *SourceLine*, *BasketLine*, and *PriceInfo* which is returned e.g. by *get_price_info* method.

Services API

Shoop provides a service API which is based on polymorphic models. Currently Shoop provides two implementations of these services: *payment methods* and *shipping methods*, but the API allows implementing other kind of services too.

From the Services API point of view a *service* is something that may conditionally *generate some new lines* for a basket. Also various aspects of these services can be customized:

- *Availability* of the service for certain basket
- *Costs* of the service for certain basket
- *Effective name* of the service for certain basket

Each service should have a *service provider* and *service choice*, which is selected from a list of service choices specified by the service provider.

Behavior of a service can be customized with *behavior components*. Each behavior component may limit availability of the service, or (conditionally) add costs to it, or affect service delivery time (which in case of shipping methods is the shipping time, but could have other meaning for other kinds of services). It is possible to assign several behavior components to a service.

Notification Framework

The purpose of the notification framework is to be a generic, run-time configurable and code extensible system to inform interested parties about events in the store.

Events could include

- orders being created
- shipments being dispatched
- orders being canceled
- products reaching a given stock threshold
- a user requesting a password reset

et cetera.

Notifications may be delivered over different channels, such as email, SMS, phone or even instant messages, as delivery channels are also pluggable.

It is known that the notification framework partially overlaps the Django signal system as used in Shoop in scope. However Django signals are not user-configurable and their purpose is different.

Glossary

- **Event** - a class of notifiable event
- **Event Variables** - a set of typed variables pertaining to a given event
- **Script** - a configurable object describing the chain of system actions in response to an Event
- **Script Context** - a set of variables populated from the event variables before the script is begun
- **Action** - a configured action within a script, such as “Send email to `x@y.local`”
- **Action Class** - a type of action provided by an app (or built-in to the notification framework), such as “Send Email”
- **Condition** - a configured conditional enabling actions within a script, such as “If Order Contains Product XYZ-1”
- **Condition Class** - a type of condition provided by an app (or built-in to the notification framework), such as “If Order Contains Product”
- **Template** - a possibly multilingual set of textual templates attached to Actions that require configurable text.
- **Attachment** - a named, MIME-typed assumedly binary blob that may be delivered along a notification, such as a PDF order confirmation.

Events

An Event represents a single event that may occur in the system.

Events are registered through the Shoop Apps' `provides` mechanism and must have unique identifiers ("event identifier").

Events provide typed variables that may be utilized in script items, in either variable bindings or message templates.

In addition, a number of system variables are made available for all events.

For instance, an "Order Created" event could provide the variables

- `order` (type `Order`) - the order itself
- `customer_email` (type `email`) - the customer's email address (extracted from the order)
- `customer_phone` (type `phone`) - the customer's phone number (extracted from the order)
- `payment_email` (type `email`) - the payment email address (extracted from the order)
- `shipment_email` (type `email`) - the shipment email address (extracted from the order)
- `language` (type `language`) - the customer's preferred language (extracted from the order)

and a default script that sends a pre-defined text template to the customer email.

A "Password Reset" event could provide the variables

- `user` (type `User`) - the requesting user
- `user_email` (type `email`) - the email address of the user
- `password_reset_url` (type `URL`) - the URL for resetting the password.

Scripts

Event scripts define the rules, i.e. conditions and actions defining what to do when a notification event occurs. A single event may have multiple scripts attached; all of them are executed if they are enabled.

A script may be as simple as "Always -> Send Email -> Stop", or it may have conditions that send emails with different templates depending on the language of the order, or products contained within the order.

Events may provide a script template, which can be loaded for further configuration by the shop administrator. No scripts are loaded by default, though. If no scripts exist for an event, nothing is done when the event occurs.

The model for scripts is a "routing table" with steps of the form "Conditions / Actions / Next". This is somewhat modeled after [uWSGI's Internal Routing](#) system. Note: This would be easy to upgrade to a full-fledged flowchart/data-flow programming environment akin to [Unreal Engine 4's Blueprint Visual Scripting](#) system.

The Conditions set for a script step may be joined with different conditional operators. Currently, "All", "Any" and "None" are implemented. The actions for a single step are executed sequentially.

The actions for "Next" are "Continue" and "Stop". (A "Goto" action is also possible, but it is not considered a requirement at present.) "Continue" will continue executing the routing table from the next step, and "Stop" will cease script execution.

The Condition and Action classes available for notification scripts are also provided via the `provides` mechanism; many actions are built-in (provided by the notification framework itself), but may be extended by other apps. Like Events, Condition classes and Action classes have unique identifiers.

Conditions and Actions are configurable. The configurable variables are set by the Condition classes and Action classes, using the same typology as Event variables.

For instance, Conditional classes could include

- “Language Equals” (configured by a variable of type Language and a constant of type Language)
- “Order Contains Product SKU” (configured by a variable of type Order and a constant of type String)
- “Order Is Paid” (configured by a variable of type Order)

and Action classes might include

- “Send Plain-Text Email” (configured by variable/constant of type Email and a template (see below))
- “Send Text and HTML Email”

Extension Action classes for, say, integration might even contain

- “Send Order To External System XYZ”

The recipients, etc. for emails are configurable, making it possible to easily implement merchant order notifications using the same Actions and Conditions.

Templates

Most, if not all, actions require some sort of templating. The Jinja2 language is used for the templates. Templates may contain multiple sections, such as “Subject” or “Content”; these are set by the Action requesting an editable template, as the template editor is embedded in the Action’s configuration view.

An action may request multilingual templates. Multilingual templates duplicate each section for all languages set in the system configuration. (The sending Action is naturally then expected to be configurable by a variable or constant of type Language.)

Attachments

Many actions may also require attachments and other data such as PDF order confirmations, product manuals, etc. that may or may not be generated during dispatch.

To solve this, the script context also includes a list of Attachment objects (details TBD).

Actions such as “Render Order Confirmation PDF” would add Attachment objects to the context, while sending actions would consume them (optionally without removing them) from the context.

Notification Dispatch

Depending on the deployment and implementation, notification dispatch may occur asynchronously (in a non-blocking manner).

The author currently foresees no use case where asynchronous dispatch would cause issues, and as such, the specification contains no mechanism for declaring an event or script to be forcibly synchronous.

Persistent Notifications

In addition to the script core, the Notify app provides a Django model for notifications stored in the shop’s database. These are currently used only in the admin backend, but could be used in the frontend as well.

Template Design

This part of the documentation covers the structural elements of Shoop's default templates and instructs you on how to create your own customized templates.

To be able to create customized templates you'll need to have understanding of the principles of HTML and CSS.

If you would like to start creating your own customized templates with these instructions you should already have a working Shoop installation with the default frontend theme up and running. If not, you can start by reading [Getting Started guide](#).

Shoop's default frontend theme

Shoop's frontend templates are written with [Jinja2](#) which is a templating engine for Python very similar to Django's templates.

The default frontend theme uses the [Bootstrap 3](#) framework, which consists of Bootstrap's HTML structure and Bootstrap specified CSS classes. If you want to create your own templates, it would require using Bootstrap 3 or overwriting all the template files with your custom HTML structure and HTML classes.

Shoop's template files are easy to modify and basic knowledge of HTML and CSS takes you far. Shoop's frontend and the default theme already include the necessary template tags to print out all the features a basic shop would need. It is fairly simple to add your custom HTML elements around template tags and customize your shop to your needs.

Template folder structure

Shoop utilizes a similar folder structure for all the templates in different apps. All the template files are always included in the app folder `shoop/APP/templates/`.

Within this template folder the folder structure is: `APP/MODULE/TEMPLATE.jinja`. For example, this could be converted into `shoop/product/detail.jinja`

The default frontend theme can be found in `shoop/front/apps/default_theme/`.

Example

The Simple CMS module has a template to show pages created with it. This `page.jinja` template can be found under the Simple CMS template folder: `shoop/simple_cms/templates/` where the path to the template file is `shoop/simple_cms/page.jinja`.

Other default features such as user authentication, customer info, order history, registration and search etc. can be found in their own application templates under `shoop/front/apps/`. Each app has its own template folder containing application specific templates.

Templates have been split into separate files and each file has its own purpose. Template files inherit the base layout from `shoop/base.jinja`.

General

General template files can be found under `shoop/front/templates/`

Base `shoop/front/base.jinja` Defines the structure of your templates. It includes the `<html>`, `<head>` and `<body>` tags, and the general structure of all frontend pages (unless explicitly overridden).

Index `shoop/front/index.jinja` Your shop's home page.

Macros `shoop/front/macros.jinja` Additional template macros that can be used in other template files. For example single product box is rendered with a macro, where it can be called with customized parameters. Also form fields, alerts and order details can be generated with macros.

Includes `shoop/front/includes/` Additional HTML that can be included in pages. In the default frontend theme all the included filenames start with `_`. All navigation related HTML and template tags are included to `base.jinja` and for example you could create a `_footer.jinja` to be included if needed.

Products and Categories

Product and category templates can be found under `shoop/front/templates/`

Detail `shoop/front/product/detail.jinja` The view for a single product. Displays a product and its details. The file uses template tags to include product attributes and ordering sections.

Category `shoop/front/product/category.jinja` A view for a single category. This template lists all the products of the selected category.

Shopping basket

All shopping basket related templates go in the `shoop/front/templates/shoop/front/basket` folder. This includes the default structure of the shopping basket and additional shopping basket elements.

The default shopping basket template also includes the ordering form. This does not apply to shops using multi-phase checkout.

Default Basket `shoop/front/basket/default Basket.jinja` The structure of shopping basket. It includes the shopping basket's contents as a table from a separate file in the `partials` folder. The ordering form is also displayed in this file.

Orders

Order related templates can be found in `shoop/front/templates/shoop/front/order/`.

Complete `shoop/front/order/complete.jinja` Displays the order success message and details of the order.

Payment Canceled `shoop/front/order/payment_canceled.jinja` Template for displaying payment cancellation.

Simple Search

Simple Search is its own application that can be found in the front apps folder: `shoop/apps/simple_search/templates`

Search `shoop/simple_search/search.jinja` The search template includes the search form, search result sorting options and a list of search results.

Authentication

Authentication through the Shoop Front is another sub-app. Its templates can be found in its own folder: `shoop/front/apps/auth/templates/shoop/user/`

Login and Logout Templates for login form and logout message pages.

Password Recovery Password recovery process including the templates for shop and e-mail.

Registration

Registration is another sub-app. Its templates can be found in: `shoop/front/apps/registration/templates`

Registration Form `shoop/registration/register.jinja` Registration form template for new users.

Activation Failed `shoop/registration/activation_failed.jinja` A template for displaying an error message when account activation fails.

Customer Information

Customer information is another sub-app. Its templates can be found in: `shoop/front/apps/customer_information/templates/`

Edit `shoop/customer_information/edit.jinja` Template for editing customer details.

Personal Order History

Personal Order History, another sub-app, naturally has its templates in its own folder. `shoop/front/apps/personal_order_history/templates/`

Order Detail `shoop/personal_order_history/order_detail.jinja` Template for displaying single order's details.

Order List `shoop/personal_order_history/order_list.jinja` Template for listing all the previous personal orders.

Custom Template Helper Functions

This paragraph explains how to register template functions in Shoop's sub-apps. If you are interested in Jinja2's way to do it, please refer to the [Jinja2 documentation](#).

The AppConfig

The `front_template_helper_namespace` category in the `provides` dictionary tells the framework that there are template helper functions to be found in the namespace class (`TemplateHelper`) given.

For more information about `provides` please refer to the [documentation](#)

The TemplateHelper class

This class contains all the functions that are exposed for frontend templates.

Using helpers in a template

The template helpers can be used in templates with `shoop.<module_name>.<TemplateHelper::method>()`. For example `shoop.my_module.get_day_names()`.

Static files

Static files such as images, stylesheets and scripts go under the static folder, using the Django `staticfiles` framework.

You can access static data files in templates by using the `{{ static() }}` function. For example, if you have `img/image.jpg` in your static files, generating a `` tag would be as easy as ``.

Creating custom templates

You may either derive your own theme from the default theme, or write your own from scratch.

The basic principle of deriving custom Shoop templates is not to modify the original files (default frontend themes) within the app directory, but to copy them into to your own application's template directory. If your own application is listed before `shoop.front` (and/or other theme apps) in Django's `INSTALLED_APPS` configuration, Django will prefer your templates over others with the same path.

This means it is possible to overwrite only some of the default files or all of them. If there is no customized template with the same path and filename, Django will use the default file instead.

All the template files that you want to customize go under your application's template folder in the same folder hierarchy as under the original app's `templates` folder. The folder hierarchy for frontend templates was discussed earlier in this document.

Example

Let's say you only would like to make a customized home page for your shop, but leave all other templates as they are. Let's call your application `myshop`.

Simply copy `index.jinja` from `shoop/front/templates/shoop/index.jinja` to your application's template folder `myshop/templates/shoop/index.jinja`, then modify it to your heart's content.

Now let's say you want to tweak the product category view too.

Copy `shoop/front/templates/shoop/product/category.jinja` to `myshop/templates/shoop/product/category.jinja`, then start modifying. As you can see, the template directory structure within your `myshop` application reflects the one in the original app.

Shoop API Documentation

Shoop Application API

See `shoop.apps`.

Shoop Packages and Modules

shoop package

Subpackages

shoop.addons package

Subpackages

shoop.addons.admin_module package

Subpackages

shoop.addons.admin_module.views package

Submodules

shoop.addons.admin_module.views.list module

```
class shoop.addons.admin_module.views.list.AddonEnableDisableForm(**kwargs)
    Bases: django.forms.forms.Form

    get_enabled_addons()

    base_fields = OrderedDict()

    declared_fields = OrderedDict()

    media
class shoop.addons.admin_module.views.list.AddonListView(**kwargs)
    Bases: django.views.generic.edit.FormView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    template_name = 'shoop/admin/addons/list.jinja'

    form_class
        alias of AddonEnableDisableForm

    form_valid(form)

    get_context_data(**kwargs)
```

shoop.addons.admin_module.views.reload module

```
class shoop.addons.admin_module.views.reload.ReloadMethodForm(**kwargs)
    Bases: django.forms.forms.Form

    get_viable_reload_methods()

    get_selected_reload_method()

    base_fields = OrderedDict()

    declared_fields = OrderedDict()

    media
shoop.addons.admin_module.views.reload.finalize_installation_for_enabled_apps()
class shoop.addons.admin_module.views.reload.ReloadView(**kwargs)
    Bases: django.views.generic.edit.FormView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    template_name = 'shoop/admin/addons/reload.jinja'

    form_class
        alias of ReloadMethodForm
```

```

form_valid (form)
get (request, *args, **kwargs)

```

shoop.addons.admin_module.views.upload module

```

class shoop.addons.admin_module.views.upload.AddonUploadForm (data=None,
                                                             files=None,
                                                             auto_id='id_%s',
                                                             prefix=None,      ini-
                                                             tial=None,      er-
                                                             ror_class=<class
                                                             'django.forms.utils.ErrorList'>,
                                                             label_suffix=None,
                                                             empty_permitted=False)

```

Bases: `django.forms.forms.Form`

```
base_fields = OrderedDict([('file', <django.forms.fields.FileField object at 0x7f713cffa0b8>)])
```

```
declared_fields = OrderedDict([('file', <django.forms.fields.FileField object at 0x7f713cffa0b8>)])
```

media

```
class shoop.addons.admin_module.views.upload.AddonUploadView (**kwargs)
```

Bases: `django.views.generic.edit.FormView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
form_class
    alias of AddonUploadForm
```

```
template_name = 'shoop/admin/addons/upload.jinja'
```

```
title = 'Upload Addon'
```

```
form_valid (form)
```

```
get_context_data (**kwargs)
```

```
class shoop.addons.admin_module.views.upload.AddonUploadConfirmView (**kwargs)
```

Bases: `django.views.generic.edit.FormView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
form_class
    alias of Form
```

```
template_name = 'shoop/admin/addons/upload_confirm.jinja'
```

```
title = 'Upload Addon'
```

```
get_addon_path ()
```

```
get_context_data (**kwargs)
```

```
form_valid (form)
```

Module contents

```
class shoop.addons.admin_module.views.AddonListView (**kwargs)
```

Bases: `django.views.generic.edit.FormView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
form_class
    alias of AddonEnableDisableForm
```

```
    form_valid(form)

    get_context_data(**kwargs)

    template_name = 'shoop/admin/addons/list.jinja'
class shoop.addons.admin_module.views.AddonUploadView(**kwargs)
    Bases: django.views.generic.edit.FormView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    form_class
        alias of AddonUploadForm

    form_valid(form)

    get_context_data(**kwargs)

    template_name = 'shoop/admin/addons/upload.jinja'
    title = 'Upload Addon'

class shoop.addons.admin_module.views.AddonUploadConfirmView(**kwargs)
    Bases: django.views.generic.edit.FormView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    form_class
        alias of Form

    form_valid(form)

    get_addon_path()

    get_context_data(**kwargs)

    template_name = 'shoop/admin/addons/upload_confirm.jinja'
    title = 'Upload Addon'

class shoop.addons.admin_module.views.ReloadView(**kwargs)
    Bases: django.views.generic.edit.FormView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    form_class
        alias of ReloadMethodForm

    form_valid(form)

    get(request, *args, **kwargs)

    template_name = 'shoop/admin/addons/reload.jinja'
```

Module contents

```
class shoop.addons.admin_module.AddonModule
    Bases: shoop.admin.base.AdminModule

    name = <django.utils.functional.lazy.<locals>.__proxy__ object>

    category = <django.utils.functional.lazy.<locals>.__proxy__ object>

    breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>

    get_urls()

    get_menu_category_icons()
```

```
get_menu_entries(request)
get_required_permissions()
```

Submodules

shoop.addons.installer module

```
shoop.addons.installer.get_pip_path()
    Try to figure out an explicit path to the Pip executable script.
```

Returns Pip path

Return type `str`

```
class shoop.addons.installer.PackageInstaller
    Bases: object

    install_package(package_path)
    get_log()
```

shoop.addons.manager module

```
shoop.addons.manager.get_addons_from_entry_points()
shoop.addons.manager.get_enabled_addons(file_path)

shoop.addons.manager.set_enabled_addons(file_path, addons, comment=None)
shoop.addons.manager.add_enabled_addons(addon_filename, apps)
```

shoop.addons.reloader module

```
class shoop.addons.reloader.ReloadMethod
    Bases: object

    identifier = None
    title = None
    execute()
    is_viable()
class shoop.addons.reloader.UwsgiReloadMethod
    Bases: shoop.addons.reloader.ReloadMethod

    identifier = 'uwsgi'
    title = 'Reload uWSGI (uwsgi.reload())'
    is_viable()
    execute()
class shoop.addons.reloader.DevServerReloadMethod
    Bases: shoop.addons.reloader.ReloadMethod

    identifier = 'devserver'
    title = 'Reload Django Dev Server'
    is_viable()
    execute()
```

```
class shoop.addons.reloader.ModWSGIReloadMethod
    Bases: shoop.addons.reloader.ReloadMethod

    identifier = 'mod_wsgi'
    title = 'Reload Daemon Mode mod_wsgi'

    is_viable()

    execute()

class shoop.addons.reloader.GunicornReloadMethod
    Bases: shoop.addons.reloader.ReloadMethod

    identifier = 'gunicorn'
    title = 'Reload Gunicorn Master'

    is_parent_an_unicorn()

    is_viable()

    execute()

shoop.addons.reloader.get_reload_method_classes()
```

Module contents

```
shoop.addons.add_enabled_addons(addon_filename, apps)
```

shoop.admin package

Subpackages

shoop.admin.dashboard package

Submodules

shoop.admin.dashboard.blocks module

```
class shoop.admin.dashboard.blocks.DashboardBlock(id, size=None, color=None)
    Bases: object

    type = None

    SIZES = ('small', 'medium', 'large', 'full')

    default_size = 'normal'

class shoop.admin.dashboard.blocks.DashboardContentBlock(id, content, size='normal')
    Bases: shoop.admin.dashboard.blocks.DashboardBlock

    type = 'normal'

    classmethod by_rendering_template(id, request, template_name, context)

class shoop.admin.dashboard.blocks.DashboardValueBlock(id, value, title, **kwargs)
    Bases: shoop.admin.dashboard.blocks.DashboardBlock

    type = 'value'

    default_size = 'small'
```

```
class shoop.admin.dashboard.blocks.DashboardNumberBlock (id, value, title, **kwargs)
    Bases: shoop.admin.dashboard.blocks.DashboardValueBlock
```

```
class shoop.admin.dashboard.blocks.DashboardMoneyBlock (id, value, title, currency,
                                                         **kwargs)
    Bases: shoop.admin.dashboard.blocks.DashboardValueBlock
```

```
class shoop.admin.dashboard.blocks.DashboardChartBlock (id, size='normal')
    Bases: shoop.admin.dashboard.blocks.DashboardBlock
```

```
    type = 'chart'
```

```
    default_size = 'medium'
```

```
    BLOCK_TEMPLATE = '\n <h2 class="block-title"><i class="fa fa-bar-chart"></i> %(title)s</h2>\n <div id="chart-%(id)s">
```

```
    get_chart ()
```

```
        Get the actual chart instance for this block.
```

```
        Returns The chart (or None, if it can't be rendered)
```

```
        Return type shoop.admin.dashboard.charts.Chart|None
```

shoop.admin.dashboard.charts module

```
class shoop.admin.dashboard.charts.Chart (title)
    Bases: object
```

```
    get_config ()
```

```
        Get a JSONable dictionary of configuration data for this chart. This is passed on as CHART_CONFIGS in
        the JS environment and eventually processed by dashboard-charts.js.
```

```
        Returns Dict of configuration
```

```
        Return type dict
```

```
    get_config_json ()
```

```
class shoop.admin.dashboard.charts.BarChart (title, labels)
```

```
    Bases: shoop.admin.dashboard.charts.Chart
```

```
    add_data (name, data)
```

```
    get_config ()
```

shoop.admin.dashboard.utils module

```
shoop.admin.dashboard.utils.get_activity (request, n_entries=30, cutoff_hours=10)
```

```
    Get Activity objects from all modules as a list in latest-first order.
```

Parameters

- **request** (*django.http.request.HttpRequest*) – Request context
- **n_entries** (*int*) – Number of entries to return in total.
- **cutoff_hours** (*float*) – Calculate cutoff datetime so the oldest entry should be at most this old

```
    Returns List of Activity objects
```

```
    Return type list[Activity]
```

Module contents

class `shoop.admin.dashboard.BarChart` (*title, labels*)
Bases: `shoop.admin.dashboard.charts.Chart`

add_data (*name, data*)

get_config ()

class `shoop.admin.dashboard.DashboardBlock` (*id, size=None, color=None*)
Bases: `object`

SIZES = ('small', 'medium', 'large', 'full')

default_size = 'normal'

type = None

class `shoop.admin.dashboard.DashboardChartBlock` (*id, size='normal'*)
Bases: `shoop.admin.dashboard.blocks.DashboardBlock`

BLOCK_TEMPLATE = '\n <h2 class="block-title"><i class="fa fa-bar-chart"></i> %(title)s</h2>\n <div id="chart-%(id)s'

default_size = 'medium'

get_chart ()

Get the actual chart instance for this block.

Returns The chart (or None, if it can't be rendered)

Return type `shoop.admin.dashboard.charts.Chart|None`

type = 'chart'

class `shoop.admin.dashboard.DashboardContentBlock` (*id, content, size='normal'*)
Bases: `shoop.admin.dashboard.blocks.DashboardBlock`

classmethod **by_rendering_template** (*id, request, template_name, context*)

type = 'normal'

class `shoop.admin.dashboard.DashboardMoneyBlock` (*id, value, title, currency, **kwargs*)
Bases: `shoop.admin.dashboard.blocks.DashboardValueBlock`

class `shoop.admin.dashboard.DashboardNumberBlock` (*id, value, title, **kwargs*)
Bases: `shoop.admin.dashboard.blocks.DashboardValueBlock`

class `shoop.admin.dashboard.DashboardValueBlock` (*id, value, title, **kwargs*)
Bases: `shoop.admin.dashboard.blocks.DashboardBlock`

default_size = 'small'

type = 'value'

`shoop.admin.dashboard.get_activity` (*request, n_entries=30, cutoff_hours=10*)
Get Activity objects from all modules as a list in latest-first order.

Parameters

- **request** (`django.http.request.HttpRequest`) – Request context
- **n_entries** (`int`) – Number of entries to return in total.
- **cutoff_hours** (`float`) – Calculate cutoff datetime so the oldest entry should be at most this old

Returns List of Activity objects

Return type `list[Activity]`

shoop.admin.forms package**Submodules****shoop.admin.forms.fields module**

```
class shoop.admin.forms.fields.PercentageField (max_value=None, min_value=None,  
max_digits=None, decimal_places=None,  
*args, **kwargs)
```

Bases: `django.forms.fields.DecimalField`

MULTIPLIER = `Decimal('100')`

prepare_value (*value*)

to_python (*value*)

widget_attrs (*widget*)

```
class shoop.admin.forms.fields.Select2MultipleField (model, *args, **kwargs)
```

Bases: `django.forms.fields.Field`

widget

alias of `SelectMultiple`

to_python (*value*)

widget_attrs (*widget*)

shoop.admin.forms.widgets module

```
class shoop.admin.forms.widgets.BasePopupChoiceWidget (attrs=None, clearable=False,  
empty_text='—')
```

Bases: `django.forms.widgets.Widget`

browse_kind = `None`

filter = `None`

get_browse_markup ()

get_clear_markup ()

render_text (*obj*)

get_object (*value*)

render (*name, value, attrs=None*)

media

```
class shoop.admin.forms.widgets.MediaChoiceWidget (attrs=None, clearable=False,  
empty_text='—')
```

Bases: `shoop.admin.forms.widgets.BasePopupChoiceWidget`

browse_kind = `'media'`

get_object (*value*)

media

```
class shoop.admin.forms.widgets.ImageChoiceWidget (attrs=None, clearable=False,  
empty_text='—')
```

Bases: `shoop.admin.forms.widgets.MediaChoiceWidget`

filter = `'images'`

media

```
class shoop.admin.forms.widgets.ProductChoiceWidget (attrs=None, clearable=False,
                                                    empty_text='—')
    Bases: shoop.admin.forms.widgets.BasePopupChoiceWidget
```

browse_kind = 'product'

get_object (*value*)

media

```
class shoop.admin.forms.widgets.ContactChoiceWidget (attrs=None, clearable=False,
                                                    empty_text='—')
    Bases: shoop.admin.forms.widgets.BasePopupChoiceWidget
```

browse_kind = 'contact'

get_object (*value*)

get_browse_markup ()

media

```
class shoop.admin.forms.widgets.PersonContactChoiceWidget (attrs=None, clear-
                                                            able=False,
                                                            empty_text='—')
    Bases: shoop.admin.forms.widgets.ContactChoiceWidget
```

filter

media

```
class shoop.admin.forms.widgets.PackageProductChoiceWidget (attrs=None, clear-
                                                            able=False,
                                                            empty_text='—')
    Bases: shoop.admin.forms.widgets.ProductChoiceWidget
```

filter = '{"modes": [0, 4]}'

media

Module contents

```
class shoop.admin.forms.ShoopAdminForm (**kwargs)
    Bases: shoop.utils.multilanguage_model_form.MultiLanguageModelForm
```

base_fields = *OrderedDict*()

declared_fields = *OrderedDict*()

media

shoop.admin.modules package

Subpackages

shoop.admin.modules.attributes package

Subpackages

shoop.admin.modules.attributes.views package**Submodules****shoop.admin.modules.attributes.views.edit module**

```

class shoop.admin.modules.attributes.views.edit.AttributeForm (**kwargs)
    Bases: shoop.utils.multilanguage_model_form.MultiLanguageModelForm

    class Meta
        Bases: object

        model
            alias of Attribute

        exclude = ()

    AttributeForm.base_fields = OrderedDict([('identifier', <django.forms.fields.CharField object at 0x7f713d12df98>)]
    AttributeForm.declared_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713d12d000>)]
    AttributeForm.media
class shoop.admin.modules.attributes.views.edit.AttributeEditView (**kwargs)
    Bases: shoop.admin.utils.views.CreateOrUpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of Attribute

    form_class
        alias of AttributeForm

    template_name = 'shoop/admin/attributes/edit.jinja'
    context_object_name = 'attribute'

```

shoop.admin.modules.attributes.views.list module

```

class shoop.admin.modules.attributes.views.list.AttributeListView (**kwargs)
    Bases: shoop.admin.utils.views.PicotableView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of Attribute

    columns = [<shoop.admin.utils.picotable.Column object at 0x7f713c849550>, <shoop.admin.utils.picotable.Column object at 0x7f713c849550>]
    get_queryset ()

```

Module contents

```

class shoop.admin.modules.attributes.views.AttributeEditView (**kwargs)
    Bases: shoop.admin.utils.views.CreateOrUpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    context_object_name = 'attribute'

    form_class
        alias of AttributeForm

```

```
    model
        alias of Attribute

    template_name = 'shoop/admin/attributes/edit.jinja'
class shoop.admin.modules.attributes.views.AttributeListView(**kwargs)
    Bases: shoop.admin.utils.views.PicortableListView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    columns = [<shoop.admin.utils.picotable.Column object at 0x7f713c849550>, <shoop.admin.utils.picotable.Column object at 0x7f713c849550>]
    get_queryset()

    model
        alias of Attribute
```

Module contents

```
class shoop.admin.modules.attributes.AttributeModule
    Bases: shoop.admin.base.AdminModule

    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>
    get_urls()
    get_menu_category_icons()
    get_menu_entries(request)
    get_required_permissions()
    get_model_url(object, kind)
```

shoop.admin.modules.categories package

Subpackages

shoop.admin.modules.categories.views package

Submodules

shoop.admin.modules.categories.views.edit module

```
class shoop.admin.modules.categories.views.edit.CategoryBaseForm(**kwargs)
    Bases: shoop.utils.multilanguage_model_form.MultiLanguageModelForm

    class Meta
        Bases: object

        model
            alias of Category

        fields = ('parent', 'shops', 'status', 'ordering', 'visibility', 'visibility_groups', 'name', 'image', 'description', 'slug')
        widgets = {'visibility': <class 'django.forms.widgets.RadioSelect'>, 'status': <class 'django.forms.widgets.RadioSelect'>}

    CategoryBaseForm.base_fields = OrderedDict([('parent', <mptt.forms.TreeNodeChoiceField object at 0x7f713c849550>), ('shops', <django.forms.fields.ModelChoiceField object at 0x7f713c849550>), ('status', <django.forms.fields.ChoiceField object at 0x7f713c849550>), ('ordering', <django.forms.fields.IntegerField object at 0x7f713c849550>), ('visibility', <django.forms.fields.BooleanField object at 0x7f713c849550>), ('visibility_groups', <django.forms.fields.CharField object at 0x7f713c849550>), ('name', <django.forms.fields.CharField object at 0x7f713c849550>), ('image', <django.forms.fields.ImageField object at 0x7f713c849550>), ('description', <django.forms.fields.CharField object at 0x7f713c849550>), ('slug', <django.forms.fields.SlugField object at 0x7f713c849550>)])
    CategoryBaseForm.declared_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713c849550>), ('image', <django.forms.fields.ImageField object at 0x7f713c849550>), ('description', <django.forms.fields.CharField object at 0x7f713c849550>), ('slug', <django.forms.fields.SlugField object at 0x7f713c849550>), ('visibility_groups', <django.forms.fields.CharField object at 0x7f713c849550>), ('visibility', <django.forms.fields.BooleanField object at 0x7f713c849550>), ('ordering', <django.forms.fields.IntegerField object at 0x7f713c849550>), ('status', <django.forms.fields.ChoiceField object at 0x7f713c849550>), ('shops', <django.forms.fields.ModelChoiceField object at 0x7f713c849550>), ('parent', <mptt.forms.TreeNodeChoiceField object at 0x7f713c849550>)])
```

```

    CategoryBaseForm.media
class shoop.admin.modules.categories.views.edit.CategoryBaseFormPart (request,
                                                                    ob-
                                                                    ject=None)

    Bases: shoop.admin.form_part.FormPart

    priority = -1000

    get_form_defs ()

    form_valid (form)

class shoop.admin.modules.categories.views.edit.CategoryEditView (**kwargs)
    Bases: shoop.admin.form_part.SaveFormPartsMixin, shoop.admin.form_part.FormPartsViewMixin,
shoop.admin.utils.views.CreateOrUpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of Category

    template_name = 'shoop/admin/categories/edit.jinja'

    context_object_name = 'category'

    base_form_part_classes = [<class 'shoop.admin.modules.categories.views.edit.CategoryBaseFormPart'>]

    form_part_class_provide_key = 'admin_category_form_part'

    form_valid (form)

```

shoop.admin.modules.categories.views.list module

```

class shoop.admin.modules.categories.views.list.CategoryListView (**kwargs)
    Bases: shoop.admin.utils.views.PicortableListView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of Category

    columns = [<shoop.admin.utils.picortable.Column object at 0x7f713cb5c128>, <shoop.admin.utils.picortable.Column object at 0x7f713cb5c128>]

    get_queryset ()

    get_object_abstract (instance, item)

```

Module contents

```

class shoop.admin.modules.categories.views.CategoryEditView (**kwargs)
    Bases: shoop.admin.form_part.SaveFormPartsMixin, shoop.admin.form_part.FormPartsViewMixin,
shoop.admin.utils.views.CreateOrUpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    base_form_part_classes = [<class 'shoop.admin.modules.categories.views.edit.CategoryBaseFormPart'>]

    context_object_name = 'category'

    form_part_class_provide_key = 'admin_category_form_part'

    form_valid (form)

    model
        alias of Category

```

```
template_name = 'shoop/admin/categories/edit.jinja'
class shoop.admin.modules.categories.views.CategoryListView(**kwargs)
    Bases: shoop.admin.utils.views.PicotableView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    columns = [<shoop.admin.utils.picotable.Column object at 0x7f713cb5c128>, <shoop.admin.utils.picotable.Column object at 0x7f713cb5c128>]
    get_object_abstract(instance, item)
    get_queryset()
    model
        alias of Category
```

Module contents

```
class shoop.admin.modules.categories.CategoryModule
    Bases: shoop.admin.base.AdminModule

    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    category = <django.utils.functional.lazy.<locals>.__proxy__ object>
    breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>
    get_urls()
    get_menu_entries(request)
    get_search_results(request, query)
    get_required_permissions()
    get_model_url(object, kind)
```

shoop.admin.modules.contact_groups package

Subpackages

shoop.admin.modules.contact_groups.views package

Submodules

shoop.admin.modules.contact_groups.views.delete module

```
class shoop.admin.modules.contact_groups.views.delete.ContactGroupDeleteView(**kwargs)
    Bases: django.views.generic.edit.DeleteView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of ContactGroup

    success_url = <django.utils.functional.lazy.<locals>.__proxy__ object>
```

shoop.admin.modules.contact_groups.views.edit module

```
class shoop.admin.modules.contact_groups.views.edit.ContactGroupEditView(**kwargs)
    Bases: shoop.admin.form_part.SaveFormPartsMixin, shoop.admin.form_part.FormPartsViewMixin,
    shoop.admin.utils.views.CreateOrUpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of ContactGroup

    template_name = 'shoop/admin/contact_groups/edit.jinja'

    context_object_name = 'contact_group'

    base_form_part_classes = [<class 'shoop.admin.modules.contact_groups.views.forms.ContactGroupBaseFormPart']

    form_part_class_provide_key = 'admin_contact_group_form_part'

    form_valid(form)

    get_toolbar()
```

shoop.admin.modules.contact_groups.views.forms module

```
class shoop.admin.modules.contact_groups.views.forms.ContactGroupBaseForm(*args,
                                                                            **kwargs)
    Bases: shoop.utils.multilanguage_model_form.MultiLanguageModelForm

    class Meta
        Bases: object

        model
            alias of ContactGroup

        fields = ('name',)

    ContactGroupBaseForm.save(commit=True)

    ContactGroupBaseForm.base_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713d...)

    ContactGroupBaseForm.declared_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f...)

    ContactGroupBaseForm.media
class shoop.admin.modules.contact_groups.views.forms.ContactGroupBaseFormPart(request,
                                                                              ob-
                                                                              ject=None)
    Bases: shoop.admin.form_part.FormPart

    priority = 0

    get_form_defs()

    form_valid(form)
```

```
class shoop.admin.modules.contact_groups.views.forms.ContactGroupMembersForm (data=None,
                                                                              files=None,
                                                                              auto_id='id_%s',
                                                                              pre-
                                                                              fix=None,
                                                                              ini-
                                                                              tial=None,
                                                                              er-
                                                                              ror_class=<class
                                                                              'django.forms.utils.Error
                                                                              la-
                                                                              bel_suffix=None,
                                                                              empty_permitted=False

Bases: django.forms.forms.Form

base_fields = OrderedDict([('member', <django.forms.models.ModelChoiceField object at 0x7f713d1dd438>)])
declared_fields = OrderedDict([('member', <django.forms.models.ModelChoiceField object at 0x7f713d1dd438>)])
media

class shoop.admin.modules.contact_groups.views.forms.ContactGroupMembersFormSet (**kwargs)
Bases: django.forms.formsets.BaseFormSet

save()

get_selected_and_removed()

class shoop.admin.modules.contact_groups.views.forms.ContactGroupMembersFormPart (request,
                                                                              ob-
                                                                              ject=None)

Bases: shoop.admin.form_part.FormPart

priority = 1

get_form_defs()

form_valid(form)
```

shoop.admin.modules.contact_groups.views.list module

```
class shoop.admin.modules.contact_groups.views.list.ContactGroupListView (**kwargs)
Bases: shoop.admin.utils.picotable.PicotableViewMixin,
        django.views.generic.list.ListView

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model
    alias of ContactGroup

columns = [<shoop.admin.utils.picotable.Column object at 0x7f713c8ac860>, <shoop.admin.utils.picotable.Column object at 0x7f713c8ac860>]

get_queryset()

get_context_data (**kwargs)
```

Module contents

```
class shoop.admin.modules.contact_groups.views.ContactGroupDeleteView (**kwargs)
Bases: django.views.generic.edit.DeleteView

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
```



```

model
    alias of ContactGroup

success_url = <django.utils.functional.lazy.<locals>.__proxy__ object>
class shoop.admin.modules.contact_groups.views.ContactGroupEditView (**kwargs)
    Bases: shoop.admin.form_part.SaveFormPartsMixin, shoop.admin.form_part.FormPartsViewMixin,
shoop.admin.utils.views.CreateOrUpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

base_form_part_classes = [<class 'shoop.admin.modules.contact_groups.views.forms.ContactGroupBaseFormPart
context_object_name = 'contact_group'

form_part_class_provide_key = 'admin_contact_group_form_part'

form_valid (form)

get_toolbar ()

model
    alias of ContactGroup

template_name = 'shoop/admin/contact_groups/edit.jinja'
class shoop.admin.modules.contact_groups.views.ContactGroupListView (**kwargs)
    Bases: shoop.admin.utils.picotable.PicotableViewMixin,
django.views.generic.list.ListView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

columns = [<shoop.admin.utils.picotable.Column object at 0x7f713c8ac860>, <shoop.admin.utils.picotable.Column obje
get_context_data (**kwargs)

get_queryset ()

model
    alias of ContactGroup

```

Module contents

```

class shoop.admin.modules.contact_groups.ContactGroupModule
    Bases: shoop.admin.base.AdminModule

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

category = <django.utils.functional.lazy.<locals>.__proxy__ object>

breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>

get_urls ()

get_menu_entries (request)

get_required_permissions ()

get_model_url (object, kind)

```

shoop.admin.modules.contacts package

Subpackages

shoop.admin.modules.contacts.views package

Submodules

shoop.admin.modules.contacts.views.detail module

class shoop.admin.modules.contacts.views.detail.**ContactDetailToolbar** (*contact*,
request)

Bases: *shoop.admin.toolbar.Toolbar*

build_renew_password_button()

build_new_user_button()

build_new_order_button()

build_deactivate_button()

build()

class shoop.admin.modules.contacts.views.detail.**ContactDetailView** (***kwargs*)

Bases: *django.views.generic.detail.DetailView*

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of *Contact*

template_name = 'shoop/admin/contacts/detail.jinja'

context_object_name = 'contact'

get_context_data (***kwargs*)

post (*request*, **args*, ***kwargs*)

shoop.admin.modules.contacts.views.edit module

class shoop.admin.modules.contacts.views.edit.**ContactBaseForm** (*bind_user=None*,
args*, *kwargs*)

Bases: *django.forms.models.BaseModelForm*

This form is notoriously confusing in that it works in several different modes depending on what the instance being passed in is.

If the instance is an unsaved object, the form will show fields for the common superclass *Contact* as well as a type selection field. When saving the object, a *_new_* instance is created, as its class will have been specialized into the actual concrete polymorphic type. (I said this is confusing.)

If the instance is a saved object, its type is checked and only the related fields are shown and none of that specialization stuff occurs.

FIELDS_BY_MODEL_NAME = {'Contact': ('is_active', 'language', 'marketing_permission', 'phone', 'www', 'timezone', 'p

formfield_callback (*f*, ***kwargs*)

generate_fields()

set_model_from_cleaned_data()

save (*commit=True*)

class shoop.admin.modules.contacts.views.edit.**ContactBaseFormPart** (*request*, *ob-*
ject=None)

Bases: *shoop.admin.form_part.FormPart*

priority = -1000

```

    get_form_defs()

    form_valid(form)

class shoop.admin.modules.contacts.views.edit.AddressForm(data=None, files=None,
                                                         auto_id='id_%s', pre-
                                                         fix=None, initial=None,
                                                         error_class=<class
                                                         'django.forms.utils.ErrorList'>,
                                                         label_suffix=None,
                                                         empty_permitted=False,
                                                         instance=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    model
        alias of MutableAddress

    fields = ('prefix', 'name', 'suffix', 'name_ext', 'phone', 'email', 'street', 'street2', 'street3', 'postal_code', 'city', 'r
AddressForm.base_fields = OrderedDict([('prefix', <django.forms.fields.CharField object at 0x7f7141e63160>), ('n
AddressForm.declared_fields = OrderedDict()

AddressForm.media

class shoop.admin.modules.contacts.views.edit.ContactAddressesFormPart(request,
                                                                    ob-
                                                                    ject=None)

Bases: shoop.admin.form_part.FormPart

priority = -900

get_form_defs()

form_valid(form)

class shoop.admin.modules.contacts.views.edit.ContactEditView(**kwargs)
Bases: shoop.admin.form_part.SaveFormPartsMixin, shoop.admin.form_part.FormPartsViewMixin,
shoop.admin.utils.views.CreateOrUpdateView

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model
    alias of Contact

template_name = 'shoop/admin/contacts/edit.jinja'

context_object_name = 'contact'

base_form_part_classes = [<class 'shoop.admin.modules.contacts.views.edit.ContactBaseFormPart'>, <class 'shoop
form_part_class_provide_key = 'admin_contact_form_part'

form_valid(form)

get_toolbar()

shoop.admin.modules.contacts.views.list module

class shoop.admin.modules.contacts.views.list.ContactListView(**kwargs)
Bases: shoop.admin.utils.views.PicotableView

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```

```
model
    alias of Contact

columns = [<shoop.admin.utils.picotable.Column object at 0x7f713ccf27b8>, <shoop.admin.utils.picotable.Column object at 0x7f713ccf27b8>]

get_queryset ()

get_type_display (instance)

get_object_abstract (instance, item)
```

shoop.admin.modules.contacts.views.reset module

```
class shoop.admin.modules.contacts.views.reset.ContactResetPasswordView (**kwargs)
    Bases: shoop.admin.modules.users.views.password.UserResetPasswordView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    get_contact ()

    get_object (queryset=None)

    get_success_url ()
```

Module contents

```
class shoop.admin.modules.contacts.views.ContactListView (**kwargs)
    Bases: shoop.admin.utils.views.PicotableListView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    columns = [<shoop.admin.utils.picotable.Column object at 0x7f713ccf27b8>, <shoop.admin.utils.picotable.Column object at 0x7f713ccf27b8>]

    get_object_abstract (instance, item)

    get_queryset ()

    get_type_display (instance)

    model
        alias of Contact

class shoop.admin.modules.contacts.views.ContactDetailView (**kwargs)
    Bases: django.views.generic.detail.DetailView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    context_object_name = 'contact'

    get_context_data (**kwargs)

    model
        alias of Contact

    post (request, *args, **kwargs)

    template_name = 'shoop/admin/contacts/detail.jinja'

class shoop.admin.modules.contacts.views.ContactResetPasswordView (**kwargs)
    Bases: shoop.admin.modules.users.views.password.UserResetPasswordView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    get_contact ()
```

```

    get_object (queryset=None)
    get_success_url ()
class shoop.admin.modules.contacts.views.ContactEditView (**kwargs)
    Bases: shoop.admin.form_part.SaveFormPartsMixin, shoop.admin.form_part.FormPartsViewMixin,
shoop.admin.utils.views.CreateOrUpdateView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    base_form_part_classes = [<class 'shoop.admin.modules.contacts.views.edit.ContactBaseFormPart'>, <class 'shoop
    context_object_name = 'contact'
    form_part_class_provide_key = 'admin_contact_form_part'
    form_valid (form)
    get_toolbar ()
    model
        alias of Contact
    template_name = 'shoop/admin/contacts/edit.jinja'

```

Submodules

shoop.admin.modules.contacts.dashboard module

```
shoop.admin.modules.contacts.dashboard.get_active_customers_block (request)
```

Module contents

```

class shoop.admin.modules.contacts.ContactModule
    Bases: shoop.admin.base.AdminModule
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    category = <django.utils.functional.lazy.<locals>.__proxy__ object>
    breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>
    get_urls ()
    get_menu_category_icons ()
    get_menu_entries (request)
    get_required_permissions ()
    get_search_results (request, query)
    get_dashboard_blocks (request)
    get_model_url (object, kind)

```

shoop.admin.modules.demo package

Module contents

class `shoop.admin.modules.demo.DemoModule`

Bases: `shoop.admin.base.AdminModule`

name = `<django.utils.functional.lazy.<locals>.__proxy__ object>`

get_urls ()

check_demo_optin (*request*)

Check whether or not the user has opted in to see demo content. This may be toggled with `?demo=0` or `?demo=1`, and it's a persistent session flag.

Parameters **request** (`django.http.HttpRequest`) – HTTP request

Returns Opt-in flag

Return type `bool`

get_menu_entries (*request*)

get_search_results (*request*, *query*)

get_notifications (*request*)

get_dashboard_blocks (*request*)

get_activity (*request*, *cutoff*)

shoop.admin.modules.manufacturers package

Subpackages

shoop.admin.modules.manufacturers.views package

Submodules

shoop.admin.modules.manufacturers.views.edit module

class `shoop.admin.modules.manufacturers.views.edit.ManufacturerForm` (*data=None*,
files=None,
auto_id='id_%s',
pre-
fix=None,
ini-
tial=None,
er-
ror_class=<class
'django.forms.utils.ErrorList'>,
la-
bel_suffix=None,
empty_permitted=False,
in-
stance=None)

Bases: `django.forms.models.ModelForm`

class **Meta**

Bases: `object`

```

    model
        alias of Manufacturer

    exclude = ('identifier', 'created_on')

ManufacturerForm.base_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713cd6f7b>)]

ManufacturerForm.declared_fields = OrderedDict()

ManufacturerForm.media

class shoop.admin.modules.manufacturers.views.edit.ManufacturerEditView(**kwargs)
    Bases: shoop.admin.utils.views.CreateOrUpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of Manufacturer

    form_class
        alias of ManufacturerForm

    template_name = 'shoop/admin/manufacturers/edit.jinja'

    context_object_name = 'manufacturer'

```

shoop.admin.modules.manufacturers.views.list module

```

class shoop.admin.modules.manufacturers.views.list.ManufacturerListView(**kwargs)
    Bases: shoop.admin.utils.views.PicotableView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of Manufacturer

    columns = [<shoop.admin.utils.picotable.Column object at 0x7f713cfd6400>]

```

Module contents

```

class shoop.admin.modules.manufacturers.views.ManufacturerEditView(**kwargs)
    Bases: shoop.admin.utils.views.CreateOrUpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    context_object_name = 'manufacturer'

    form_class
        alias of ManufacturerForm

    model
        alias of Manufacturer

    template_name = 'shoop/admin/manufacturers/edit.jinja'

class shoop.admin.modules.manufacturers.views.ManufacturerListView(**kwargs)
    Bases: shoop.admin.utils.views.PicotableView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    columns = [<shoop.admin.utils.picotable.Column object at 0x7f713cfd6400>]

    model
        alias of Manufacturer

```

Module contents

```
class shoop.admin.modules.manufacturers.ManufacturerModule
    Bases: shoop.admin.base.AdminModule

    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>
    get_urls()
    get_menu_entries(request)
    get_required_permissions()
    get_model_url(object, kind)
```

shoop.admin.modules.media package

Submodules

shoop.admin.modules.media.utils module

```
shoop.admin.modules.media.utils.delete_folder(folder)
    Delete a Filer folder and move files and subfolders up to the parent.

    Parameters folder (filer.models.Folder) – Folder

    Returns Success message

    Return type str
```

shoop.admin.modules.media.views module

```
shoop.admin.modules.media.views.get_folder_name(folder)
class shoop.admin.modules.media.views.MediaBrowserView(**kwargs)
    Bases: django.views.generic.base.TemplateView

    A view for browsing media.

    Most of this is just a JSON API that the Javascript (static_src/media/browser) uses.

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    template_name = 'shoop/admin/media/browser.jinja'
    title = <django.utils.functional.lazy.<locals>.__proxy__ object>
    get_context_data(**kwargs)
    get(request, *args, **kwargs)
    post(request, *args, **kwargs)
    handle_get_folders(data)
    handle_post_new_folder(data)
    handle_get_folder(data)
    handle_upload()
    handle_post_rename_folder(data)
    handle_post_delete_folder(data)
```



```

handle_post_rename_file(data)
handle_post_delete_file(data)
handle_post_move_file(data)

```

Module contents

class `shoop.admin.modules.media.MediaModule`

Bases: `shoop.admin.base.AdminModule`

A module for handling site media. Basically a frontend for the Django-Filer app.

name = `<django.utils.functional.lazy.<locals>.__proxy__ object>`

get_urls ()

get_menu_category_icons ()

get_required_permissions ()

get_menu_entries (*request*)

shoop.admin.modules.orders package

Subpackages

shoop.admin.modules.orders.views package

Submodules

shoop.admin.modules.orders.views.detail module

class `shoop.admin.modules.orders.views.detail.OrderDetailView` (***kwargs*)

Bases: `django.views.generic.detail.DetailView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of `Order`

template_name = `'shoop/admin/orders/detail.jinja'`

context_object_name = `'order'`

get_toolbar ()

get_context_data (***kwargs*)

class `shoop.admin.modules.orders.views.detail.OrderSetStatusView` (***kwargs*)

Bases: `django.views.generic.detail.DetailView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of `Order`

get (*request*, **args*, ***kwargs*)

post (*request*, **args*, ***kwargs*)

shoop.admin.modules.orders.views.edit module

```
shoop.admin.modules.orders.views.edit.create_order_from_state (state, **kwargs)
shoop.admin.modules.orders.views.edit.update_order_from_state (state, or-
                                                                der_to_update,
                                                                **kwargs)

shoop.admin.modules.orders.views.edit.create_source_from_state (state,
                                                                **kwargs)

shoop.admin.modules.orders.views.edit.encode_address (address)
shoop.admin.modules.orders.views.edit.encode_shop (shop)
shoop.admin.modules.orders.views.edit.encode_method (method)
shoop.admin.modules.orders.views.edit.encode_line (line)
shoop.admin.modules.orders.views.edit.get_line_data_for_edit (shop, line)
shoop.admin.modules.orders.views.edit.get_price_info (shop, customer, product, quan-
                                                         tity)

    Get price info of given product for given context parameters.
```

class shoop.admin.modules.orders.views.edit.**OrderEditView** (***kwargs*)

Bases: *shoop.admin.utils.views.CreateOrUpdateView*

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of `Order`

template_name = 'shoop/admin/orders/create.jinja'

context_object_name = 'order'

title = 'Create Order'

fields = []

get_context_data (***kwargs*)

get_toolbar ()

get_config ()

get_initial_order_data ()

get_customer_data (customer_id)

dispatch (request, *args, ***kwargs*)

dispatch_command (request)

handle_product_data (request)

handle_customer_data (request)

handle_customer_details (request)

handle_source_data (request)

handle_finalize (request)

shoop.admin.modules.orders.views.list module

class shoop.admin.modules.orders.views.list.**OrderListView** (**kwargs)

Bases: `shoop.admin.utils.views.PicotableListView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of `Order`

columns = [`<shoop.admin.utils.picotable.Column object at 0x7f713cb27048>`, `<shoop.admin.utils.picotable.Column object at 0x7f713cb27048>`]

get_queryset ()

format_order_date (instance, *args, **kwargs)

format_taxful_total_price (instance, *args, **kwargs)

get_object_abstract (instance, item)

shoop.admin.modules.orders.views.log module

class shoop.admin.modules.orders.views.log.**NewLogEntryView** (**kwargs)

Bases: `django.views.generic.base.View`

Create a log note item associated with a particular order.

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

post (request, *args, **kwargs)

shoop.admin.modules.orders.views.payment module

class shoop.admin.modules.orders.views.payment.**OrderCreatePaymentView** (**kwargs)

Bases: `django.views.generic.edit.UpdateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of `Order`

template_name = 'shoop/admin/orders/create_payment.jinja'

context_object_name = 'order'

form_class

alias of `Form`

get_context_data (**kwargs)

get_form_kwargs ()

get_form (form_class=None)

form_invalid (form)

form_valid (form)

shoop.admin.modules.orders.views.refund module

```
class shoop.admin.modules.orders.views.refund.RefundForm(data=None,      files=None,
                                                         auto_id='id_%s',      pre-
                                                         fix=None,      initial=None,
                                                         error_class=<class
                                                         'django.forms.utils.ErrorList'>,
                                                         label_suffix=None,
                                                         empty_permitted=False)

    Bases: django.forms.forms.Form

    clean_line_number()

    clean_quantity()

    clean_amount()

    base_fields = OrderedDict([('line_number', <django.forms.fields.ChoiceField object at 0x7f713ca64d30>), ('quantity', <django.forms.fields.IntegerField object at 0x7f713ca64d30>)])

    declared_fields = OrderedDict([('line_number', <django.forms.fields.ChoiceField object at 0x7f713ca64d30>), ('quantity', <django.forms.fields.IntegerField object at 0x7f713ca64d30>)])

    media

class shoop.admin.modules.orders.views.refund.OrderCreateRefundView(**kwargs)
    Bases: django.views.generic.edit.UpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of Order

    template_name = 'shoop/admin/orders/create_refund.jinja'

    context_object_name = 'order'

    form_class
        alias of RefundFormFormSet

    get_context_data(**kwargs)

    get_form_kwargs()

    get_form(form_class=None)

    form_valid(form)

class shoop.admin.modules.orders.views.refund.FullRefundConfirmationForm(data=None,
                                                         files=None,
                                                         auto_id='id_%s',
                                                         pre-
                                                         fix=None,
                                                         ini-
                                                         tial=None,
                                                         er-
                                                         ror_class=<class
                                                         'django.forms.utils.ErrorList'>,
                                                         la-
                                                         bel_suffix=None,
                                                         empty_permitted=False)

    Bases: django.forms.forms.Form

    base_fields = OrderedDict([('restock_products', <django.forms.fields.BooleanField object at 0x7f713c8acc0>)])

    declared_fields = OrderedDict([('restock_products', <django.forms.fields.BooleanField object at 0x7f713c8acc0>)])

    media
```

```
class shoop.admin.modules.orders.views.refund.OrderCreateFullRefundView (**kwargs)
    Bases: django.views.generic.edit.UpdateView
```

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of `Order`

template_name = 'shoop/admin/orders/create_full_refund.jinja'

context_object_name = 'order'

form_class

alias of `FullRefundConfirmationForm`

get_context_data (**kwargs)

get_form_kwargs ()

form_valid (form)

shoop.admin.modules.orders.views.shipment module

```
class shoop.admin.modules.orders.views.shipment.ShipmentFormModifier
```

Bases: `object`

get_extra_fields (order)

Extra fields for shipment creation view.

Parameters **order** (`shoop.core.models.Order`) – Order linked to form

Returns List of extra fields that should be added to form.

Field list ends without a blank line; unexpected unindent.

Tuple should contain field name and Django form field. :rtype: list[(str,django.forms.Field)]

clean_hook (form)

Extra clean for shipment creation form.

This hook will be called in `clean` method of the form, after calling parent `clean`. Implementor of this hook may call `add_error` to add errors to form or modify the `form.cleaned_data` dictionary.

Parameters **form** (`ShipmentForm`) – Form that is currently cleaned

Return type `None`

form_valid_hook (form, shipment)

Extra form valid handler for shipment creation view.

This is called from `OrderCreateShipmentView` just before the `Order.create_shipment`

Parameters

- **form** (`ShipmentForm`) – Form that is currently handled
- **shipment** (`shoop.core.models.Shipment`) – Unsaved shipment

Return type `None`

```
class shoop.admin.modules.orders.views.shipment.ShipmentForm(data=None,
                                                             files=None,
                                                             auto_id='id_%s',
                                                             prefix=None,      ini-
                                                             tial=None,      er-
                                                             ror_class=<class
                                                             'django.forms.utils.ErrorList'>,
                                                             label_suffix=None,
                                                             empty_permitted=False)

Bases: django.forms.forms.Form

clean()

base_fields = OrderedDict()

declared_fields = OrderedDict()

media

class shoop.admin.modules.orders.views.shipment.OrderCreateShipmentView(**kwargs)
Bases: django.views.generic.edit.UpdateView

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model
    alias of Order

template_name = 'shoop/admin/orders/create_shipment.jinja'

context_object_name = 'order'

form_class
    alias of ShipmentForm

get_context_data(**kwargs)

get_form_kwargs()

get_form(form_class=None)

form_invalid(form)

form_valid(form)
```

Module contents

```
class shoop.admin.modules.orders.views.NewLogEntryView(**kwargs)
Bases: django.views.generic.base.View

Create a log note item associated with a particular order.

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

post(request, *args, **kwargs)

class shoop.admin.modules.orders.views.OrderDetailView(**kwargs)
Bases: django.views.generic.detail.DetailView

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

context_object_name = 'order'

get_context_data(**kwargs)

get_toolbar()
```

```

    model
        alias of Order

    template_name = 'shoop/admin/orders/detail.jinja'

```

class `shoop.admin.modules.orders.views.OrderEditView(**kwargs)`
 Bases: `shoop.admin.utils.views.CreateOrUpdateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```

    context_object_name = 'order'

    dispatch(request, *args, **kwargs)

    dispatch_command(request)

    fields = []

    get_config()

    get_context_data(**kwargs)

    get_customer_data(customer_id)

    get_initial_order_data()

    get_toolbar()

    handle_customer_data(request)

    handle_customer_details(request)

    handle_finalize(request)

    handle_product_data(request)

    handle_source_data(request)

    model
        alias of Order

    template_name = 'shoop/admin/orders/create.jinja'

    title = 'Create Order'

```

class `shoop.admin.modules.orders.views.OrderListView(**kwargs)`
 Bases: `shoop.admin.utils.views.PicotableView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```

    columns = [<shoop.admin.utils.picotable.Column object at 0x7f713cb27048>, <shoop.admin.utils.picotable.Column object at 0x7f713cb27048>]

    format_order_date(instance, *args, **kwargs)

    format_taxful_total_price(instance, *args, **kwargs)

    get_object_abstract(instance, item)

    get_queryset()

    model
        alias of Order

```

class `shoop.admin.modules.orders.views.OrderCreatePaymentView(**kwargs)`
 Bases: `django.views.generic.edit.UpdateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```

    context_object_name = 'order'

```

```
form_class
    alias of Form

form_invalid(form)

form_valid(form)

get_context_data(**kwargs)

get_form(form_class=None)

get_form_kwargs()

model
    alias of Order

template_name = 'shoop/admin/orders/create_payment.jinja'
```

class shoop.admin.modules.orders.views.**OrderCreateFullRefundView**(***kwargs*)
Bases: `django.views.generic.edit.UpdateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
context_object_name = 'order'

form_class
    alias of FullRefundConfirmationForm

form_valid(form)

get_context_data(**kwargs)

get_form_kwargs()

model
    alias of Order

template_name = 'shoop/admin/orders/create_full_refund.jinja'
```

class shoop.admin.modules.orders.views.**OrderCreateRefundView**(***kwargs*)
Bases: `django.views.generic.edit.UpdateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
context_object_name = 'order'

form_class
    alias of RefundFormFormSet

form_valid(form)

get_context_data(**kwargs)

get_form(form_class=None)

get_form_kwargs()

model
    alias of Order

template_name = 'shoop/admin/orders/create_refund.jinja'
```

class shoop.admin.modules.orders.views.**OrderCreateShipmentView**(***kwargs*)
Bases: `django.views.generic.edit.UpdateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
context_object_name = 'order'
```



```

form_class
    alias of ShipmentForm

form_invalid(form)

form_valid(form)

get_context_data(**kwargs)

get_form(form_class=None)

get_form_kwargs()

model
    alias of Order

template_name = 'shoop/admin/orders/create_shipment.jinja'

class shoop.admin.modules.orders.views.OrderSetStatusView(**kwargs)
    Bases: django.views.generic.detail.DetailView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    get(request, *args, **kwargs)

    model
        alias of Order

    post(request, *args, **kwargs)

```

Submodules

shoop.admin.modules.orders.dashboard module

```

shoop.admin.modules.orders.dashboard.get_orders_by_currency(currency)

class shoop.admin.modules.orders.dashboard.OrderValueChartDashboardBlock(id,
                                                                           cur-
                                                                           rency,
                                                                           **kwargs)

    Bases: shoop.admin.dashboard.blocks.DashboardChartBlock

    get_chart()

shoop.admin.modules.orders.dashboard.get_subtitle(count)

shoop.admin.modules.orders.dashboard.get_sales_of_the_day_block(request, cur-
                                                                rency)

shoop.admin.modules.orders.dashboard.get_lifetime_sales_block(request, cur-
                                                                rency)

shoop.admin.modules.orders.dashboard.get_avg_purchase_size_block(request, cur-
                                                                rency)

shoop.admin.modules.orders.dashboard.get_open_orders_block(request, currency)

shoop.admin.modules.orders.dashboard.get_order_value_chart_dashboard_block(request,
                                                                cur-
                                                                rency)

```

shoop.admin.modules.orders.json_order_creator module**class** shoop.admin.modules.orders.json_order_creator.JsonOrderCreatorBases: `object`**static** `safe_get_first` (*model*, ***lookup*)**add_error** (*error*)**is_valid****errors****create_source_from_state** (*state*, *creator=None*, *ip_address=None*, *save=False*, *order_to_update=None*)

Create an order source from a state dict unserialized from JSON.

Parameters

- **state** (*dict*) – State dictionary
- **creator** (*django.contrib.auth.models.User|None*) – Creator user
- **save** (*boolean*) – Flag whether order customer and addresses is saved to database
- **order_to_update** (*shoop.core.models.Order|None*) – Order object to edit

Returns The created order source, or None if something failed along the way**Return type** OrderSource|None**create_order_from_state** (*state*, *creator=None*, *ip_address=None*)

Create an order from a state dict unserialized from JSON.

Parameters

- **state** (*dict*) – State dictionary
- **creator** (*django.contrib.auth.models.User|None*) – Creator user
- **ip_address** (*str*) – Remote IP address (IPv4 or IPv6)

Returns The created order, or None if something failed along the way**Return type** Order|None**update_order_from_state** (*state*, *order_to_update*, *modified_by=None*)

Update an order from a state dict unserialized from JSON.

Parameters

- **state** (*dict*) – State dictionary
- **order_to_update** (*shoop.core.models.Order*) – Order object to edit

Returns The created order, or None if something failed along the way**Return type** Order|None**shoop.admin.modules.orders.receivers module****shoop.admin.modules.orders.receivers.handle_custom_payment_return_requests** (*sender*,
or-
der,
**args*,
***kwargs*)

shoop.admin.modules.orders.sections module

```

class shoop.admin.modules.orders.sections.PaymentOrderSection
    Bases: shoop.admin.base.OrderSection

    identifier = 'payments'
    name = 'Payments'
    icon = 'fa-dollar'
    template = 'shoop/admin/orders/_detail_payments.jinja'
    order = 1
    static visible_for_order (order)
    static get_context_data (order)
class shoop.admin.modules.orders.sections.ContentsOrderSection
    Bases: shoop.admin.base.OrderSection

    identifier = 'contents'
    name = 'Order Contents'
    icon = 'fa-file-text'
    template = 'shoop/admin/orders/_order_contents.jinja'
    order = 2
    static visible_for_order (order)
    static get_context_data (order)
class shoop.admin.modules.orders.sections.LogEntriesOrderSection
    Bases: shoop.admin.base.OrderSection

    identifier = 'log_entries'
    name = 'Log Entries'
    icon = 'fa-pencil'
    template = 'shoop/admin/orders/_order_log_entries.jinja'
    extra_js = 'shoop/admin/orders/_order_log_entries_extra_js.jinja'
    order = 3
    static visible_for_order (order)
    static get_context_data (order)

```

Module contents

```

class shoop.admin.modules.orders.OrderModule (currency=None, *args, **kwargs)
    Bases: shoop.admin.currencybound.CurrencyBound, shoop.admin.base.AdminModule

    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>
    get_urls ()
    get_menu_category_icons ()
    get_menu_entries (request)
    get_required_permissions ()

```

```
get_search_results(request, query)
get_dashboard_blocks(request)
get_notifications(request)
get_model_url(object, kind)
```

shoop.admin.modules.permission_groups package

Subpackages

shoop.admin.modules.permission_groups.views package

Submodules

shoop.admin.modules.permission_groups.views.edit module

```
class shoop.admin.modules.permission_groups.views.edit.PermissionGroupForm(*args,
                                                                            **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        model
            alias of Group

        exclude = ('permissions',)

    PermissionGroupForm.clean_members()
    PermissionGroupForm.clean()
    PermissionGroupForm.save()
    PermissionGroupForm.base_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713c8a...
    PermissionGroupForm.declared_fields = OrderedDict()
    PermissionGroupForm.media

class shoop.admin.modules.permission_groups.views.edit.PermissionGroupEditView(**kwargs)
    Bases: shoop.admin.utils.views.CreateOrUpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of Group

    form_class
        alias of PermissionGroupForm

    template_name = 'shoop/admin/permission_groups/edit.jinja'
    context_object_name = 'permission_group'
    add_form_errors_as_messages = True
```

shoop.admin.modules.permission_groups.views.list module

class shoop.admin.modules.permission_groups.views.list.**PermissionGroupListView** (**kwargs)
Bases: *shoop.admin.utils.views.PicotableListView*

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of Group

columns = [<shoop.admin.utils.picotable.Column object at 0x7f713c0184e0>]

get_context_data (**kwargs)

Module contents

class shoop.admin.modules.permission_groups.views.**PermissionGroupEditView** (**kwargs)
Bases: *shoop.admin.utils.views.CreateOrUpdateView*

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

add_form_errors_as_messages = True

context_object_name = 'permission_group'

form_class

alias of PermissionGroupForm

model

alias of Group

template_name = 'shoop/admin/permission_groups/edit.jinja'

class shoop.admin.modules.permission_groups.views.**PermissionGroupListView** (**kwargs)
Bases: *shoop.admin.utils.views.PicotableListView*

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

columns = [<shoop.admin.utils.picotable.Column object at 0x7f713c0184e0>]

get_context_data (**kwargs)

model

alias of Group

Module contents

class shoop.admin.modules.permission_groups.**PermissionGroupModule**
Bases: *shoop.admin.base.AdminModule*

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

category = <django.utils.functional.lazy.<locals>.__proxy__ object>

breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>

get_urls ()

get_menu_category_icons ()

get_menu_entries (request)

get_required_permissions ()

get_model_url (object, kind)

shoop.admin.modules.product_types package

Subpackages

`shoop.admin.modules.product_types.views` package

Submodules

`shoop.admin.modules.product_types.views.edit` module

```
class shoop.admin.modules.product_types.views.edit.ProductTypeForm (**kwargs)
    Bases: shoop.utils.multilanguage_model_form.MultiLanguageModelForm
```

```
    class Meta
```

```
        Bases: object
```

```
        model
```

```
            alias of ProductType
```

```
        exclude = ()
```

```
        widgets = {'attributes': <class 'django.forms.widgets.CheckboxSelectMultiple'>}
```

```
    ProductTypeForm.base_fields = OrderedDict([('attributes', <django.forms.models.ModelMultipleChoiceField object at 0x7f713be713be>)])
```

```
    ProductTypeForm.declared_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713be713be>)])
```

```
    ProductTypeForm.media
```

```
class shoop.admin.modules.product_types.views.edit.ProductTypeEditView (**kwargs)
```

```
    Bases: shoop.admin.utils.views.CreateOrUpdateView
```

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
    model
```

```
        alias of ProductType
```

```
    form_class
```

```
        alias of ProductTypeForm
```

```
    template_name = 'shoop/admin/product_types/edit.jinja'
```

```
    context_object_name = 'product_type'
```

`shoop.admin.modules.product_types.views.list` module

```
class shoop.admin.modules.product_types.views.list.ProductTypeListView (**kwargs)
```

```
    Bases: shoop.admin.utils.views.PicotableView
```

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
    model
```

```
        alias of ProductType
```

```
    columns = [<shoop.admin.utils.picotable.Column object at 0x7f713c176a90>, <shoop.admin.utils.picotable.Column object at 0x7f713c176a90>]
```

```
    get_queryset ()
```

Module contents

```
class shoop.admin.modules.product_types.views.ProductTypeEditView (**kwargs)
```

```
    Bases: shoop.admin.utils.views.CreateOrUpdateView
```

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
    context_object_name = 'product_type'
```

```

form_class
    alias of ProductTypeForm

model
    alias of ProductType

template_name = 'shoop/admin/product_types/edit.jinja'
class shoop.admin.modules.product_types.views.ProductTypeListView (**kwargs)
    Bases: shoop.admin.utils.views.PicortableListView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

columns = [<shoop.admin.utils.picortable.Column object at 0x7f713c176a90>, <shoop.admin.utils.picortable.Column object at 0x7f713c176a90>]

get_queryset ()

model
    alias of ProductType

```

Module contents

```

class shoop.admin.modules.product_types.ProductTypeModule
    Bases: shoop.admin.base.AdminModule

    name = <django.utils.functional.lazy.<locals>.__proxy__ object>

    breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>

    get_urls ()

    get_menu_entries (request)

    get_required_permissions ()

    get_model_url (object, kind)

```

shoop.admin.modules.products package

Subpackages

shoop.admin.modules.products.forms package

Submodules

shoop.admin.modules.products.forms.base_forms module

```

class shoop.admin.modules.products.forms.base_forms.ProductBaseForm (**kwargs)
    Bases: shoop.utils.multilanguage_model_form.MultiLanguageModelForm

    class Meta
        Bases: object

        model
            alias of Product

        fields = ('accounting_identifier', 'barcode', 'category', 'cost_center', 'depth', 'gross_weight', 'gtin', 'height', 'manufacturer', 'material', 'name', 'parent', 'product_type', 'sku', 'stock', 'unit_of_measure', 'weight')

        widgets = {'keywords': <django.forms.widgets.TextInput object at 0x7f713bdec550>}

    ProductBaseForm.base_fields = OrderedDict([('accounting_identifier', <django.forms.fields.CharField object at 0x7f713bdec550>), ('barcode', <django.forms.fields.CharField object at 0x7f713bdec550>), ('category', <django.forms.fields.CharField object at 0x7f713bdec550>), ('cost_center', <django.forms.fields.CharField object at 0x7f713bdec550>), ('depth', <django.forms.fields.CharField object at 0x7f713bdec550>), ('gross_weight', <django.forms.fields.DecimalField object at 0x7f713bdec550>), ('gtin', <django.forms.fields.CharField object at 0x7f713bdec550>), ('height', <django.forms.fields.DecimalField object at 0x7f713bdec550>), ('manufacturer', <django.forms.fields.CharField object at 0x7f713bdec550>), ('material', <django.forms.fields.CharField object at 0x7f713bdec550>), ('name', <django.forms.fields.CharField object at 0x7f713bdec550>), ('parent', <django.forms.fields.CharField object at 0x7f713bdec550>), ('product_type', <django.forms.fields.CharField object at 0x7f713bdec550>), ('sku', <django.forms.fields.CharField object at 0x7f713bdec550>), ('stock', <django.forms.fields.IntegerField object at 0x7f713bdec550>), ('unit_of_measure', <django.forms.fields.CharField object at 0x7f713bdec550>), ('weight', <django.forms.fields.DecimalField object at 0x7f713bdec550>)])

```

```
ProductBaseForm.declared_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713bd...
ProductBaseForm.media
class shoop.admin.modules.products.forms.base_forms.ShopProductForm (**kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        model
            alias of ShopProduct

        fields = ('default_price_value', 'minimum_price_value', 'suppliers', 'visible', 'listed', 'purchasable', 'searchable',

ShopProductForm.clean_minimum_purchase_quantity ()

ShopProductForm.base_fields = OrderedDict([('default_price_value', <shoop.core.fields.FormattedDecimalForm

ShopProductForm.declared_fields = OrderedDict()

ShopProductForm.media

class shoop.admin.modules.products.forms.base_forms.ProductAttributesForm (**kwargs)
    Bases: django.forms.forms.Form

    save ()

    base_fields = OrderedDict()

    declared_fields = OrderedDict()

    media

class shoop.admin.modules.products.forms.base_forms.BaseProductMediaForm (**kwargs)
    Bases: shoop.utils.multilanguage_model_form.MultiLanguageModelForm

    class Meta
        Bases: object

        model
            alias of ProductMedia

        fields = ('file', 'ordering', 'external_url', 'public', 'title', 'description', 'purchased', 'shops', 'kind')

BaseProductMediaForm.get_thumbnail (request)
    Get thumbnail url.

    If thumbnail creation fails for whatever reason, an error message is displayed for user.

BaseProductMediaForm.pre_master_save (instance)

BaseProductMediaForm.base_fields = OrderedDict([('file', <filer.fields.file.AdminFileFormField object at 0x7f71

BaseProductMediaForm.declared_fields = OrderedDict([('title', <django.forms.fields.CharField object at 0x7f7

BaseProductMediaForm.media

class shoop.admin.modules.products.forms.base_forms.BaseProductMediaFormSet (*args,
                                                                              **kwargs)

    Bases: django.forms.models.BaseModelFormSet

    validate_min = False

    min_num = 0

    validate_max = False

    max_num = 1000
```



```

    absolute_max = 1000

    model
        alias of ProductMedia

    can_delete = True

    can_order = False

    extra = 1

    allowed_media_kinds = []

    get_queryset ()

    form (**kwargs)

class shoop.admin.modules.products.forms.base_forms.ProductMediaForm (**kwargs)
    Bases: shoop.admin.modules.products.forms.base_forms.BaseProductMediaForm

    clean_external_url ()

    base_fields = OrderedDict([('file', <filer.fields.file.AdminFileFormField object at 0x7f713bdea518>), ('ordering', <dja

    declared_fields = OrderedDict([('title', <django.forms.fields.CharField object at 0x7f713bdec160>), ('description', <

    media

class shoop.admin.modules.products.forms.base_forms.ProductMediaFormSet (*args,
                                **kwargs)
    Bases: shoop.admin.modules.products.forms.base_forms.BaseProductMediaFormSet

    form_class
        alias of ProductMediaForm

    allowed_media_kinds = [<ProductMediaKind.GENERIC_FILE: 1>, <ProductMediaKind.DOCUMENTATION: 3>],

class shoop.admin.modules.products.forms.base_forms.ProductImageMediaForm (**kwargs)
    Bases: shoop.admin.modules.products.forms.base_forms.BaseProductMediaForm

    clean_file ()

    save (commit=True)

    base_fields = OrderedDict([('file', <filer.fields.file.AdminFileFormField object at 0x7f713c04d390>), ('ordering', <dja

    declared_fields = OrderedDict([('title', <django.forms.fields.CharField object at 0x7f713bdec160>), ('description', <

    media

class shoop.admin.modules.products.forms.base_forms.ProductImageMediaFormSet (*args,
                                **kwargs)
    Bases: shoop.admin.modules.products.forms.base_forms.ProductMediaFormSet

    allowed_media_kinds = [<ProductMediaKind.IMAGE: 2>]

    form_class
        alias of ProductImageMediaForm

    save (commit=True)
        Save the form.

        In addition add the first saved image as primary image for the product if none is selected as such.

```

shoop.admin.modules.products.forms.package_forms module

```
class shoop.admin.modules.products.forms.package_forms.PackageChildForm (data=None,
                                                                    files=None,
                                                                    auto_id='id_%s',
                                                                    pre-
                                                                    fix=None,
                                                                    ini-
                                                                    tial=None,
                                                                    er-
                                                                    ror_class=<class
                                                                    'django.forms.utils.ErrorList'>
                                                                    la-
                                                                    bel_suffix=None,
                                                                    empty_permitted=False)

    Bases: django.forms.forms.Form

    base_fields = OrderedDict([('child', <django.forms.models.ModelChoiceField object at 0x7f713be9fef0>), ('quantity',
    declared_fields = OrderedDict([('child', <django.forms.models.ModelChoiceField object at 0x7f713be9fef0>), ('qua
    media
class shoop.admin.modules.products.forms.package_forms.PackageChildFormSet (**kwargs)
    Bases: shoop.admin.modules.products.forms.parent_forms.ProductChildBaseFormSet

    deletion_label = <django.utils.functional.lazy.<locals>.__proxy__ object>

    save()

    get_selected_and_removed()
```

shoop.admin.modules.products.forms.parent_forms module

```
class shoop.admin.modules.products.forms.parent_forms.ProductChildBaseFormSet (**kwargs)
    Bases: django.forms.formsets.BaseFormSet

    deletion_label = None
```

shoop.admin.modules.products.forms.simple_variation_forms module

```
class shoop.admin.modules.products.forms.simple_variation_forms.SimpleVariationChildForm (data=
                                                                    files=
                                                                    auto_
                                                                    pre-
                                                                    fix=N
                                                                    ini-
                                                                    tial=
                                                                    er-
                                                                    ror_c
                                                                    'djan
                                                                    la-
                                                                    bel_s
                                                                    empty

    Bases: django.forms.forms.Form

    base_fields = OrderedDict([('child', <django.forms.models.ModelChoiceField object at 0x7f713be84710>)])
    declared_fields = OrderedDict([('child', <django.forms.models.ModelChoiceField object at 0x7f713be84710>)])
    media
```

```

class shoop.admin.modules.products.forms.simple_variation_forms.SimpleVariationChildFormSet (
    Bases: shoop.admin.modules.products.forms.parent_forms.ProductChildBaseFormSet

    deletion_label = <django.utils.functional.lazy.<locals>.__proxy__ object>

    save ()

    get_selected_and_unlinked ()

```

shoop.admin.modules.products.forms.variable_variation_forms module

```

class shoop.admin.modules.products.forms.variable_variation_forms.VariableVariationChildrenForm (
    Bases: django.forms.forms.Form

    save ()

    base_fields = OrderedDict()

    declared_fields = OrderedDict()

    media

class shoop.admin.modules.products.forms.variable_variation_forms.VariationVariablesDataForm (
    Bases: django.forms.forms.Form

    get_variable_data ()

    get_editor_args ()

    process_var_datum (var_datum)

    process_val_datum (var, val_datum)

    save ()

    base_fields = OrderedDict([('data', <django.forms.fields.CharField object at 0x7f713beecf28>)])

    declared_fields = OrderedDict([('data', <django.forms.fields.CharField object at 0x7f713beecf28>)])

    media

```

Module contents

```

class shoop.admin.modules.products.forms.BaseProductMediaForm (**kwargs)
    Bases: shoop.utils.multilanguage_model_form.MultiLanguageModelForm

    class Meta
        Bases: object

        fields = ('file', 'ordering', 'external_url', 'public', 'title', 'description', 'purchased', 'shops', 'kind')

        model
            alias of ProductMedia

    BaseProductMediaForm.base_fields = OrderedDict([('file', <filer.fields.file.AdminFileFormField object at 0x7f713beecf28>)])

    BaseProductMediaForm.declared_fields = OrderedDict([('title', <django.forms.fields.CharField object at 0x7f713beecf28>)])

    BaseProductMediaForm.get_thumbnail (request)
        Get thumbnail url.

        If thumbnail creation fails for whatever reason, an error message is displayed for user.

    BaseProductMediaForm.media

```

```
BaseProductMediaForm.pre_master_save(instance)
class shoop.admin.modules.products.forms.PackageChildForm(data=None, files=None,
                                                         auto_id='id_%s', pre-
                                                         fix=None, initial=None,
                                                         error_class=<class
                                                         'django.forms.utils.ErrorList'>,
                                                         label_suffix=None,
                                                         empty_permitted=False)

Bases: django.forms.forms.Form

base_fields = OrderedDict([('child', <django.forms.models.ModelChoiceField object at 0x7f713be9fef0>), ('quantity',
declared_fields = OrderedDict([('child', <django.forms.models.ModelChoiceField object at 0x7f713be9fef0>), ('qua
media

class shoop.admin.modules.products.forms.PackageChildFormSet(**kwargs)
Bases: shoop.admin.modules.products.forms.parent_forms.ProductChildBaseFormSet

deletion_label = <django.utils.functional.lazy.<locals>.__proxy__ object>
get_selected_and_removed()
save()

class shoop.admin.modules.products.forms.ProductAttributesForm(**kwargs)
Bases: django.forms.forms.Form

base_fields = OrderedDict()
declared_fields = OrderedDict()
media

save()

class shoop.admin.modules.products.forms.ProductBaseForm(**kwargs)
Bases: shoop.utils.multilanguage_model_form.MultiLanguageModelForm

class Meta
    Bases: object

    fields = ('accounting_identifier', 'barcode', 'category', 'cost_center', 'depth', 'gross_weight', 'gtin', 'height', 'mar
    model
        alias of Product

    widgets = {'keywords': <django.forms.widgets.TextInput object at 0x7f713bdec550>}

ProductBaseForm.base_fields = OrderedDict([('accounting_identifier', <django.forms.fields.CharField object at 0
ProductBaseForm.declared_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713bd
ProductBaseForm.media

class shoop.admin.modules.products.forms.ProductImageMediaForm(**kwargs)
Bases: shoop.admin.modules.products.forms.base_forms.BaseProductMediaForm

base_fields = OrderedDict([('file', <filer.fields.file.AdminFileFormField object at 0x7f713c04d390>), ('ordering', <dja
clean_file()

declared_fields = OrderedDict([('title', <django.forms.fields.CharField object at 0x7f713bdec160>), ('description', <
media
```

```

    save (commit=True)

class shoop.admin.modules.products.forms.ProductImageMediaFormSet (*args,
                                                                    **kwargs)
    Bases: shoop.admin.modules.products.forms.base_forms.ProductMediaFormSet
    allowed_media_kinds = [<ProductMediaKind.IMAGE: 2>]

    form_class
        alias of ProductImageMediaForm

    save (commit=True)
        Save the form.

        In addition add the first saved image as primary image for the product if none is selected as such.

class shoop.admin.modules.products.forms.ProductMediaForm (**kwargs)
    Bases: shoop.admin.modules.products.forms.base_forms.BaseProductMediaForm
    base_fields = OrderedDict([('file', <filer.fields.file.AdminFileFormField object at 0x7f713bdea518>), ('ordering', <dja
    clean_external_url ()
    declared_fields = OrderedDict([('title', <django.forms.fields.CharField object at 0x7f713bdec160>), ('description', <
    media

class shoop.admin.modules.products.forms.ProductMediaFormSet (*args, **kwargs)
    Bases: shoop.admin.modules.products.forms.base_forms.BaseProductMediaFormSet

    allowed_media_kinds = [<ProductMediaKind.GENERIC_FILE: 1>, <ProductMediaKind.DOCUMENTATION: 3>],

    form_class
        alias of ProductMediaForm

class shoop.admin.modules.products.forms.ShopProductForm (**kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object
        fields = ('default_price_value', 'minimum_price_value', 'suppliers', 'visible', 'listed', 'purchasable', 'searchable',
        model
            alias of ShopProduct

ShopProductForm.base_fields = OrderedDict([('default_price_value', <shoop.core.fields.FormattedDecimalForm
ShopProductForm.clean_minimum_purchase_quantity ()
ShopProductForm.declared_fields = OrderedDict()
ShopProductForm.media

```

```
class shoop.admin.modules.products.forms.SimpleVariationChildForm (data=None,
                                                                    files=None,
                                                                    auto_id='id_%s',
                                                                    pre-
                                                                    fix=None, ini-
                                                                    tial=None, er-
                                                                    ror_class=<class
                                                                    'django.forms.utils.ErrorList'>,
                                                                    la-
                                                                    bel_suffix=None,
                                                                    empty_permitted=False)

    Bases: django.forms.forms.Form

    base_fields = OrderedDict([('child', <django.forms.models.ModelChoiceField object at 0x7f713be84710>)])
    declared_fields = OrderedDict([('child', <django.forms.models.ModelChoiceField object at 0x7f713be84710>)])
    media

class shoop.admin.modules.products.forms.SimpleVariationChildFormSet (**kwargs)
    Bases: shoop.admin.modules.products.forms.parent_forms.ProductChildBaseFormSet

    deletion_label = <django.utils.functional.lazy.<locals>.__proxy__ object>
    get_selected_and_unlinked()
    save()

class shoop.admin.modules.products.forms.VariableVariationChildrenForm (**kwargs)
    Bases: django.forms.forms.Form

    base_fields = OrderedDict()
    declared_fields = OrderedDict()
    media
    save()

class shoop.admin.modules.products.forms.VariationVariablesDataForm (**kwargs)
    Bases: django.forms.forms.Form

    base_fields = OrderedDict([('data', <django.forms.fields.CharField object at 0x7f713beecf28>)])
    declared_fields = OrderedDict([('data', <django.forms.fields.CharField object at 0x7f713beecf28>)])
    get_editor_args()
    get_variable_data()
    media
    process_val_datum (var, val_datum)

    process_var_datum (var_datum)
    save()
```

shoop.admin.modules.products.views package

Submodules

shoop.admin.modules.products.views.delete module

```
class shoop.admin.modules.products.views.delete.ProductDeleteView (**kwargs)
    Bases: django.views.generic.detail.DetailView
```

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
model
```

```
    alias of Product
```

```
context_object_name = 'product'
```

```
get (request, *args, **kwargs)
```

```
post (request, *args, **kwargs)
```

shoop.admin.modules.products.views.edit module

```
class shoop.admin.modules.products.views.edit.ProductBaseFormPart (request, ob-
                                                                    ject=None)
```

```
    Bases: shoop.admin.form_part.FormPart
```

```
priority = -1000
```

```
get_form_defs ()
```

```
form_valid (form)
```

```
get_initial ()
```

```
class shoop.admin.modules.products.views.edit.ShopProductFormPart (request, ob-
                                                                    ject=None)
```

```
    Bases: shoop.admin.form_part.FormPart
```

```
priority = -900
```

```
get_shop_instance (shop)
```

```
get_form_defs ()
```

```
form_valid (form)
```

```
class shoop.admin.modules.products.views.edit.ProductAttributeFormPart (request,
                                                                    ob-
                                                                    ject=None)
```

```
    Bases: shoop.admin.form_part.FormPart
```

```
priority = -800
```

```
get_form_defs ()
```

```
form_valid (form)
```

```
class shoop.admin.modules.products.views.edit.BaseProductMediaFormPart (request,
                                                                    ob-
                                                                    ject=None)
```

```
    Bases: shoop.admin.form_part.FormPart
```

```
get_form_defs ()
```

```
form_valid (form)
```

```
class shoop.admin.modules.products.views.edit.ProductMediaFormPart (request, ob-
                                                                    ject=None)
```

```
    Bases: shoop.admin.modules.products.views.edit.BaseProductMediaFormPart
```

```
name = 'media'
```

```
priority = -700
```

```
formset
    alias of ProductMediaFormSet

class shoop.admin.modules.products.views.edit.ProductImageMediaFormPart (request,
                                                                    ob-
                                                                    ject=None)
    Bases: shoop.admin.modules.products.views.edit.BaseProductMediaFormPart
    name = 'images'
    priority = -600
    formset
        alias of ProductImageMediaFormSet

class shoop.admin.modules.products.views.edit.ProductEditView (**kwargs)
    Bases: shoop.admin.form_part.SaveFormPartsMixin, shoop.admin.form_part.FormPartsViewMixin,
shoop.admin.utils.views.CreateOrUpdateView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of Product

    template_name = 'shoop/admin/products/edit.jinja'
    context_object_name = 'product'
    base_form_part_classes = [<class 'shoop.admin.modules.products.views.edit.ProductBaseFormPart'>, <class 'shoop
    form_part_class_provide_key = 'admin_product_form_part'
    form_valid (form)
    get_toolbar ()
    get_context_data (**kwargs)

shoop.admin.modules.products.views.edit_cross_sell module
class shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSellForm (**kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        model
            alias of ProductCrossSell

        fields = ('product2', 'weight', 'type')
    ProductCrossSellForm.save (commit=True)
    ProductCrossSellForm.base_fields = OrderedDict([('product2', <django.forms.models.ModelChoiceField object
    ProductCrossSellForm.declared_fields = OrderedDict()
    ProductCrossSellForm.media
class shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSellFormSet (*args,
                                                                    **kwargs)
    Bases: django.forms.models.BaseModelFormSet
    validate_min = False
    min_num = 0
    validate_max = False
```



```

    max_num = 1000
    absolute_max = 1000
    model
        alias of ProductMedia
    can_delete = True
    can_order = False
    extra = 5
    form ( **kwargs )

class shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSellEditView ( **kwargs )
    Bases: django.views.generic.edit.UpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of Product
    template_name = 'shoop/admin/products/edit_cross_sell.jinja'
    context_object_name = 'product'
    form_class
        alias of ProductCrossSellFormSet
    get_breadcrumb_parents ( )
    get_context_data ( **kwargs )
    get_form_kwargs ( )
    form_valid ( form )

shoop.admin.modules.products.views.edit_media module
class shoop.admin.modules.products.views.edit_media.ProductMediaForm ( **kwargs )
    Bases: shoop.utils.multilanguage_model_form.MultiLanguageModelForm

    class Meta
        Bases: object

        model
            alias of ProductMedia

        fields = ('shops', 'kind', 'file', 'external_url', 'ordering', 'enabled', 'public', 'purchased', 'title', 'description')

    ProductMediaForm.pre_master_save ( instance )

    ProductMediaForm.base_fields = OrderedDict([('shops', <django.forms.models.ModelMultipleChoiceField object
    ProductMediaForm.declared_fields = OrderedDict([('title', <django.forms.fields.CharField object at 0x7f713bd5
    ProductMediaForm.media

class shoop.admin.modules.products.views.edit_media.ProductMediaFormSet ( *args,
                                                                    **kwargs )
    Bases: django.forms.models.BaseModelFormSet

    validate_min = False
    min_num = 0
    validate_max = False

```

```
max_num = 1000
absolute_max = 1000
model
    alias of ProductMedia
can_delete = True
can_order = False
extra = 5
form ( **kwargs )
```

class `shoop.admin.modules.products.views.edit_media.ProductMediaEditView (**kwargs)`
Bases: `django.views.generic.edit.UpdateView`

A view for editing all the media for a product, including attachments that are not just images.
Currently sort of utilitarian and confusing.
Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
model
    alias of Product
template_name = 'shoop/admin/products/edit_media.jinja'
context_object_name = 'product'
form_class
    alias of ProductMediaFormSet
get_breadcrumb_parents ( )
get_context_data ( **kwargs )
get_form_kwargs ( )
form_valid ( form )
```

shoop.admin.modules.products.views.edit_package module

```
class shoop.admin.modules.products.views.edit_package.ProductChildrenFormPart ( request,
                                                                                   ob-
                                                                                   ject=None )
Bases: shoop.admin.modules.products.views.edit_parent.ProductChildrenBaseFormPart

invalid_modes = [<ProductMode.VARIATION_CHILD: 4>, <ProductMode.VARIABLE_VARIATION_PARENT: 3>]
priority = 0
get_form_defs ( )
```

class `shoop.admin.modules.products.views.edit_package.ProductPackageViewToolbar (view)`
Bases: `shoop.admin.modules.products.views.edit_parent.ProductParentBaseToolbar`

class `shoop.admin.modules.products.views.edit_package.ProductPackageView (**kwargs)`
Bases: `shoop.admin.modules.products.views.edit_parent.ProductParentBaseView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
template_name = 'shoop/admin/products/package/edit.jinja'
form_part_classes = [<class 'shoop.admin.modules.products.views.edit_package.ProductChildrenFormPart'>]
```

```

toolbar_class
    alias of ProductPackageViewToolbar

dispatch (request, *args, **kwargs)

get_context_data (**kwargs)

dispatch_command (request, command)

```

shoop.admin.modules.products.views.edit_parent module

```

class shoop.admin.modules.products.views.edit_parent.ProductChildrenBaseFormPart (request,
                                                                                   ob-
                                                                                   ject=None)

    Bases: shoop.admin.form_part.FormPart

    invalid_modes = []

    priority = 0

    form_name = None

    get_form_defs (form, template_name)

    form_valid (form)

class shoop.admin.modules.products.views.edit_parent.ProductParentBaseToolbar (view)
    Bases: shoop.admin.toolbar.Toolbar

class shoop.admin.modules.products.views.edit_parent.ProductParentBaseView (**kwargs)
    Bases: shoop.admin.form_part.FormPartsViewMixin, django.views.generic.edit.UpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of Product

    context_object_name = 'product'

    form_class
        alias of Form

    form_part_classes = []

    toolbar_class = None

    dispatch (request, *args, **kwargs)

    get_breadcrumb_parents ()

    post (request, *args, **kwargs)

    get_form_part_classes ()

    get_context_data (**kwargs)

    form_valid (form)

    get_success_url ()

    dispatch_command (request, command)

```

shoop.admin.modules.products.views.edit_variation module

```
class shoop.admin.modules.products.views.edit_variation.VariationChildrenFormPart (request,
                                                                                      ob-
                                                                                      ject=None)

    Bases: shoop.admin.modules.products.views.edit_parent.ProductChildrenBaseFormPart

    invalid_modes = [<ProductMode.VARIATION_CHILD: 4>, <ProductMode.PACKAGE_PARENT: 1>]
    priority = 0
    get_form_defs ()

class shoop.admin.modules.products.views.edit_variation.VariationVariablesFormPart (request,
                                                                                      ob-
                                                                                      ject=None)

    Bases: shoop.admin.form_part.FormPart

    form_def_name = 'variables'
    priority = 1
    get_form_defs ()
    form_valid (form)

class shoop.admin.modules.products.views.edit_variation.ProductVariationViewToolbar (view)
    Bases: shoop.admin.modules.products.views.edit_parent.ProductParentBaseToolbar

class shoop.admin.modules.products.views.edit_variation.ProductVariationView (**kwargs)
    Bases: shoop.admin.modules.products.views.edit_parent.ProductParentBaseView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    template_name = 'shoop/admin/products/variation/edit.jinja'
    form_part_classes = [<class 'shoop.admin.modules.products.views.edit_variation.VariationChildrenFormPart'>, <class 'shoop.admin.modules.products.views.edit_variation.VariationVariablesFormPart'>]
    toolbar_class
        alias of ProductVariationViewToolbar
    dispatch (request, *args, **kwargs)
    get_context_data (**kwargs)
    dispatch_command (request, command)
```

shoop.admin.modules.products.views.list module

```
class shoop.admin.modules.products.views.list.ProductListView (**kwargs)
    Bases: shoop.admin.utils.views.PicotableListView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of Product
    columns = [<shoop.admin.utils.picotable.Column object at 0x7f713bd059e8>, <shoop.admin.utils.picotable.Column object at 0x7f713bd059e8>]
    get_queryset ()
    get_object_abstract (instance, item)
```

shoop.admin.modules.products.views.toolbars module

```
class shoop.admin.modules.products.views.toolbars.EditProductToolbar (view)
    Bases: shoop.admin.toolbar.Toolbar
```

Module contents

class `shoop.admin.modules.products.views.ProductCrossSellEditView` (***kwargs*)
 Bases: `django.views.generic.edit.UpdateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

context_object_name = 'product'

form_class

alias of `ProductCrossSellFormSet`

form_valid (*form*)

get_breadcrumb_parents ()

get_context_data (***kwargs*)

get_form_kwargs ()

model

alias of `Product`

template_name = 'shoop/admin/products/edit_cross_sell.jinja'

class `shoop.admin.modules.products.views.ProductDeleteView` (***kwargs*)

Bases: `django.views.generic.detail.DetailView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

context_object_name = 'product'

get (*request, *args, **kwargs*)

model

alias of `Product`

post (*request, *args, **kwargs*)

class `shoop.admin.modules.products.views.ProductEditView` (***kwargs*)

Bases: `shoop.admin.form_part.SaveFormPartsMixin`, `shoop.admin.form_part.FormPartsViewMixin`,
`shoop.admin.utils.views.CreateOrUpdateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

base_form_part_classes = [`<class 'shoop.admin.modules.products.views.edit.ProductBaseFormPart'>`, `<class 'shoop.admin.modules.products.views.edit.ProductFormPart'>`]

context_object_name = 'product'

form_part_class_provide_key = 'admin_product_form_part'

form_valid (*form*)

get_context_data (***kwargs*)

get_toolbar ()

model

alias of `Product`

template_name = 'shoop/admin/products/edit.jinja'

class `shoop.admin.modules.products.views.ProductListView` (***kwargs*)

Bases: `shoop.admin.utils.views.PicortableListView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

columns = [`<shoop.admin.utils.picortable.Column object at 0x7f713bd059e8>`, `<shoop.admin.utils.picortable.Column object at 0x7f713bd059e8>`]

get_object_abstract (*instance, item*)

```
get_queryset ()

model
    alias of Product

class shoop.admin.modules.products.views.ProductPackageView (**kwargs)
    Bases: shoop.admin.modules.products.views.edit_parent.ProductParentBaseView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    dispatch (request, *args, **kwargs)
    dispatch_command (request, command)
    form_part_classes = [<class 'shoop.admin.modules.products.views.edit_package.ProductChildrenFormPart'>]
    get_context_data (**kwargs)
    template_name = 'shoop/admin/products/package/edit.jinja'
    toolbar_class
        alias of ProductPackageViewToolbar

class shoop.admin.modules.products.views.ProductVariationView (**kwargs)
    Bases: shoop.admin.modules.products.views.edit_parent.ProductParentBaseView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    dispatch (request, *args, **kwargs)
    dispatch_command (request, command)
    form_part_classes = [<class 'shoop.admin.modules.products.views.edit_variation.VariationChildrenFormPart'>, <class 'shoop.admin.modules.products.views.edit_variation.VariationFormPart'>]
    get_context_data (**kwargs)
    template_name = 'shoop/admin/products/variation/edit.jinja'
    toolbar_class
        alias of ProductVariationViewToolbar

class shoop.admin.modules.products.views.ProductMediaEditView (**kwargs)
    Bases: django.views.generic.edit.UpdateView
    A view for editing all the media for a product, including attachments that are not just images.
    Currently sort of utilitarian and confusing.
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    context_object_name = 'product'
    form_class
        alias of ProductMediaFormSet
    form_valid (form)
    get_breadcrumb_parents ()
    get_context_data (**kwargs)
    get_form_kwargs ()
    model
        alias of Product
    template_name = 'shoop/admin/products/edit_media.jinja'
```

Submodules

shoop.admin.modules.products.utils module

shoop.admin.modules.products.utils.**clear_existing_package** (*parent_product*)
 Utility function for clearing existing package.

Module contents

```
class shoop.admin.modules.products.ProductModule
    Bases: shoop.admin.base.AdminModule

    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>
    get_urls ()
    get_menu_category_icons ()
    get_menu_entries (request)
    get_search_results (request, query)
    get_required_permissions ()
    get_model_url (object, kind)
```

shoop.admin.modules.sales_units package

Subpackages

shoop.admin.modules.sales_units.views package

Submodules

shoop.admin.modules.sales_units.views.edit module

```
class shoop.admin.modules.sales_units.views.edit.SalesUnitForm (**kwargs)
    Bases: shoop.utils.multilanguage_model_form.MultiLanguageModelForm

    class Meta
        Bases: object

        model
            alias of SalesUnit

        exclude = ()

    SalesUnitForm.base_fields = OrderedDict([('decimals', <django.forms.fields.IntegerField object at 0x7f713ba292...
    SalesUnitForm.declared_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713be0ea...
    SalesUnitForm.media
class shoop.admin.modules.sales_units.views.edit.SalesUnitEditView (**kwargs)
    Bases: shoop.admin.utils.views.CreateOrUpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
```

```
model
    alias of SalesUnit

form_class
    alias of SalesUnitForm

template_name = 'shoop/admin/sales_units/edit.jinja'

context_object_name = 'sales_unit'
```

shoop.admin.modules.sales_units.views.list module

```
class shoop.admin.modules.sales_units.views.list.SalesUnitListView(**kwargs)
    Bases: shoop.admin.utils.views.PicortableListView
```

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
model
    alias of SalesUnit

columns = [<shoop.admin.utils.picortable.Column object at 0x7f713ba2e7b8>, <shoop.admin.utils.picortable.Column object at 0x7f713ba2e7b8>]

get_queryset()
```

Module contents

```
class shoop.admin.modules.sales_units.views.SalesUnitEditView(**kwargs)
    Bases: shoop.admin.utils.views.CreateOrUpdateView
```

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
context_object_name = 'sales_unit'
```

```
form_class
    alias of SalesUnitForm
```

```
model
    alias of SalesUnit
```

```
template_name = 'shoop/admin/sales_units/edit.jinja'
```

```
class shoop.admin.modules.sales_units.views.SalesUnitListView(**kwargs)
    Bases: shoop.admin.utils.views.PicortableListView
```

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
columns = [<shoop.admin.utils.picortable.Column object at 0x7f713ba2e7b8>, <shoop.admin.utils.picortable.Column object at 0x7f713ba2e7b8>]
```

```
get_queryset()
```

```
model
    alias of SalesUnit
```

Module contents

```
class shoop.admin.modules.sales_units.SalesUnitModule
    Bases: shoop.admin.base.AdminModule
```

```
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
```

```
breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>
```

```
get_urls()
```

```
get_menu_entries(request)
```



```
get_required_permissions()
```

```
get_model_url(object, kind)
```

shoop.admin.modules.service_providers package

Subpackages

shoop.admin.modules.service_providers.views package

Module contents

```
class shoop.admin.modules.service_providers.views.ServiceProviderDeleteView(**kwargs)
    Bases: django.views.generic.edit.DeleteView
```

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
model
```

```
    alias of ServiceProvider
```

```
success_url = <django.utils.functional.lazy.<locals>.__proxy__ object>
```

```
class shoop.admin.modules.service_providers.views.ServiceProviderEditView(**kwargs)
    Bases: shoop.admin.utils.views.CreateOrUpdateView
```

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
add_form_errors_as_messages = True
```

```
context_object_name = 'service_provider'
```

```
form_class
```

```
    alias of Form
```

```
form_provide_key = 'service_provider_admin_form'
```

```
get_breadcrumb_parents()
```

```
get_form(form_class=None)
```

```
get_success_url()
```

```
get_toolbar()
```

```
model
```

```
    alias of ServiceProvider
```

```
template_name = 'shoop/admin/service_providers/edit.jinja'
```

```
title
```

```
class shoop.admin.modules.service_providers.views.ServiceProviderListView(**kwargs)
    Bases: shoop.admin.utils.views.PicortableListView
```

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
columns = [<shoop.admin.utils.picortable.Column object at 0x7f713b9c09b0>, <shoop.admin.utils.picortable.Column object at 0x7f713b9c09b0>]
```

```
get_object_abstract(instance, item)
```

```
get_type_display(instance)
```

```
model
```

```
    alias of ServiceProvider
```

Submodules

shoop.admin.modules.service_providers.forms module

class shoop.admin.modules.service_providers.forms.**CustomCarrierForm**(**kwargs)

Bases: shoop.admin.forms._base.ShoopAdminForm

class **Meta**

Bases: *object*

model

alias of CustomCarrier

exclude = ('identifier',)

CustomCarrierForm.**base_fields** = OrderedDict([('enabled', <django.forms.fields.BooleanField object at 0x7f713...

CustomCarrierForm.**declared_fields** = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713...

CustomCarrierForm.**media**

class shoop.admin.modules.service_providers.forms.**CustomPaymentProcessorForm**(**kwargs)

Bases: shoop.admin.forms._base.ShoopAdminForm

class **Meta**

Bases: *object*

model

alias of CustomPaymentProcessor

exclude = ('identifier',)

CustomPaymentProcessorForm.**base_fields** = OrderedDict([('enabled', <django.forms.fields.BooleanField obje...

CustomPaymentProcessorForm.**declared_fields** = OrderedDict([('name', <django.forms.fields.CharField obje...

CustomPaymentProcessorForm.**media**

Module contents

class shoop.admin.modules.service_providers.**ServiceProviderModule**

Bases: *shoop.admin.base.AdminModule*

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

category = <django.utils.functional.lazy.<locals>.__proxy__ object>

get_urls()

get_menu_category_icons()

get_menu_entries(request)

get_required_permissions()

get_model_url(object, kind)

shoop.admin.modules.services package

Subpackages

shoop.admin.modules.services.views package

Module contents

class `shoop.admin.modules.services.views.PaymentMethodDeleteView` (***kwargs*)
 Bases: `django.views.generic.edit.DeleteView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of `PaymentMethod`

success_url = `<django.utils.functional.lazy.<locals>.__proxy__ object>`

class `shoop.admin.modules.services.views.PaymentMethodEditView` (***kwargs*)
 Bases: `shoop.admin.modules.services.views._edit.ServiceEditView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

base_form_part_classes = [`<class 'shoop.admin.modules.services.base_form_part.PaymentMethodBaseFormPart'`]

model

alias of `PaymentMethod`

class `shoop.admin.modules.services.views.PaymentMethodListView` (***kwargs*)
 Bases: `shoop.admin.modules.services.views._list.ServiceListView`

model

alias of `PaymentMethod`

class `shoop.admin.modules.services.views.ShippingMethodDeleteView` (***kwargs*)
 Bases: `django.views.generic.edit.DeleteView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of `ShippingMethod`

success_url = `<django.utils.functional.lazy.<locals>.__proxy__ object>`

class `shoop.admin.modules.services.views.ShippingMethodEditView` (***kwargs*)
 Bases: `shoop.admin.modules.services.views._edit.ServiceEditView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

base_form_part_classes = [`<class 'shoop.admin.modules.services.base_form_part.ShippingMethodBaseFormPart'`]

model

alias of `ShippingMethod`

class `shoop.admin.modules.services.views.ShippingMethodListView` (***kwargs*)
 Bases: `shoop.admin.modules.services.views._list.ServiceListView`

model

alias of `ShippingMethod`

shoop.admin.modules.services.weight_based_pricing package**Module contents**

class `shoop.admin.modules.services.weight_based_pricing.WeightBasedPricingFormPart` (*request, object*)
 Bases: `shoop.admin.form_part.FormPart`

form_valid (*form*)

```
formset
    alias of WeightBasedPriceRangeFormFormSet
get_form_defs()
name = 'weight_based_price_ranges'
```

Submodules

shoop.admin.modules.services.base_form_part module

```
class shoop.admin.modules.services.base_form_part.ServiceBaseFormPart (*args,
                                                                    **kwargs)
    Bases: shoop.admin.form_part.FormPart
priority = -1000
form = None
get_form_defs()
form_valid(form)
class shoop.admin.modules.services.base_form_part.ShippingMethodBaseFormPart (*args,
                                                                    **kwargs)
    Bases: shoop.admin.modules.services.base_form_part.ServiceBaseFormPart
form
    alias of ShippingMethodForm
class shoop.admin.modules.services.base_form_part.PaymentMethodBaseFormPart (*args,
                                                                    **kwargs)
    Bases: shoop.admin.modules.services.base_form_part.ServiceBaseFormPart
form
    alias of PaymentMethodForm
```

shoop.admin.modules.services.behavior_form_part module

```
class shoop.admin.modules.services.behavior_form_part.BehaviorFormSet (*args,
                                                                    **kwargs)
    Bases: django.forms.models.BaseModelFormSet
model
    alias of ServiceBehaviorComponent
validate_min = False
min_num = 0
validate_max = False
max_num = 20
absolute_max = 20
can_delete = True
can_order = False
extra = 0
form_class = None
get_name()
```

```

    get_queryset ()

    form (**kwargs)
class shoop.admin.modules.services.behavior_form_part.BehaviorComponentFormPart (request,
                                                                 form,
                                                                 name,
                                                                 owner)

Bases: shoop.admin.form_part.FormPart

formset
    alias of BehaviorFormSet

template_name = 'shoop/admin/services/_edit_behavior_components_form.jinja'

get_form_defs ()

form_valid (form)

```

shoop.admin.modules.services.forms module

```

class shoop.admin.modules.services.forms.BaseMethodForm (**kwargs)
    Bases: shoop.admin.forms._base.ShoopAdminForm

    class Meta
        Bases: object

        model = None

        exclude = ['identifier', 'behavior_components', 'old_module_identifier', 'old_module_data']

        base_fields = ['choice_identifier', 'name', 'description', 'enabled', 'shop', 'logo', 'tax_class']

    BaseMethodForm.get_service_provider (id)

    BaseMethodForm.service_provider

    BaseMethodForm.base_fields = OrderedDict()

    BaseMethodForm.declared_fields = OrderedDict()

    BaseMethodForm.media

class shoop.admin.modules.services.forms.AlwaysChangedModelForm (data=None,
                                                                 files=None,
                                                                 auto_id='id_%s',
                                                                 prefix=None,
                                                                 initial=None, error_class=<class
                                                                 'django.forms.utils.ErrorList'>,
                                                                 label_suffix=None,
                                                                 empty_permitted=False,
                                                                 instance=None)

    Bases: django.forms.models.ModelForm

    ModelForm that can be saved if it is empty or has unchanged lines on creation

    has_changed (*args, **kwargs)

    base_fields = OrderedDict()

    declared_fields = OrderedDict()

    media

```

```
class shoop.admin.modules.services.forms.ShippingMethodForm(**kwargs)
    Bases: shoop.admin.modules.services.forms.BaseMethodForm

    service_provider_attr = 'carrier'

    class Meta
        Bases: shoop.admin.modules.services.forms.Meta

        model
            alias of ShippingMethod

        fields = ['carrier', 'choice_identifier', 'name', 'description', 'enabled', 'shop', 'logo', 'tax_class']

        help_texts = {'carrier': <django.utils.functional.lazy.<locals>.__proxy__ object at 0x7f713c3740b8>}

ShippingMethodForm.base_fields = OrderedDict([('carrier', <django.forms.models.ModelChoiceField object at 0x7f713c3740b8>)]
ShippingMethodForm.declared_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713c3740b8>)]
ShippingMethodForm.media

class shoop.admin.modules.services.forms.PaymentMethodForm(**kwargs)
    Bases: shoop.admin.modules.services.forms.BaseMethodForm

    service_provider_attr = 'payment_processor'

    class Meta
        Bases: shoop.admin.modules.services.forms.Meta

        model
            alias of PaymentMethod

        fields = ['payment_processor', 'choice_identifier', 'name', 'description', 'enabled', 'shop', 'logo', 'tax_class']

        help_texts = {'payment_processor': <django.utils.functional.lazy.<locals>.__proxy__ object at 0x7f713bff5080>}

PaymentMethodForm.base_fields = OrderedDict([('payment_processor', <django.forms.models.ModelChoiceField object at 0x7f713bff5080>)]
PaymentMethodForm.declared_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713bff5080>)]
PaymentMethodForm.media

class shoop.admin.modules.services.forms.FixedCostBehaviorComponentForm(**kwargs)
    Bases: shoop.admin.forms._base.ShoopAdminForm

    class Meta
        Bases: object

        model
            alias of FixedCostBehaviorComponent

        exclude = ['identifier']

        labels = {'price_value': <django.utils.functional.lazy.<locals>.__proxy__ object at 0x7f713be4a390>}

FixedCostBehaviorComponentForm.base_fields = OrderedDict([('price_value', <shoop.core.fields.FormattedDecimalField object at 0x7f713be4a390>)]
FixedCostBehaviorComponentForm.declared_fields = OrderedDict([('description', <django.forms.fields.CharField object at 0x7f713be4a390>)]
FixedCostBehaviorComponentForm.media

class shoop.admin.modules.services.forms.WaivingCostBehaviorComponentForm(**kwargs)
    Bases: shoop.admin.forms._base.ShoopAdminForm

    class Meta
        Bases: object
```

2.10. Shoop API Documentation 99

```
class Meta
    Bases: object

    model
        alias of GroupAvailabilityBehaviorComponent

    exclude = ['identifier']

GroupAvailabilityBehaviorComponentForm.base_fields = OrderedDict([('groups', <django.forms.models.M
GroupAvailabilityBehaviorComponentForm.declared_fields = OrderedDict()
GroupAvailabilityBehaviorComponentForm.media

class shoop.admin.modules.services.forms.StaffOnlyBehaviorComponentForm (data=None,
                                                                    files=None,
                                                                    auto_id='id_%s',
                                                                    pre-
                                                                    fix=None,
                                                                    ini-
                                                                    tial=None,
                                                                    er-
                                                                    ror_class=<class
                                                                    'django.forms.utils.ErrorList'>
                                                                    la-
                                                                    bel_suffix=None,
                                                                    empty_permitted=False,
                                                                    in-
                                                                    stance=None)

Bases: shoop.admin.modules.services.forms.AlwaysChangedModelForm

class Meta
    Bases: object

    model
        alias of StaffOnlyBehaviorComponent

    exclude = ['identifier']

StaffOnlyBehaviorComponentForm.base_fields = OrderedDict()
StaffOnlyBehaviorComponentForm.declared_fields = OrderedDict()
StaffOnlyBehaviorComponentForm.media

class shoop.admin.modules.services.forms.RoundingBehaviorComponentForm (data=None,
                                                                    files=None,
                                                                    auto_id='id_%s',
                                                                    pre-
                                                                    fix=None,
                                                                    ini-
                                                                    tial=None,
                                                                    er-
                                                                    ror_class=<class
                                                                    'django.forms.utils.ErrorList'>,
                                                                    la-
                                                                    bel_suffix=None,
                                                                    empty_permitted=False,
                                                                    in-
                                                                    stance=None)

Bases: shoop.admin.modules.services.forms.AlwaysChangedModelForm
```



```

class Meta
    Bases: object

    model
        alias of RoundingBehaviorComponent

    exclude = ['identifier']

RoundingBehaviorComponentForm.base_fields = OrderedDict([('quant', <django.forms.fields.DecimalField ob
RoundingBehaviorComponentForm.declared_fields = OrderedDict()
RoundingBehaviorComponentForm.media

```

Module contents

```

class shoop.admin.modules.services.ServiceModule
    Bases: shoop.admin.base.AdminModule

    category = <django.utils.functional.lazy.<locals>.__proxy__ object>

    model = None

    name = None

    url_prefix = None

    view_template = None

    name_template = None

    menu_entry_url = None

    url_name_prefix = None

    get_urls()

    get_menu_entries(request)

    get_required_permissions()

    get_model_url(object, kind)
class shoop.admin.modules.services.ShippingMethodModule
    Bases: shoop.admin.modules.services.ServiceModule

    model
        alias of ShippingMethod

    name = <django.utils.functional.lazy.<locals>.__proxy__ object>

    url_prefix = '^shipping_methods'

    view_template = 'shoop.admin.modules.services.views.ShippingMethod%sView'

    name_template = 'shipping_methods.%s'

    menu_entry_url = 'shoop_admin:shipping_methods.list'

    url_name_prefix = 'shoop_admin:shipping_methods'

    breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>

class shoop.admin.modules.services.PaymentMethodModule
    Bases: shoop.admin.modules.services.ServiceModule

    model
        alias of PaymentMethod

```

```
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
url_prefix = '^payment_methods'
view_template = 'shoop.admin.modules.services.views.PaymentMethod%sView'
name_template = 'payment_methods.%s'
menu_entry_url = 'shoop_admin:payment_methods.list'
url_name_prefix = 'shoop_admin:payment_methods'
breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>
```

shoop.admin.modules.shops package

Subpackages

shoop.admin.modules.shops.views package

Submodules

shoop.admin.modules.shops.views.edit module

```
class shoop.admin.modules.shops.views.edit.ShopBaseForm(**kwargs)
    Bases: shoop.core.utils.form_mixins.ProtectedFieldsMixin,
shoop.utils.multilanguage_model_form.MultiLanguageModelForm
    change_protect_field_text = <django.utils.functional.lazy.<locals>.__proxy__ object>
    class Meta
        Bases: object
        model
            alias of Shop
        exclude = ('owner', 'options', 'contact_address')
    ShopBaseForm.base_fields = OrderedDict([('domain', <django.forms.fields.CharField object at 0x7f713c2a9cc0>),
    ShopBaseForm.declared_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713d0f4660>),
    ShopBaseForm.media
class shoop.admin.modules.shops.views.edit.ShopBaseFormPart(request, object=None)
    Bases: shoop.admin.form_part.FormPart
    priority = 1
    get_form_defs()
    form_valid(form)
```

```

class shoop.admin.modules.shops.views.edit.ContactAddressForm (data=None,
                                                                files=None,
                                                                auto_id='id_%s',
                                                                prefix=None,    ini-
                                                                tial=None,      er-
                                                                ror_class=<class
                                                                'django.forms.utils.ErrorList'>,
                                                                label_suffix=None,
                                                                empty_permitted=False,
                                                                instance=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    model
        alias of MutableAddress

    fields = ('prefix', 'name', 'suffix', 'name_ext', 'phone', 'email', 'street', 'street2', 'street3', 'postal_code', 'city', 'r
ContactAddressForm.base_fields = OrderedDict([('prefix', <django.forms.fields.CharField object at 0x7f713c4f7
ContactAddressForm.declared_fields = OrderedDict()
ContactAddressForm.media

class shoop.admin.modules.shops.views.edit.ContactAddressFormPart (request,    ob-
                                                                    ject=None)

Bases: shoop.admin.form_part.FormPart

priority = 2

get_form_defs ()

form_valid (form)

class shoop.admin.modules.shops.views.edit.ShopEditView (**kwargs)
Bases: shoop.admin.form_part.SaveFormPartsMixin, shoop.admin.form_part.FormPartsViewMixin,
        shoop.admin.utils.views.CreateOrUpdateView

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model
    alias of Shop

template_name = 'shoop/admin/shops/edit.jinja'

context_object_name = 'shop'

base_form_part_classes = [<class 'shoop.admin.modules.shops.views.edit.ShopBaseFormPart'>, <class 'shoop.adm
form_part_class_provide_key = 'admin_shop_form_part'

get_object (queryset=None)

get_toolbar ()

form_valid (form)

shoop.admin.modules.shops.views.list module
class shoop.admin.modules.shops.views.list.ShopListView (**kwargs)
    Bases: shoop.admin.utils.views.PicotableListView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```

model

alias of Shop

columns = [<shoop.admin.utils.picotable.Column object at 0x7f713bd86dd8>, <shoop.admin.utils.picotable.Column obj

get_toolbar()

Module contents

class `shoop.admin.modules.shops.views.ShopEditView` (**kwargs)

Bases: `shoop.admin.form_part.SaveFormPartsMixin`, `shoop.admin.form_part.FormPartsViewMixin`,
`shoop.admin.utils.views.CreateOrUpdateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

base_form_part_classes = [<class 'shoop.admin.modules.shops.views.edit.ShopBaseFormPart'>, <class 'shoop.adm

context_object_name = 'shop'

form_part_class_provide_key = 'admin_shop_form_part'

form_valid(form)

get_object(queryset=None)

get_toolbar()

model

alias of Shop

template_name = 'shoop/admin/shops/edit.jinja'

class `shoop.admin.modules.shops.views.ShopListView` (**kwargs)

Bases: `shoop.admin.utils.views.PicotableListView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

columns = [<shoop.admin.utils.picotable.Column object at 0x7f713bd86dd8>, <shoop.admin.utils.picotable.Column obj

get_toolbar()

model

alias of Shop

Module contents

class `shoop.admin.modules.shops.ShopModule`

Bases: `shoop.admin.base.AdminModule`

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>

get_urls()

get_menu_entries(request)

get_required_permissions()

get_model_url(object, kind)

`shoop.admin.modules.suppliers` package

Subpackages

shoop.admin.modules.suppliers.views package**Submodules****shoop.admin.modules.suppliers.views.edit module**

```
class shoop.admin.modules.suppliers.views.edit.SupplierForm(data=None, files=None,
    auto_id='id_%s', pre-
    fix=None, initial=None,
    error_class=<class
    'django.forms.utils.ErrorList'>,
    label_suffix=None,
    empty_permitted=False,
    instance=None)
```

Bases: `django.forms.models.ModelForm`

class Meta

Bases: `object`

model

alias of `Supplier`

exclude = ('module_data',)

widgets = {'module_identifier': <class 'django.forms.widgets.Select'>}

`SupplierForm.base_fields` = `OrderedDict`([('name', <django.forms.fields.CharField object at 0x7f713c20d1d0>), (

`SupplierForm.declared_fields` = `OrderedDict`()

`SupplierForm.media`

```
class shoop.admin.modules.suppliers.views.edit.SupplierEditView(**kwargs)
```

Bases: `shoop.admin.utils.views.CreateOrUpdateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of `Supplier`

form_class

alias of `SupplierForm`

template_name = 'shoop/admin/suppliers/edit.jinja'

context_object_name = 'supplier'

get_object (*queryset=None*)

get_form (*form_class=None*)

shoop.admin.modules.suppliers.views.list module

```
class shoop.admin.modules.suppliers.views.list.SupplierListView(**kwargs)
```

Bases: `shoop.admin.utils.views.PicortableListView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of `Supplier`

columns = [<shoop.admin.utils.picotable.Column object at 0x7f713cb3f5f8>, <shoop.admin.utils.picotable.Column object at 0x7f713cb3f5f8>]

get_module_display (*instance*)

```
get_toolbar()
```

Module contents

```
class shoop.admin.modules.suppliers.views.SupplierEditView(**kwargs)
```

Bases: *shoop.admin.utils.views.CreateOrUpdateView*

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
context_object_name = 'supplier'
```

```
form_class
```

alias of *SupplierForm*

```
get_form(form_class=None)
```

```
get_object(queryset=None)
```

```
model
```

alias of *Supplier*

```
template_name = 'shoop/admin/suppliers/edit.jinja'
```

```
class shoop.admin.modules.suppliers.views.SupplierListView(**kwargs)
```

Bases: *shoop.admin.utils.views.PicotableView*

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
columns = [<shoop.admin.utils.picotable.Column object at 0x7f713cb3f5f8>, <shoop.admin.utils.picotable.Column object at 0x7f713cb3f5f8>]
```

```
get_module_display(instance)
```

```
get_toolbar()
```

```
model
```

alias of *Supplier*

Module contents

```
class shoop.admin.modules.suppliers.SupplierModule
```

Bases: *shoop.admin.base.AdminModule*

```
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
```

```
breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>
```

```
get_urls()
```

```
get_menu_entries(request)
```

```
get_required_permissions()
```

```
get_model_url(object, kind)
```

shoop.admin.modules.system package

Subpackages

shoop.admin.modules.system.views package

Submodules

shoop.admin.modules.system.views.telemetry module

```
class shoop.admin.modules.system.views.telemetry.TelemetryView (**kwargs)
    Bases: django.views.generic.base.TemplateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    template_name = 'shoop/admin/system/telemetry.jinja'

    get_context_data (**kwargs)

    get (request, *args, **kwargs)

    post (request, *args, **kwargs)
```

Module contents**Module contents**

```
class shoop.admin.modules.system.SystemModule
    Bases: shoop.admin.base.AdminModule

    name = <django.utils.functional.lazy.<locals>.__proxy__ object>

    category = <django.utils.functional.lazy.<locals>.__proxy__ object>

    get_urls ()

    get_menu_category_icons ()

    get_menu_entries (request)

    get_required_permissions ()

    get_notifications (request)
```

shoop.admin.modules.taxes package**Subpackages****shoop.admin.modules.taxes.views package****Submodules****shoop.admin.modules.taxes.views.edit module**

```
class shoop.admin.modules.taxes.views.edit.TaxForm (**kwargs)
    Bases: shoop.utils.multilanguage_model_form.MultiLanguageModelForm

    class Meta
        Bases: object

        model
            alias of Tax

        fields = ['name', 'code', 'rate', 'amount_value', 'currency', 'enabled']

    TaxForm.base_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713c3dd748>), ('code',
    TaxForm.declared_fields = OrderedDict([('rate', <shoop.admin.forms.fields.PercentageField object at 0x7f713b8d
    TaxForm.media
```

```
class shoop.admin.modules.taxes.views.edit.CustomerTaxGroupForm(**kwargs)
    Bases: shoop.utils.multilanguage_model_form.MultiLanguageModelForm

    class Meta
        Bases: object

        model
            alias of CustomerTaxGroup

        fields = ['name']

    CustomerTaxGroupForm.base_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713c3dd6a0>), ('enabled', <django.forms.fields.BooleanField object at 0x7f713c3dd6a0>)])
    CustomerTaxGroupForm.declared_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713c3dd6a0>), ('enabled', <django.forms.fields.BooleanField object at 0x7f713c3dd6a0>)])
    CustomerTaxGroupForm.media

class shoop.admin.modules.taxes.views.edit.TaxClassForm(**kwargs)
    Bases: shoop.utils.multilanguage_model_form.MultiLanguageModelForm

    class Meta
        Bases: object

        model
            alias of TaxClass

        fields = ['name', 'enabled']

    TaxClassForm.base_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713c3dd6a0>), ('enabled', <django.forms.fields.BooleanField object at 0x7f713c3dd6a0>)])
    TaxClassForm.declared_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713c3dd6a0>), ('enabled', <django.forms.fields.BooleanField object at 0x7f713c3dd6a0>)])
    TaxClassForm.media

class shoop.admin.modules.taxes.views.edit.TaxEditView(**kwargs)
    Bases: shoop.admin.modules.taxes.views.edit._Breadcrumbed,
    shoop.admin.utils.views.CreateOrUpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of Tax

    form_class
        alias of TaxForm

    template_name = 'shoop/admin/taxes/edit_tax.jinja'
    context_object_name = 'tax'
    add_form_errors_as_messages = True
    parent_name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    parent_url = 'shoop_admin:tax.list'

class shoop.admin.modules.taxes.views.edit.CustomerTaxGroupEditView(**kwargs)
    Bases: shoop.admin.modules.taxes.views.edit._Breadcrumbed,
    shoop.admin.utils.views.CreateOrUpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of CustomerTaxGroup

    form_class
        alias of CustomerTaxGroupForm
```



```

    template_name = 'shoop/admin/taxes/edit_customer_tax_group.jinja'
    context_object_name = 'customer_tax_group'
    parent_name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    parent_url = 'shoop_admin:customer_tax_group.list'
class shoop.admin.modules.taxes.views.edit.TaxClassEditView (**kwargs)
    Bases: shoop.admin.modules.taxes.views.edit._Breadcrumbed,
            shoop.admin.utils.views.CreateOrUpdateView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    model
        alias of TaxClass
    template_name = 'shoop/admin/taxes/edit_tax_class.jinja'
    form_class
        alias of TaxClassForm
    context_object_name = 'tax_class'
    parent_name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    parent_url = 'shoop_admin:tax_class.list'

```

shoop.admin.modules.taxes.views.list module

```

class shoop.admin.modules.taxes.views.list.TaxListView (**kwargs)
    Bases: shoop.admin.utils.views.PicortableListView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    model
        alias of Tax
    columns = [<shoop.admin.utils.picortable.Column object at 0x7f713be6bd30>, <shoop.admin.utils.picortable.Column object at 0x7f713be6bd30>]
class shoop.admin.modules.taxes.views.list.CustomerTaxGroupListView (**kwargs)
    Bases: shoop.admin.utils.views.PicortableListView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    model
        alias of CustomerTaxGroup
    columns = [<shoop.admin.utils.picortable.Column object at 0x7f713be6ba90>]
class shoop.admin.modules.taxes.views.list.TaxClassListView (**kwargs)
    Bases: shoop.admin.utils.views.PicortableListView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    model
        alias of TaxClass
    columns = [<shoop.admin.utils.picortable.Column object at 0x7f713be6b518>]

```

Module contents

```

class shoop.admin.modules.taxes.views.TaxClassListView (**kwargs)
    Bases: shoop.admin.utils.views.PicortableListView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```

```
columns = [<shoop.admin.utils.picotable.Column object at 0x7f713be6b518>]

model
    alias of TaxClass
class shoop.admin.modules.taxes.views.TaxClassEditView(**kwargs)
    Bases: shoop.admin.modules.taxes.views.edit._Breadcrumbed,
           shoop.admin.utils.views.CreateOrUpdateView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    context_object_name = 'tax_class'

    form_class
        alias of TaxClassForm

    model
        alias of TaxClass

    parent_name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    parent_url = 'shoop_admin:tax_class.list'
    template_name = 'shoop/admin/taxes/edit_tax_class.jinja'

class shoop.admin.modules.taxes.views.TaxEditView(**kwargs)
    Bases: shoop.admin.modules.taxes.views.edit._Breadcrumbed,
           shoop.admin.utils.views.CreateOrUpdateView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    add_form_errors_as_messages = True

    context_object_name = 'tax'

    form_class
        alias of TaxForm

    model
        alias of Tax

    parent_name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    parent_url = 'shoop_admin:tax.list'
    template_name = 'shoop/admin/taxes/edit_tax.jinja'

class shoop.admin.modules.taxes.views.TaxListView(**kwargs)
    Bases: shoop.admin.utils.views.PicotableListView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    columns = [<shoop.admin.utils.picotable.Column object at 0x7f713be6bd30>, <shoop.admin.utils.picotable.Column object at 0x7f713be6bd30>]

    model
        alias of Tax

class shoop.admin.modules.taxes.views.CustomerTaxGroupEditView(**kwargs)
    Bases: shoop.admin.modules.taxes.views.edit._Breadcrumbed,
           shoop.admin.utils.views.CreateOrUpdateView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    context_object_name = 'customer_tax_group'

    form_class
        alias of CustomerTaxGroupForm
```

```

    model
        alias of CustomerTaxGroup

    parent_name = <django.utils.functional.lazy.<locals>.__proxy__ object>

    parent_url = 'shoop_admin:customer_tax_group.list'

    template_name = 'shoop/admin/taxes/edit_customer_tax_group.jinja'

class shoop.admin.modules.taxes.views.CustomerTaxGroupListView (**kwargs)
    Bases: shoop.admin.utils.views.PicotableView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    columns = [<shoop.admin.utils.picotable.Column object at 0x7f713be6ba90>]

    model
        alias of CustomerTaxGroup

```

Module contents

```

class shoop.admin.modules.taxes.TaxModule
    Bases: shoop.admin.base.AdminModule

    name = <django.utils.functional.lazy.<locals>.__proxy__ object>

    category = <django.utils.functional.lazy.<locals>.__proxy__ object>

    get_urls ()

    get_menu_category_icons ()

    get_menu_entries (request)

    get_required_permissions ()

    get_model_url (object, kind)

```

shoop.admin.modules.users package

Subpackages

shoop.admin.modules.users.views package

Submodules

shoop.admin.modules.users.views.detail module

```

class shoop.admin.modules.users.views.detail.BaseUserForm (*args, **kwargs)
    Bases: django.forms.models.ModelForm

    save (commit=True)

    base_fields = OrderedDict([('password', <django.forms.fields.CharField object at 0x7f713cd385c0>), ('permission_in

    declared_fields = OrderedDict([('password', <django.forms.fields.CharField object at 0x7f713cd385c0>), ('permissi

    media

class shoop.admin.modules.users.views.detail.UserDetailToolbar (view)
    Bases: shoop.admin.toolbar.Toolbar

```

```
class shoop.admin.modules.users.views.detail.UserDetailView (**kwargs)
    Bases: shoop.admin.utils.views.CreateOrUpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    template_name = 'shoop/admin/users/detail.jinja'
    context_object_name = 'user'
    fields = ('username', 'email', 'first_name', 'last_name')
    get_form_class()
    get_initial()
    get_toolbar()
    save_form(form)
    get_success_url()
    post(request, *args, **kwargs)
    dispatch(request, *args, **kwargs)
```

shoop.admin.modules.users.views.list module

```
class shoop.admin.modules.users.views.list.UserListView (**kwargs)
    Bases: shoop.admin.utils.views.PicortableListView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model = 'auth.User'
    columns = [<shoop.admin.utils.picortable.Column object at 0x7f713cf2ea20>, <shoop.admin.utils.picortable.Column object at 0x7f713cf2ea20>]
    get_model()
    get_queryset()
    get_context_data(**kwargs)
    get_object_abstract(instance, item)
```

shoop.admin.modules.users.views.password module

```
class shoop.admin.modules.users.views.password.PasswordChangeForm(changing_user,
                                                                    target_user,
                                                                    *args,
                                                                    **kwargs)
    Bases: django.forms.forms.Form

    error_messages = {'password_incorrect': <django.utils.functional.lazy.<locals>.__proxy__ object at 0x7f713cf2ef60>, 'password_too_short': <django.utils.functional.lazy.<locals>.__proxy__ object at 0x7f713cf2ef60>, 'passwords_dont_match': <django.utils.functional.lazy.<locals>.__proxy__ object at 0x7f713cf2ef60>}
    clean_password2()
    clean_old_password()
        Validates that the old_password field is correct.
    save(commit=True)
    base_fields = OrderedDict([('old_password', <django.forms.fields.CharField object at 0x7f713cf2efd0>), ('password1', <django.forms.fields.CharField object at 0x7f713cf2efd0>), ('password2', <django.forms.fields.CharField object at 0x7f713cf2efd0>)])
    declared_fields = OrderedDict([('old_password', <django.forms.fields.CharField object at 0x7f713cf2efd0>), ('password1', <django.forms.fields.CharField object at 0x7f713cf2efd0>), ('password2', <django.forms.fields.CharField object at 0x7f713cf2efd0>)])
    media
```

```
class shoop.admin.modules.users.views.password.UserChangePasswordView (**kwargs)
    Bases: django.views.generic.edit.UpdateView
```

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

form_class

alias of `PasswordChangeForm`

template_name = 'shoop/admin/users/change_password.jinja'

model = 'auth.User'

title = <django.utils.functional.lazy.<locals>.__proxy__ object>

get_queryset ()

get_toolbar ()

get_form_kwargs ()

get_context_data (**kwargs)

form_valid (form)

get_success_url ()

```
class shoop.admin.modules.users.views.password.UserResetPasswordView (**kwargs)
    Bases: django.views.generic.detail.DetailView
```

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model = 'auth.User'

template_name = 'shoop/admin/users/reset_password.jinja'

title = <django.utils.functional.lazy.<locals>.__proxy__ object>

get_queryset ()

process_user (user)

post (request, *args, **kwargs)

get_success_url ()

shoop.admin.modules.users.views.permissions module

```
class shoop.admin.modules.users.views.permissions.PermissionChangeFormBase (changing_user,
                                                                            *args,
                                                                            **kwargs)
```

Bases: `django.forms.models.ModelForm`

clean_old_password ()

Validates that the old_password field is correct.

clean_members ()

clean_permission_groups ()

clean ()

save ()

base_fields = `OrderedDict`([(‘old_password’, <django.forms.fields.CharField object at 0x7f713cd91be0>)])

declared_fields = `OrderedDict`([(‘old_password’, <django.forms.fields.CharField object at 0x7f713cd91be0>)])

media

```
class shoop.admin.modules.users.views.permissions.UserChangePermissionsView (**kwargs)
    Bases: django.views.generic.edit.UpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    template_name = 'shoop/admin/users/change_permissions.jinja'
    model = 'auth.User'
    title = <django.utils.functional.lazy.<locals>.__proxy__ object>
    get_form_class()
    get_queryset()
    get_toolbar()
    get_form_kwargs()
    get_context_data (**kwargs)
    form_valid(form)
    get_success_url()
```

Module contents

```
class shoop.admin.modules.users.views.UserListView (**kwargs)
    Bases: shoop.admin.utils.views.PicotableView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    columns = [<shoop.admin.utils.picotable.Column object at 0x7f713cf2ea20>, <shoop.admin.utils.picotable.Column object at 0x7f713cf2ea20>]
    get_context_data (**kwargs)
    get_model()
    get_object_abstract(instance, item)
    get_queryset()
    model = 'auth.User'

class shoop.admin.modules.users.views.UserDetailView (**kwargs)
    Bases: shoop.admin.utils.views.CreateOrUpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    context_object_name = 'user'
    dispatch(request, *args, **kwargs)
    fields = ('username', 'email', 'first_name', 'last_name')
    get_form_class()
    get_initial()
    get_success_url()
    get_toolbar()
    post(request, *args, **kwargs)
    save_form(form)
    template_name = 'shoop/admin/users/detail.jinja'
```

```

class shoop.admin.modules.users.views.UserChangePasswordView (**kwargs)
    Bases: django.views.generic.edit.UpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    form_class
        alias of PasswordChangeForm

    form_valid (form)

    get_context_data (**kwargs)

    get_form_kwargs ()

    get_queryset ()

    get_success_url ()

    get_toolbar ()

    model = 'auth.User'

    template_name = 'shoop/admin/users/change_password.jinja'

    title = <django.utils.functional.lazy.<locals>.__proxy__ object>

class shoop.admin.modules.users.views.UserResetPasswordView (**kwargs)
    Bases: django.views.generic.detail.DetailView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    get_queryset ()

    get_success_url ()

    model = 'auth.User'

    post (request, *args, **kwargs)

    process_user (user)

    template_name = 'shoop/admin/users/reset_password.jinja'

    title = <django.utils.functional.lazy.<locals>.__proxy__ object>

class shoop.admin.modules.users.views.UserChangePermissionsView (**kwargs)
    Bases: django.views.generic.edit.UpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    form_valid (form)

    get_context_data (**kwargs)

    get_form_class ()

    get_form_kwargs ()

    get_queryset ()

    get_success_url ()

    get_toolbar ()

    model = 'auth.User'

    template_name = 'shoop/admin/users/change_permissions.jinja'

    title = <django.utils.functional.lazy.<locals>.__proxy__ object>

```

Module contents

```
class shoop.admin.modules.users.UserModule
    Bases: shoop.admin.base.AdminModule

    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    category = <django.utils.functional.lazy.<locals>.__proxy__ object>
    breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>
    get_urls ()
    get_menu_category_icons ()
    get_menu_entries (request)
    get_search_results (request, query)
    get_required_permissions ()
    get_model_url (object, kind)
```

Module contents

shoop.admin.template_helpers package

Submodules

shoop.admin.template_helpers.shoop_admin module This module is installed as the `shoop_admin` template function namespace.

```
shoop.admin.template_helpers.shoop_admin.get_menu_entry_categories (context)
shoop.admin.template_helpers.shoop_admin.get_front_url (context)
shoop.admin.template_helpers.shoop_admin.get_config (context)
shoop.admin.template_helpers.shoop_admin.model_url (model, kind='detail', de-
                                                    fault=None)
    Get a model URL of the given kind for a model (instance or class).
```

Parameters

- **model** (*django.db.Model*) – The model instance or class.
- **kind** (*str*) – The URL kind to retrieve. See `get_model_url`.
- **default** (*str/None*) – Default value to return if model URL retrieval fails. If `None`, the `NoModelUrl` exception is (re)raised.

Returns URL string.

Return type *str*

Module contents

shoop.admin.templatetags package

Submodules

shoop.admin.templatetags.shoop_admin module**class** shoop.admin.templatetags.shoop_admin.**Bootstrap3Namespace**Bases: `object`**field** (*field*, ***kwargs*)**form** (*form*, ***kwargs*)**Module contents****shoop.admin.utils package****Submodules****shoop.admin.utils.bs3_renderers module****class** shoop.admin.utils.bs3_renderers.**AdminFieldRenderer** (*field*, ***kwargs*)Bases: `bootstrap3.renderers.FieldRenderer`**get_label** ()**add_class_attrs** (**args*, ***kwargs*)**add_help_attrs** (**args*, ***kwargs*)**shoop.admin.utils.forms module****shoop.admin.utils.forms.filter_form_field_choices** (*field*, *predicate*, *invert=False*)

Filter choices of a form field and its widget by predicate.

The predicate may be a callable of the signature `(pair) -> bool` or an iterable of allowable values.**Parameters**

- **field** (*django.forms.Field*) – Form field.
- **predicate** (*function/Iterable*) – Predicate
- **invert** (*bool*) – Invert the semantics of the predicate, i.e. items matching it will be rejected.

Returns Nothing. The field is modified in-place.**shoop.admin.utils.forms.add_form_errors_as_messages** (*request*, *form*)

Add the form's errors, if any, into the request as messages.

Parameters

- **request** (*django.http.HttpRequest*) – Request to messagify
- **form** (*django.forms.Form*) – The errorful form.

Returns Number of messages added. May be thousands, for a very unlucky form.**Return type** `int`**shoop.admin.utils.forms.flatatt_filter** (*attrs*)

shoop.admin.utils.permissions module

`shoop.admin.utils.permissions.get_default_model_permissions(model)`

Return a set of all default permissions for a given model.

Parameters `model` (`django.db.Model`) – Model class

Returns Set of default model permissions as strings

Return type `set[str]`

`shoop.admin.utils.permissions.get_missing_permissions(user, permissions)`

Return a set of missing permissions for a given iterable of permission strings.

Parameters

- **user** (`django.contrib.auth.models.User`) – User instance to check for permissions
- **permissions** (`Iterable[str]`) – Iterable of permission strings

Returns Set of missing permission strings

Return type `set[str]`

`shoop.admin.utils.permissions.get_permissions_from_urls(urls)`

Return a set of permissions for a given iterable of urls.

Parameters `urls` (`Iterable[django.core.urlresolvers.RegexURLPattern]`) – Iterable of url objects to check for permissions

Returns Set of permissions for urls as strings

Return type `set[str]`

`shoop.admin.utils.permissions.get_permission_object_from_string(permission_string)`

Given a permission string of the form `app_label.permission_string`, get actual permission object.

Parameters `permission_string` – Permission string

Returns Permission object

Return type `django.contrib.auth.models.Permission`

shoop.admin.utils.picotable module

`shoop.admin.utils.picotable.maybe_callable(thing, context=None)`

If `thing` is callable, return it. If `thing` names a callable attribute of `context`, return it.

`shoop.admin.utils.picotable.maybe_call(thing, context, args=None, kwargs=None)`

If `thing` is callable, call it with `args` and `kwargs` and return the value. If `thing` names a callable attribute of `context`, call it with `args` and `kwargs` and return the value. Otherwise return `thing`.

class `shoop.admin.utils.picotable.Filter`

Bases: `object`

type = `None`

to_json (`context`)

filter_queryset (`queryset, column, value`)

class `shoop.admin.utils.picotable.ChoicesFilter` (`choices=None, filter_field=None, default=None`)

Bases: `shoop.admin.utils.picotable.Filter`

type = `'choices'`

to_json (`context`)

filter_queryset (*queryset, column, value*)

class shoop.admin.utils.picotable.**RangeFilter** (*min=None, max=None, step=None, field_type=None, filter_field=None*)

Bases: *shoop.admin.utils.picotable.Filter*

Parameters

- **filter_field** (*str/None*) – Filter field (Django query expression). If None, column ID is used.
- **min** – Minimum value.
- **max** – Maximum value.
- **step** – Step value. See the HTML5 documentation for semantics.
- **field_type** (*str/None*) – Field type string. See the HTML5 documentation for semantics.

type = 'range'

to_json (*context*)

filter_queryset (*queryset, column, value*)

class shoop.admin.utils.picotable.**DateRangeFilter** (**args, **kwargs*)

Bases: *shoop.admin.utils.picotable.RangeFilter*

filter_queryset (*queryset, column, value*)

class shoop.admin.utils.picotable.**TextFilter** (*field_type=None, placeholder=None, operator='icontains', filter_field=None*)

Bases: *shoop.admin.utils.picotable.Filter*

Parameters

- **filter_field** (*str/None*) – Filter field (Django query expression). If None, column ID is used.
- **field_type** (*str/None*) – Field type string. See the HTML5 documentation for semantics.
- **placeholder** (*str/None*) – Field placeholder string.
- **operator** (*str*) – Django operator for the queryset.

type = 'text'

to_json (*context*)

filter_queryset (*queryset, column, value*)

class shoop.admin.utils.picotable.**MultiFieldTextFilter** (*filter_fields, **kwargs*)

Bases: *shoop.admin.utils.picotable.TextFilter*

Parameters

- **filter_field** (*list<str>*) – List of Filter fields (Django query expression).
- **kwargs** – Kwargs for *TextFilter*.

filter_queryset (*queryset, column, value*)

class shoop.admin.utils.picotable.**Column** (*id, title, **kwargs*)

Bases: *object*

to_json (*context=None*)

```
sort_queryset (queryset, desc=False)
```

```
filter_queryset (queryset, value)
```

```
get_display_value (context, object)
```

```
class shoop.admin.utils.picotable.Picotable (request, columns, queryset, context)
```

```
    Bases: object
```

```
    process_queryset (query)
```

```
    get_data (query)
```

```
    process_item (object)
```

```
    get_verbose_name_plural ()
```

```
class shoop.admin.utils.picotable.PicotableViewMixin
```

```
    Bases: object
```

```
    columns = []
```

```
    picotable_class
```

```
        alias of Picotable
```

```
    template_name = 'shoop/admin/base_picotable.jinja'
```

```
    process_picotable (query_json)
```

```
    get (request, *args, **kwargs)
```

```
    get_object_url (instance)
```

```
    get_object_abstract (instance, item)
```

Get the object abstract lines (used for mobile layouts) for this object.

Supported keys in abstract line dicts are:

- text (required)
- title
- class (CSS class name – header for instance)
- raw (boolean; whether or not the text is raw HTML)

Parameters

- **instance** – The instance
- **item** – The item dict so far. Useful for reusing precalculated values.

Returns Iterable of dicts to pass through to the picotable javascript

Return type Iterable[dict]

```
    get_filter ()
```

shoop.admin.utils.search module

```
class shoop.admin.utils.search.FuzzyMatcher (query)
```

```
    Bases: object
```

```
    test (text)
```

```
shoop.admin.utils.search.split_query (query_string, minimum_part_length=3)
```

Split a string into a set of non-empty words, none shorter than `minimum_part_length` characters.

Parameters

- **query_string** (*str*) – Query string
- **minimum_part_length** (*int*) – Minimum part length

Returns Set of query parts

Return type set[str]

shoop.admin.utils.urls module

```
class shoop.admin.utils.urls.AdminRegexURLPattern(regex, callback, de-
                                                fault_args=None, name=None,
                                                require_authentication=True, per-
                                                missions=())
```

Bases: django.core.urlresolvers.RegexURLPattern

wrap_with_permissions (*view_func*)

callback

```
shoop.admin.utils.urls.admin_url(regex, view, kwargs=None, name=None, prefix='', re-
                                require_authentication=True, permissions=())
```

```
shoop.admin.utils.urls.get_edit_and_list_urls(url_prefix, view_template,
                                              name_template, permissions=())
```

Get a list of edit/new/list URLs for (presumably) an object type with standardized URLs and names.

Parameters

- **url_prefix** (*str*) – What to prefix the generated URLs with. E.g. "`^taxes/tax`"
- **view_template** (*str*) – A template string for the dotted name of the view class. E.g. "`shoop.admin.modules.taxes.views.Tax%sView`"
- **name_template** (*str*) – A template string for the URLnames. E.g. "`tax.%s`"

Returns List of URLs

Return type list[AdminRegexURLPattern]

```
exception shoop.admin.utils.urls.NoModelUrl
```

Bases: `ValueError`

```
shoop.admin.utils.urls.get_model_url(object, kind='detail')
```

Get a an admin object URL for the given object or object class by interrogating each admin module.

Raises `NoModelUrl` if lookup fails

Parameters

- **object** (*class*) – Model or object class.
- **kind** (*str*) – URL kind. Currently “new”, “list”, “edit”, “detail”.

Returns Resolved URL.

Return type str

```
shoop.admin.utils.urls.derive_model_url(model_class, urlname_prefix, object, kind)
```

Try to guess a model URL for the given `object` and `kind`.

An utility for people implementing `get_model_url`.

Parameters

- **model_class** (*class*) – The model class the object must be an instance or subclass of.

- **urlname_prefix**(*str*) – URLname prefix. For instance, `shoop_admin:product`.
- **object**(*django.db.models.Model* | *class*) – The model or model class as passed to `get_model_url`
- **kind**(*str*) – URL kind as passed to `get_model_url`.

Returns Resolved URL or None.

Return type `str`|None

```
shoop.admin.utils.urls.manipulate_query_string(url, **qs)
```

```
shoop.admin.utils.urls.get_model_front_url(request, object)
```

Get a frontend URL for an object.

Parameters

- **request** (*HttpRequest*) – Request
- **object** (*django.db.models.Model*) – A model instance

Returns URL or None

Return type `str`|None

shoop.admin.utils.views module

```
class shoop.admin.utils.views.CreateOrUpdateView(**kwargs)
```

Bases: `django.views.generic.edit.UpdateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

add_form_errors_as_messages = False

get_object (*queryset=None*)

get_toolbar ()

get_context_data (**kwargs)

get_save_form_id ()

get_return_url ()

get_new_url ()

get_success_url ()

get_form_kwargs ()

form_valid (*form*)

save_form (*form*)

form_invalid (*form*)

```
shoop.admin.utils.views.add_create_or_change_message(request, instance, is_new)
```

```
shoop.admin.utils.views.get_create_or_change_title(request, instance,
                                                    name_field=None)
```

Get a title suitable for an create-or-update view.

Parameters

- **request** (*HttpRequest*) – Request
- **instance** (*django.db.models.Model*) – Model instance
- **name_field** (*str*) – Which property to try to read the name from. If None, use `str`

Returns Title

Return type `str`

`shoop.admin.utils.views.check_and_raise_if_only_one_allowed(setting_name, obj)`

class `shoop.admin.utils.views.PicotableListView(**kwargs)`

Bases: `shoop.admin.utils.picotable.PicotableViewMixin`,
`django.views.generic.list.ListView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

`get_toolbar()`

`get_context_data(**kwargs)`

`get_object_abstract(instance, item)`

Module contents

shoop.admin.views package

Submodules

shoop.admin.views.dashboard module

class `shoop.admin.views.dashboard.DashboardView(**kwargs)`

Bases: `django.views.generic.base.TemplateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

`template_name = 'shoop/admin/dashboard/dashboard.jinja'`

`get_context_data(**kwargs)`

`get(request, *args, **kwargs)`

shoop.admin.views.menu module

class `shoop.admin.views.menu.MenuView(**kwargs)`

Bases: `django.views.generic.base.TemplateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

`template_name = 'shoop/admin/base/_main_menu.jinja'`

shoop.admin.views.search module

`shoop.admin.views.search.get_search_results(request, query)`

class `shoop.admin.views.search.SearchView(**kwargs)`

Bases: `django.views.generic.base.View`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

`get(request, *args, **kwargs)`

shoop.admin.views.select module

class `shoop.admin.views.select.MultiselectAjaxView` (***kwargs*)

Bases: `django.views.generic.base.TemplateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model = `None`

search_fields = `[]`

result_limit = `20`

init_search_fields (*cls*)

get_data (*request, *args, **kwargs*)

get (*request, *args, **kwargs*)

Module contents

Submodules

shoop.admin.base module

class `shoop.admin.base.AdminModule`

Bases: `object`

name = `'_Base_'`

breadcrumbs_menu_entry = `None`

get_urls ()

Return type `list[django.core.urlresolvers.RegexURLPattern]`

get_menu_category_icons ()

Return type `dict[str, str]`

get_menu_entries (*request*)

Return type `list[shoop.admin.base.MenuEntry]`

get_search_results (*request, query*)

Return type `list[shoop.admin.base.SearchResult]`

get_dashboard_blocks (*request*)

Return type `list[shoop.admin.dashboard.DashboardBlock]`

get_required_permissions ()

Return type `list[str]`

get_notifications (*request*)

Return type `list[shoop.admin.base.Notification]`

get_activity (*request, cutoff*)

Parameters

- **cutoff** (*datetime.datetime*) – Cutoff datetime
- **request** (*django.http.request.HttpRequest*) – Request

Returns list[shoop.admin.base.Activity]

get_model_url (*object*, *kind*)

Retrieve an admin URL for the given object of the kind *kind*.

A falsy value must be returned if the module does not know how to reverse the given object.

Parameters

- **object** (*django.db.models.Model*) – A object instance (or object class).
- **kind** (*str*) – URL kind. Currently “detail”, “list” or “new”.

Returns The reversed URL or none.

Return type str|None

class shoop.admin.base.**Resolvable**

Bases: *object*

url

Resolve this object’s `_url` to an actual URL.

Returns URL or no URL.

Return type str|None

original_url

class shoop.admin.base.**MenuEntry** (*text*, *url*, *icon=None*, *category=None*, *aliases=()*)

Bases: *shoop.admin.base.Resolvable*

get_search_query_texts ()

class shoop.admin.base.**SearchResult** (*text*, *url*, *icon=None*, *category=None*, *is_action=False*, *relevance=100*, *target=None*)

Bases: *shoop.admin.base.Resolvable*

to_json ()

class shoop.admin.base.**Notification** (*text*, *title=None*, *url=None*, *kind='info'*, *dismissal_url=None*, *datetime=None*)

Bases: *shoop.admin.base.Resolvable*

Parameters

- **text** (*str*) – The notification’s text.
- **title** (*str/None*) – An optional title for the notification.
- **url** (*str/None*) – The optional main URL for the notification.
- **kind** (*str*) – The kind of the notification (see KINDS)
- **dismissal_url** (*str/None*) – An optional dismissal URL for the notification. The admin framework will add a button that will cause an AJAX post into this URL.
- **datetime** (*datetime*) – An optional date+time for this notification.

KINDS = ('info', 'success', 'warning', 'danger')

class shoop.admin.base.**Activity** (*datetime*, *text*, *url=None*)

Bases: *shoop.admin.base.Resolvable*

class shoop.admin.base.**OrderSection**

Bases: *object*

Subclass this and add the class to the `admin_order_section` provide list to show a custom section on the order detail on admin.

identifier must be unique *name* the section caption *icon* the section icon *template* the section template file *extra_js* the section extra javascript template file,

Unexpected indentation.

set a file which contains js code inside a <script> tag

Block quote ends without a blank line; unexpected unindent.

order the order

```
identifier = ''
```

```
name = ''
```

```
icon = ''
```

```
template = ''
```

```
extra_js = ''
```

```
order = 0
```

```
static visible_for_order (order)
```

Returns whether this sections must be visible while editing the *order* :type order:
shoop.core.models.Order :return whether this section must be shown in order section list :rtype:
bool

```
static get_context_data (order)
```

Returns additional information to be used in the template

To fetch this data in the template, just access it through `OrderSection_identifier.data`

Field list ends without a blank line; unexpected unindent.

:return additional context data :rtype: object|None

shoop.admin.breadcrumbs module

```
class shoop.admin.breadcrumbs.Breadcrumbs (entries)
```

Bases: `object`

```
classmethod infer (context)
```

Infer breadcrumbs from the rendering context.

Parameters `context` (`jinja2.runtime.Context`) – Jinja Context

Returns Breadcrumbs object or None if things fail

Return type Breadcrumbs|None

```
get_entries (request)
```

shoop.admin.currencybound module

```
class shoop.admin.currencybound.CurrencyBound (currency=None, *args, **kwargs)
```

Bases: `object`

Mixin for adding currency property defaulting currency of the first Shop.

The currency property is “lazy” so that database is not accessed on initialization, since this mixin will be used by some `AdminModule` classes and they will be initialized at import time by `module_registry.register` (which is called at import because `admin.urls` calls `get_module_urls` at import).

currency**shoop.admin.form_part module**

class shoop.admin.form_part.**TemplatedFormDef** (*name*, *form_class*, *template_name*, *required=True*, *kwargs=None*)

Bases: *shoop.utils.form_group.FormDef*

class shoop.admin.form_part.**FormPart** (*request*, *object=None*)

Bases: *object*

priority = 0

get_form_defs ()

form_valid (*form*)

class shoop.admin.form_part.**FormPartsViewMixin**

Bases: *object*

fields = ()

request = None

form_part_class_provide_key = None

base_form_part_classes = ()

get_form_class ()

get_form_part_classes ()

get_form_parts (*object*)

get_form (*form_class=None*)

class shoop.admin.form_part.**SaveFormPartsMixin**

Bases: *object*

request = None

object = None

save_form_parts (*form*)

shoop.admin.menu module

shoop.admin.menu.**get_menu_entry_categories** (*request*)

shoop.admin.module_registry module

shoop.admin.module_registry.**register** (*module_class*)

shoop.admin.module_registry.**discover** ()

shoop.admin.module_registry.**get_modules** ()

Return type list[shoop.admin.base.AdminModule]

shoop.admin.module_registry.**get_module_urls** ()

shoop.admin.module_registry.**replace_modules** (*new_module_classes*)

Context manager to temporarily replace all modules with something else.

Test utility, mostly.

```
>>> def some_test():
...     with replace_modules(["foo.bar:QuuxModule"]):
...         pass # do stuff
```

Parameters `new_module_classes` – Iterable of module classes, like you’d pass to `register`

shoop.admin.toolbar module

class `shoop.admin.toolbar.BaseActionButton` (*text*='', *icon*=None, *disable_reason*=None, *tooltip*=None, *extra_css_class*='btn-default')

Bases: `object`

Parameters

- **text** – The actual text for the button.
- **icon** – Icon CSS class string
- **disable_reason** (*str*/None) – The reason for this button to be disabled. It’s considered good UX to have an user-visible reason for disabled actions; thus the only way to disable an action is to set the reason. See <http://stackoverflow.com/a/372503/51685>.
- **tooltip** (*str*/None) – Tooltip string, if any. May be replaced by the disable reason.
- **extra_css_class** (*str*) – Extra CSS class(es)

base_css_classes = ('btn',)

render (*request*)

Yield HTML bits for this object. :type request: `HttpRequest` :rtype: `Iterable[str]`

render_label ()

get_computed_class ()

class `shoop.admin.toolbar.URLActionButton` (*url*, ***kwargs*)

Bases: `shoop.admin.toolbar.BaseActionButton`

An action button that renders as a link leading to *url*.

Parameters **url** (*str*) – The URL to navigate to. For convenience, if this contains no slashes, `reverse` is automatically called on it.

render (*request*)

class `shoop.admin.toolbar.NewActionButton` (*url*, ***kwargs*)

Bases: `shoop.admin.toolbar.URLActionButton`

An URL button with sane “new” visual semantics

classmethod **for_model** (*model*, ***kwargs*)

Generate a `NewActionButton` for a model, auto-wiring the URL.

Parameters **model** – Model class

Return type `shoop.admin.toolbar.NewActionButton`|None

class `shoop.admin.toolbar.JavaScriptActionButton` (*onclick*, ***kwargs*)

Bases: `shoop.admin.toolbar.BaseActionButton`

An action button that uses `onclick` for action dispatch.

render (*request*)

```
class shoop.admin.toolbar.PostActionButton(post_url=None, name=None, value=None,
                                          form_id=None, confirm=None, **kwargs)
    Bases: shoop.admin.toolbar.BaseActionButton
```

An action button that renders as a button POSTing a form containing name='value' to post_url.

render (*request*)

```
class shoop.admin.toolbar.DropdownActionButton(items, split_button=None, **kwargs)
    Bases: shoop.admin.toolbar.BaseActionButton
```

An action button with a chevron button to open a dropdown menu.

base_css_classes = ('btn', 'dropdown-toggle')

render_dropdown (*request*)

render (*request*)

```
class shoop.admin.toolbar.DropdownItem(url='#', onclick=None, **kwargs)
    Bases: shoop.admin.toolbar.BaseActionButton
```

An item to be shown in a *DropdownActionButton*.

base_css_classes = ()

render (*request*)

```
class shoop.admin.toolbar.DropdownDivider(text='', icon=None, disable_reason=None,
                                          tooltip=None, extra_css_class='btn-default')
    Bases: shoop.admin.toolbar.BaseActionButton
```

A Divider for DropdownActionButtons.

Parameters

- **text** – The actual text for the button.
- **icon** – Icon CSS class string
- **disable_reason** (*str/None*) – The reason for this button to be disabled. It's considered good UX to have an user-visible reason for disabled actions; thus the only way to disable an action is to set the reason. See <http://stackoverflow.com/a/372503/51685>.
- **tooltip** (*str/None*) – Tooltip string, if any. May be replaced by the disable reason.
- **extra_css_class** (*str*) – Extra CSS class(es)

base_css_classes = ()

render (*request*)

```
class shoop.admin.toolbar.DropdownHeader(text='', icon=None, disable_reason=None,
                                          tooltip=None, extra_css_class='btn-default')
    Bases: shoop.admin.toolbar.BaseActionButton
```

Header for DropdownActionButtons.

Parameters

- **text** – The actual text for the button.
- **icon** – Icon CSS class string
- **disable_reason** (*str/None*) – The reason for this button to be disabled. It's considered good UX to have an user-visible reason for disabled actions; thus the only way to disable an action is to set the reason. See <http://stackoverflow.com/a/372503/51685>.

- **tooltip** (*str/None*) – Tooltip string, if any. May be replaced by the disable reason.
- **extra_css_class** (*str*) – Extra CSS class(es)

base_css_classes = ()

render (*request*)

class `shoop.admin.toolbar.ButtonGroup`

Bases: `list`

render (*request*)

class `shoop.admin.toolbar.Toolbar`

Bases: `list`

render (*request*)

render_to_string (*request*)

`shoop.admin.toolbar.try_reverse` (*viewname*, ***kwargs*)

`shoop.admin.toolbar.get_discard_button` (*discard_url*)

`shoop.admin.toolbar.get_default_edit_toolbar` (*view_object*, *save_form_id*, *discard_url=None*, *delete_url=None*, *with_split_save=True*, *toolbar=None*)

Get a toolbar with buttons used for object editing.

Parameters

- **view_object** (*django.views.generic.UpdateView*) – The class-based-view object requiring the toolbar
- **save_form_id** (*str*) – The DOM ID to target for the save button
- **discard_url** (*str/None*) – The URL/route name for the Discard button. Falsy values default to the request URL.
- **delete_url** (*str/None*) – The URL/route name for the Delete button. If this is not set, the delete button is not shown.
- **with_split_save** (*bool*) – Use split delete button with “Save and Exit” etc.?
- **toolbar** (*Toolbar*) – The toolbar to augment. If None, a new one is created.

Returns `Toolbar`

Return type `Toolbar`

`shoop.admin.urls` module

`shoop.admin.urls.login` (*request*, ***kwargs*)

`shoop.admin.urls.get_urls` ()

Module contents

class `shoop.admin.ShoopAdminAppConfig` (**args*, ***kwargs*)

Bases: `shoop.apps.AppConfig`

name = ‘shoop.admin’

verbose_name = ‘Shoop Admin’

label = ‘shoop_admin’

required_installed_apps = [‘bootstrap3’]

```

provides = {'service_behavior_component_form_part': ['shoop.admin.modules.services.weight_based_pricing.WeightB
ready ()

```

shoop.api package

Submodules

shoop.api.encoders module

```

class shoop.api.encoders.ExtJSONEncoder (skipkeys=False, ensure_ascii=True,
                                         check_circular=True, allow_nan=True,
                                         sort_keys=False, indent=None, separators=None,
                                         default=None)
Bases: rest_framework.utils.encoders.JSONEncoder

```

Constructor for JSONEncoder, with sensible defaults.

If skipkeys is false, then it is a TypeError to attempt encoding of keys that are not str, int, float or None. If skipkeys is True, such items are simply skipped.

If ensure_ascii is true, the output is guaranteed to be str objects with all incoming non-ASCII characters escaped. If ensure_ascii is false, the output can contain non-ASCII characters.

If check_circular is true, then lists, dicts, and custom encoded objects will be checked for circular references during encoding to prevent an infinite recursion (which would cause an OverflowError). Otherwise, no such check takes place.

If allow_nan is true, then NaN, Infinity, and -Infinity will be encoded as such. This behavior is not JSON specification compliant, but is consistent with most JavaScript based encoders and decoders. Otherwise, it will be a ValueError to encode such floats.

If sort_keys is true, then the output of dictionaries will be sorted by key; this is useful for regression tests to ensure that JSON serializations can be compared on a day-to-day basis.

If indent is a non-negative integer, then JSON array elements and object members will be pretty-printed with that indent level. An indent level of 0 will only insert newlines. None is the most compact representation.

If specified, separators should be an (item_separator, key_separator) tuple. The default is (' ', ' ') if indent is None and (', ', ': ') otherwise. To get the most compact JSON representation, you should specify (', ', ': ') to eliminate whitespace.

If specified, default is a function that gets called for objects that can't otherwise be serialized. It should return a JSON encodable version of the object or raise a TypeError.

default (*obj*)

```

shoop.api.encoders.apply_monkeypatch ()

```

shoop.api.factories module

```

shoop.api.factories.serializer_factory (model, serializer_class=None, attrs=None,
                                         meta=None)
shoop.api.factories.viewset_factory (model, viewset_class=<class
                                     'rest_framework.viewsets.ModelViewSet'>, **attrs)

```

shoop.api.urls module

```

class shoop.api.urls.AutoRouter (trailing_slash=True)
Bases: rest_framework.routers.DefaultRouter

```

```
populate()
```

Module contents

```
class shoop.api.ShoopApiAppConfig(*args, **kwargs)
    Bases: shoop.apps.AppConfig

    name = 'shoop.api'

    verbose_name = 'Shoop API'

    label = 'shoop_api'

    required_installed_apps = ('rest_framework',)

    ready()
```

shoop.apps package

Submodules

shoop.apps.provides module This module contains the API to deal with the Provides system.

The Provides system is Shoop's mechanism for discovering and loading components, both first-party and third-party.

See also:

See [The Provides system](#) for further information about the Provides system.

```
shoop.apps.provides.clear_provides_cache()
```

```
shoop.apps.provides.get_provide_specs_and_objects(category)
    Get a mapping of provide specs ("x.y.z:Q") to their loaded objects (<class Q>).
```

Parameters `category` (*str*) – Category to load objects for

Returns Dict of spec -> object

Return type dict[str, object]

```
shoop.apps.provides.get_provide_objects(category)
    Get an iterable of provide objects for the given category.
```

Parameters `category` (*str*) – Category to load objects for

Returns Iterable of objects

Return type Iterable[object]

```
shoop.apps.provides.get_identifier_to_spec_map(category)
```

```
shoop.apps.provides.get_identifier_to_object_map(category)
```

```
shoop.apps.provides.override_provides(category, spec_list)
    Context manager to override provides for a given category.
```

Useful for testing.

Parameters

- `category` (*str*) – Category name.
- `spec_list` (*list[str]*) – List of specs.

`shoop.apps.provides.load_module(setting_name, provide_category)`

Load a module from a module setting.

The value of the setting must be a module identifier for the given provide category.

Parameters

- **setting_name** (*str*) – The setting name for the identifier
- **provide_category** (*str*) – The provide category for the identifier lookup (e.g. `tax_module`)

Returns An object.

Return type Any

`shoop.apps.provides.load_modules(setting_name, provide_category)`

Load a list of modules from a module setting.

The value of the setting must be a list of module identifiers for the given provide category.

The modules are returned in the same order they are declared in the settings.

Parameters

- **setting_name** (*str*) – The setting name for the identifier list
- **provide_category** (*str*) – The provide category for the identifier lookup (e.g. `tax_module`)

Returns A list of objects

Return type list[Any]

`shoop.apps.provides.load_module_instances(setting_name, provide_category)`

Load a list of initialized modules from a module setting.

Basically does the same as `load_modules`, but also initializes the loaded modules by calling them.

shoop.apps.settings module

`shoop.apps.settings.collect_settings_from_app(app_config)`

`shoop.apps.settings.collect_settings(app_name, settings_module)`

`shoop.apps.settings.get_known_settings()`

Get all settings known to Shoop.

Return type Iterable[Setting]

class `shoop.apps.settings.Setting(name, default, app_name, module)`

Bases: `object`

`shoop.apps.settings.validate_templates_configuration()`

Validate the TEMPLATES configuration in the Django settings.

Shoop's admin and default frontend require some Django-Jinja configuration, so let's make sure clients configure their projects correctly.

Raises `ImproperlyConfigured` if the configuration does not seem valid.

Returns

Return type

`shoop.apps.settings.reload_apps()`

Module contents

Shoop Application API Every Shoop Application should define an app config class derived from `shoop.apps.AppConfig`.

Settings To define settings for a Shoop Application, add a `settings.py` file to your app and define each setting as a module level variable with uppercase name. The values of these setting variables will be used as the default values for the settings. To document a setting, add a special comment block using `#:` prefixed lines just before the setting assignment line.

Default values can then be changed normally by defining the changed value in your Django settings module. To read a value of a setting use the `django.conf.settings` interface.

For example, if a fancy app lives in a Python package named `fancyapp`, its settings will be in module `fancyapp.settings` and if it contains something like this

```
#: Number of donuts to use
#:
#: Must be less than 42.
FANCYAPP_NUMBER_OF_DONUTS = 3
```

then this would define a setting `FANCYAPP_NUMBER_OF_DONUTS` with a default value of 3.

See also source code of `shoop.core.settings`.

Naming Settings Applications in *Shoop Base* distribution should use the following rules for naming their settings.

1. Each setting should be prefixed with the string `SHOOP_`
2. Boolean toggle settings should have a verb in imperative mood as part of the name, e.g. `SHOOP_ALLOW_ANONYMOUS_ORDERS`, `SHOOP_ENABLE_ATTRIBUTES` or `SHOOP_ENABLE_MULTIPLE_SHOPS`.
3. Setting that is used to locate a replaceable module should have suffix `_SPEC` or `_SPECS` (if the setting is a list or mapping of those), e.g. `SHOOP_PRICING_MODULE_SPEC`.
4. Setting names do NOT have to be prefixed with the application name. For example, `SHOOP_BASKET_VIEW_SPEC` which is not prefixed with `SHOOP_FRONT` even though it is from `shoop.front` application.
5. Setting names should be unique; if two applications define a setting with a same name, they cannot be enabled in the same installation.

Warning: When you have a settings file `your_app/settings.py`, do not import Django's settings in `your_app/__init__.py` with

```
from django.conf import settings
```

since that will make `your_app.settings` ambiguous. It may point to `django.conf.settings` when `your_app.settings` is not yet imported, or when it is imported, it will point to module defined by `your_app/settings.py`.

```
class shoop.apps.AppConfig(*args, **kwargs)
    Bases: django.apps.config.AppConfig

    default_settings_module = 'settings'
        Name of the settings module for this app
```

required_installed_apps = ()

Apps that are required to be in INSTALLED_APPS for this app

This may also be a dict of the form {app_name: reason} where the reason will then be listed in the ImproperlyConfigured exception.

provides = {}

See *The Provides system* for details about the provides variable.

get_default_settings_module ()

Get default settings module.

Returns the settings module

Raises ImportError if no such module exists

shoop.apps.get_known_settings ()

Get all settings known to Shoop.

Return type Iterable[Setting]

shoop.campaigns package

Subpackages

shoop.campaigns.admin_module package

Subpackages

shoop.campaigns.admin_module.forms package

Module contents

class shoop.campaigns.admin_module.forms.**BasketCampaignForm** (*args, **kwargs)

Bases: shoop.campaigns.admin_module.forms._base.BaseCampaignForm

class **Meta**

Bases: shoop.campaigns.admin_module.forms._base.Meta

model

alias of BasketCampaign

BasketCampaignForm.**base_fields** = OrderedDict([('shop', <django.forms.models.ModelChoiceField object at 0x7...

BasketCampaignForm.**clean_coupon** ()

BasketCampaignForm.**declared_fields** = OrderedDict([('public_name', <django.forms.fields.CharField object at 0x7...

BasketCampaignForm.**media**

class shoop.campaigns.admin_module.forms.**BasketDiscountAmountForm** (**kwargs)

Bases: shoop.campaigns.admin_module.forms._base.BaseEffectModelForm

class **Meta**

Bases: shoop.campaigns.admin_module.forms._base.Meta

model

alias of BasketDiscountAmount

BasketDiscountAmountForm.**base_fields** = OrderedDict([('campaign', <django.forms.models.ModelChoiceField...

```
BasketDiscountAmountForm.declared_fields = OrderedDict()

BasketDiscountAmountForm.media

class shoop.campaigns.admin_module.forms.BasketDiscountPercentageForm (**kwargs)
    Bases: shoop.campaigns.admin_module.forms._base.BaseEffectModelForm

    class Meta
        Bases: shoop.campaigns.admin_module.forms._base.Meta

        model
            alias of BasketDiscountPercentage

    BasketDiscountPercentageForm.base_fields = OrderedDict([('campaign', <django.forms.models.ModelChoiceForm
    BasketDiscountPercentageForm.declared_fields = OrderedDict([('discount_percentage', <shoop.admin.form
    BasketDiscountPercentageForm.media

class shoop.campaigns.admin_module.forms.BasketMaxTotalAmountConditionForm (**kwargs)
    Bases: shoop.campaigns.admin_module.forms._base.BaseRuleModelForm

    class Meta
        Bases: shoop.campaigns.admin_module.forms._base.Meta

        model
            alias of BasketMaxTotalAmountCondition

    BasketMaxTotalAmountConditionForm.base_fields = OrderedDict([('amount_value', <shoop.core.fields.Form
    BasketMaxTotalAmountConditionForm.declared_fields = OrderedDict()
    BasketMaxTotalAmountConditionForm.media

class shoop.campaigns.admin_module.forms.BasketMaxTotalProductAmountConditionForm (**kwargs)
    Bases: shoop.campaigns.admin_module.forms._base.BaseRuleModelForm

    class Meta
        Bases: shoop.campaigns.admin_module.forms._base.Meta

        model
            alias of BasketMaxTotalProductAmountCondition

    BasketMaxTotalProductAmountConditionForm.base_fields = OrderedDict([('product_count', <django.form
    BasketMaxTotalProductAmountConditionForm.declared_fields = OrderedDict()
    BasketMaxTotalProductAmountConditionForm.media

class shoop.campaigns.admin_module.forms.BasketTotalAmountConditionForm (**kwargs)
    Bases: shoop.campaigns.admin_module.forms._base.BaseRuleModelForm

    class Meta
        Bases: shoop.campaigns.admin_module.forms._base.Meta

        model
            alias of BasketTotalAmountCondition

    BasketTotalAmountConditionForm.base_fields = OrderedDict([('amount_value', <shoop.core.fields.Formatte
    BasketTotalAmountConditionForm.declared_fields = OrderedDict()
    BasketTotalAmountConditionForm.media

class shoop.campaigns.admin_module.forms.BasketTotalProductAmountConditionForm (**kwargs)
    Bases: shoop.campaigns.admin_module.forms._base.BaseRuleModelForm
```

```

class Meta
    Bases: shoop.campaigns.admin_module.forms._base.Meta

    model
        alias of BasketTotalProductAmountCondition

BasketTotalProductAmountConditionForm.base_fields = OrderedDict([('product_count', <django.forms.fields.IntegerField object at 0x7f8b1b1b1b1b>)]

BasketTotalProductAmountConditionForm.declared_fields = OrderedDict()

BasketTotalProductAmountConditionForm.media

class shoop.campaigns.admin_module.forms.CatalogCampaignForm(**kwargs)
    Bases: shoop.campaigns.admin_module.forms._base.BaseCampaignForm

    class Meta
        Bases: shoop.campaigns.admin_module.forms._base.Meta

        exclude = ['identifier', 'created_by', 'modified_by', 'conditions', 'filters', 'coupon']

        model
            alias of CatalogCampaign

CatalogCampaignForm.base_fields = OrderedDict([('shop', <django.forms.models.ModelChoiceField object at 0x7f8b1b1b1b1b>)]

CatalogCampaignForm.declared_fields = OrderedDict([('public_name', <django.forms.fields.CharField object at 0x7f8b1b1b1b1b>)]

CatalogCampaignForm.media

class shoop.campaigns.admin_module.forms.CategoryFilterForm(**kwargs)
    Bases: shoop.campaigns.admin_module.forms._base.BaseRuleModelForm

    class Meta
        Bases: shoop.campaigns.admin_module.forms._base.Meta

        model
            alias of CategoryFilter

CategoryFilterForm.base_fields = OrderedDict([('categories', <django.forms.models.ModelMultipleChoiceField object at 0x7f8b1b1b1b1b>)]

CategoryFilterForm.declared_fields = OrderedDict()

CategoryFilterForm.media

class shoop.campaigns.admin_module.forms.ContactBasketConditionForm(**kwargs)
    Bases: shoop.campaigns.admin_module.forms._base.BaseRuleModelForm

    class Meta
        Bases: shoop.campaigns.admin_module.forms._base.Meta

        model
            alias of ContactBasketCondition

ContactBasketConditionForm.base_fields = OrderedDict([('contacts', <django.forms.models.ModelMultipleChoiceField object at 0x7f8b1b1b1b1b>)]

ContactBasketConditionForm.declared_fields = OrderedDict()

ContactBasketConditionForm.media

class shoop.campaigns.admin_module.forms.ContactConditionForm(**kwargs)
    Bases: shoop.campaigns.admin_module.forms._base.BaseRuleModelForm

    class Meta
        Bases: shoop.campaigns.admin_module.forms._base.Meta

        model
            alias of ContactCondition

```

```
ContactConditionForm.base_fields = OrderedDict([('contacts', <django.forms.models.ModelMultipleChoiceField object at 0x7f713b447400>)])  
ContactConditionForm.declared_fields = OrderedDict()  
ContactConditionForm.media  
  
class shoop.campaigns.admin_module.forms.ContactGroupBasketConditionForm (**kwargs)  
    Bases: shoop.campaigns.admin_module.forms._base.BaseRuleModelForm  
  
    class Meta  
        Bases: shoop.campaigns.admin_module.forms._base.Meta  
  
        model  
            alias of ContactGroupBasketCondition  
  
ContactGroupBasketConditionForm.base_fields = OrderedDict([('contact_groups', <django.forms.models.ModelMultipleChoiceField object at 0x7f713b447400>)])  
ContactGroupBasketConditionForm.declared_fields = OrderedDict()  
ContactGroupBasketConditionForm.media  
  
class shoop.campaigns.admin_module.forms.ContactGroupConditionForm (**kwargs)  
    Bases: shoop.campaigns.admin_module.forms._base.BaseRuleModelForm  
  
    class Meta  
        Bases: shoop.campaigns.admin_module.forms._base.Meta  
  
        model  
            alias of ContactGroupCondition  
  
ContactGroupConditionForm.base_fields = OrderedDict([('contact_groups', <django.forms.models.ModelMultipleChoiceField object at 0x7f713b447400>)])  
ContactGroupConditionForm.declared_fields = OrderedDict()  
ContactGroupConditionForm.media  
  
class shoop.campaigns.admin_module.forms.CouponForm (*args, **kwargs)  
    Bases: django.forms.models.ModelForm  
  
    class Meta  
        Bases: object  
  
        fields = ['code', 'usage_limit_customer', 'usage_limit', 'active']  
  
        model  
            alias of Coupon  
  
CouponForm.base_fields = OrderedDict([('code', <django.forms.fields.CharField object at 0x7f713b447400>), ('usage_limit_customer', <django.forms.fields.IntegerField object at 0x7f713b447400>), ('usage_limit', <django.forms.fields.IntegerField object at 0x7f713b447400>), ('active', <django.forms.fields.BooleanField object at 0x7f713b447400>)])  
CouponForm.clean_code ()  
CouponForm.declared_fields = OrderedDict([('autogenerate', <django.forms.fields.BooleanField object at 0x7f713b447400>)])  
CouponForm.media  
  
class shoop.campaigns.admin_module.forms.DiscountFromProductForm (**kwargs)  
    Bases: shoop.campaigns.admin_module.forms._base.BaseEffectModelForm  
  
    class Meta  
        Bases: shoop.campaigns.admin_module.forms._base.Meta  
  
        model  
            alias of DiscountFromProduct  
  
DiscountFromProductForm.base_fields = OrderedDict([('campaign', <django.forms.models.ModelChoiceField object at 0x7f713b447400>)])  
DiscountFromProductForm.declared_fields = OrderedDict()
```

```

DiscountFromProductForm.media

class shoop.campaigns.admin_module.forms.FreeProductLineForm(**kwargs)
    Bases: shoop.campaigns.admin_module.forms._base.BaseEffectModelForm

    class Meta
        Bases: shoop.campaigns.admin_module.forms._base.Meta

        model
            alias of FreeProductLine

    FreeProductLineForm.base_fields = OrderedDict([('campaign', <django.forms.models.ModelChoiceField object>)]
    FreeProductLineForm.declared_fields = OrderedDict()
    FreeProductLineForm.media

class shoop.campaigns.admin_module.forms.ProductDiscountAmountForm(**kwargs)
    Bases: shoop.campaigns.admin_module.forms._base.BaseEffectModelForm

    class Meta
        Bases: shoop.campaigns.admin_module.forms._base.Meta

        exclude = ['identifier', 'active']

        model
            alias of ProductDiscountAmount

    ProductDiscountAmountForm.base_fields = OrderedDict([('campaign', <django.forms.models.ModelChoiceField object>)]
    ProductDiscountAmountForm.declared_fields = OrderedDict()
    ProductDiscountAmountForm.media

class shoop.campaigns.admin_module.forms.ProductDiscountPercentageForm(**kwargs)
    Bases: shoop.campaigns.admin_module.forms._base.BaseEffectModelForm

    class Meta
        Bases: shoop.campaigns.admin_module.forms._base.Meta

        exclude = ['identifier', 'active']

        model
            alias of ProductDiscountPercentage

    ProductDiscountPercentageForm.base_fields = OrderedDict([('campaign', <django.forms.models.ModelChoiceField object>)]
    ProductDiscountPercentageForm.declared_fields = OrderedDict([('discount_percentage', <shoop.admin_module.forms.ProductDiscountPercentageForm object>)]
    ProductDiscountPercentageForm.media

class shoop.campaigns.admin_module.forms.ProductFilterForm(**kwargs)
    Bases: shoop.campaigns.admin_module.forms._base.BaseRuleModelForm

    class Meta
        Bases: shoop.campaigns.admin_module.forms._base.Meta

        model
            alias of ProductFilter

    ProductFilterForm.base_fields = OrderedDict([('products', <django.forms.models.ModelMultipleChoiceField object>)]
    ProductFilterForm.declared_fields = OrderedDict()
    ProductFilterForm.media

```

```
class shoop.campaigns.admin_module.forms.ProductsInBasketConditionForm (**kwargs)
    Bases: shoop.campaigns.admin_module.forms._base.BaseRuleModelForm

    class Meta
        Bases: shoop.campaigns.admin_module.forms._base.Meta

        model
            alias of ProductsInBasketCondition

    ProductsInBasketConditionForm.base_fields = OrderedDict([('quantity', <django.forms.fields.IntegerField object>)]
    ProductsInBasketConditionForm.declared_fields = OrderedDict()
    ProductsInBasketConditionForm.media

class shoop.campaigns.admin_module.forms.ProductTypeFilterForm (**kwargs)
    Bases: shoop.campaigns.admin_module.forms._base.BaseRuleModelForm

    class Meta
        Bases: shoop.campaigns.admin_module.forms._base.Meta

        model
            alias of ProductTypeFilter

    ProductTypeFilterForm.base_fields = OrderedDict([('product_types', <django.forms.models.ModelMultipleChoiceField object>)]
    ProductTypeFilterForm.declared_fields = OrderedDict()
    ProductTypeFilterForm.media
```

shoop.campaigns.admin_module.views package

Module contents

```
class shoop.campaigns.admin_module.views.CatalogCampaignEditView (**kwargs)
    Bases: shoop.campaigns.utils._Breadcrumbed, shoop.campaigns.admin_module.views._edit.CampaignEditView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    base_form_part_classes = [<class 'shoop.campaigns.admin_module.form_parts.CatalogBaseFormPart'>]
    condition_key = 'campaign_context_condition'
    effects = [('campaign_product_discount_effect_form', <class 'shoop.campaigns.admin_module.form_parts.CatalogEffectForm'>)]
    filter_key = 'campaign_catalog_filter'
    get_form_parts (object)
    model
        alias of CatalogCampaign

    parent_name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    parent_url = 'shoop_admin:catalog_campaigns.list'
    rules_form_part_class
        alias of CatalogConditionsFormPart

class shoop.campaigns.admin_module.views.CatalogCampaignListView (**kwargs)
    Bases: shoop.campaigns.admin_module.views._list.CampaignListView

    get_context_data (**kwargs)
    model
        alias of CatalogCampaign
```



```

class shoop.campaigns.admin_module.views.BasketCampaignEditView (**kwargs)
    Bases: shoop.campaigns.utils._Breadcrumbed, shoop.campaigns.admin_module.views._edit.CampaignEditView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    base_form_part_classes = [<class 'shoop.campaigns.admin_module.form_parts.BasketBaseFormPart'>]
    condition_key = 'campaign_basket_condition'
    effects = [('campaign_basket_discount_effect_form', <class 'shoop.campaigns.admin_module.form_parts.BasketDiscountFormPart'>)]
    model
        alias of BasketCampaign
    parent_name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    parent_url = 'shoop_admin:basket_campaigns.list'
    rules_form_part_class
        alias of BasketConditionsFormPart

class shoop.campaigns.admin_module.views.BasketCampaignListView (**kwargs)
    Bases: shoop.campaigns.admin_module.views._list.CampaignListView
    get_context_data (**kwargs)
    model
        alias of BasketCampaign

class shoop.campaigns.admin_module.views.CouponEditView (**kwargs)
    Bases: shoop.campaigns.utils._Breadcrumbed, shoop.admin.utils.views.CreateOrUpdateView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    add_form_errors_as_messages = True
    context_object_name = 'coupon'
    form_class
        alias of CouponForm
    model
        alias of Coupon
    parent_name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    parent_url = 'shoop_admin:coupons.list'
    template_name = 'shoop/campaigns/admin/edit_coupons.jinja'

class shoop.campaigns.admin_module.views.CouponListView (**kwargs)
    Bases: shoop.admin.utils.views.PicortableListView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    columns = [<shoop.admin.utils.picortable.Column object at 0x7f713b4b4550>, <shoop.admin.utils.picortable.Column object at 0x7f713b4b4550>]
    get_context_data (**kwargs)
    get_usages (instance, *args, **kwargs)
    model
        alias of Coupon

```

Submodules

shoop.campaigns.admin_module.form_parts module

```
class shoop.campaigns.admin_module.form_parts.SalesRangesForm (**kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        model
            alias of ContactGroupSalesRange

        fields = ['min_value', 'max_value']

        labels = {'min_value': <django.utils.functional.lazy.<locals>.__proxy__ object at 0x7f713bcef8d0>, 'max_value': <
        help_texts = {'max_value': <django.utils.functional.lazy.<locals>.__proxy__ object at 0x7f713bcef860>}

    SalesRangesForm.base_fields = OrderedDict([('min_value', <shoop.core.fields.FormattedDecimalFormField obje
    SalesRangesForm.declared_fields = OrderedDict()

    SalesRangesForm.media
class shoop.campaigns.admin_module.form_parts.SalesRangesFormPart (request, ob-
                                                ject=None)

    Bases: shoop.admin.form_part.FormPart

    priority = 3

    name = 'contact_group_sales_ranges'

    form
        alias of SalesRangesForm

    get_form_defs()

    form_valid(form)

class shoop.campaigns.admin_module.form_parts.CampaignBaseFormPart (*args,
                                                                    **kwargs)

    Bases: shoop.admin.form_part.FormPart

    priority = -1000

    form = None

    get_form_defs()

    form_valid(form)

class shoop.campaigns.admin_module.form_parts.CatalogBaseFormPart (*args,
                                                                    **kwargs)

    Bases: shoop.campaigns.admin_module.form_parts.CampaignBaseFormPart

    form
        alias of CatalogCampaignForm

class shoop.campaigns.admin_module.form_parts.BasketBaseFormPart (*args,
                                                                    **kwargs)

    Bases: shoop.campaigns.admin_module.form_parts.CampaignBaseFormPart

    form
        alias of BasketCampaignForm

class shoop.campaigns.admin_module.form_parts.BaseFormPart (request, form, name,
                                                                owner)

    Bases: shoop.admin.form_part.FormPart

    formset = None
```

```

    template_name = 'shoop/campaigns/admin/_edit_form.jinja'
    get_form_defs()
    form_valid(form)
class shoop.campaigns.admin_module.form_parts.BasketConditionsFormPart (request,
                                                                    form,
                                                                    name,
                                                                    owner)

    Bases: shoop.campaigns.admin_module.form_parts.BaseFormPart

    formset
        alias of BasketConditionsFormSet
class shoop.campaigns.admin_module.form_parts.BasketDiscountEffectsFormPart (request,
                                                                    form,
                                                                    name,
                                                                    owner)

    Bases: shoop.campaigns.admin_module.form_parts.BaseFormPart

    formset
        alias of BasketDiscountEffectsFormSet
class shoop.campaigns.admin_module.form_parts.BasketLineEffectsFormPart (request,
                                                                    form,
                                                                    name,
                                                                    owner)

    Bases: shoop.campaigns.admin_module.form_parts.BaseFormPart

    formset
        alias of BasketLineEffectsFormSet
class shoop.campaigns.admin_module.form_parts.CatalogConditionsFormPart (request,
                                                                    form,
                                                                    name,
                                                                    owner)

    Bases: shoop.campaigns.admin_module.form_parts.BaseFormPart

    formset
        alias of CatalogConditionsFormSet
class shoop.campaigns.admin_module.form_parts.CatalogFiltersFormPart (request,
                                                                    form,
                                                                    name,
                                                                    owner)

    Bases: shoop.campaigns.admin_module.form_parts.BaseFormPart

    formset
        alias of CatalogFiltersFormSet
class shoop.campaigns.admin_module.form_parts.CatalogEffectsFormPart (request,
                                                                    form,
                                                                    name,
                                                                    owner)

    Bases: shoop.campaigns.admin_module.form_parts.BaseFormPart

    formset
        alias of CatalogEffectsFormSet

```

shoop.campaigns.admin_module.form_sets module

```
class shoop.campaigns.admin_module.form_sets.BaseFormset (*args, **kwargs)
    Bases: django.forms.models.BaseModelFormSet

    model = None

    validate_min = False

    min_num = 0

    validate_max = False

    max_num = 20

    absolute_max = 20

    can_delete = True

    can_order = False

    extra = 0

    form_class = None

    get_name ()

    get_queryset ()

    form (**kwargs)
class shoop.campaigns.admin_module.form_sets.BasketConditionsFormSet (*args,
                                                                    **kwargs)
    Bases: shoop.campaigns.admin_module.form_sets.BaseFormset

    model
        alias of BasketCondition

    get_queryset ()

class shoop.campaigns.admin_module.form_sets.EffectsFormset (*args, **kwargs)
    Bases: shoop.campaigns.admin_module.form_sets.BaseFormset

    form (**kwargs)
class shoop.campaigns.admin_module.form_sets.BasketDiscountEffectsFormSet (*args,
                                                                    **kwargs)
    Bases: shoop.campaigns.admin_module.form_sets.EffectsFormset

    model
        alias of BasketDiscountEffect

    get_queryset ()

class shoop.campaigns.admin_module.form_sets.BasketLineEffectsFormSet (*args,
                                                                    **kwargs)
    Bases: shoop.campaigns.admin_module.form_sets.EffectsFormset

    model
        alias of BasketLineEffect

    get_queryset ()

class shoop.campaigns.admin_module.form_sets.CatalogConditionsFormSet (*args,
                                                                    **kwargs)
    Bases: shoop.campaigns.admin_module.form_sets.BaseFormset

    model
        alias of ContextCondition

    get_queryset ()
```

```

class shoop.campaigns.admin_module.form_sets.CatalogFiltersFormSet (*args,
                                                                    **kwargs)
    Bases: shoop.campaigns.admin_module.form_sets.BaseFormset
    model
        alias of CatalogFilter
    get_queryset ()

class shoop.campaigns.admin_module.form_sets.CatalogEffectsFormSet (*args,
                                                                    **kwargs)
    Bases: shoop.campaigns.admin_module.form_sets.EffectsFormset
    model
        alias of ProductDiscountEffect
    get_queryset ()

```

Module contents

```

class shoop.campaigns.admin_module.CampaignAdminModule
    Bases: shoop.admin.base.AdminModule
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    get_urls ()
    get_menu_category_icons ()
    get_menu_entries (request)
    get_required_permissions ()
    get_model_url (object, kind)

```

shoop.campaigns.models package

Submodules

shoop.campaigns.models.basket_conditions module

```

class shoop.campaigns.models.basket_conditions.BasketCondition (id,
                                                                polymorphic_ctype, active)
    Bases: polymorphic.models.PolymorphicModel

```

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

```

model = None
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
matches (basket, lines)

```

```
exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception BasketCondition.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

BasketCondition.base_objects = <django.db.models.manager.Manager object>

BasketCondition.basketmaxtotalamountcondition

BasketCondition.basketmaxtotalproductamountcondition

BasketCondition.basketttotalamountcondition

BasketCondition.basketttotalproductamountcondition

BasketCondition.campaign

BasketCondition.contactbasketcondition

BasketCondition.contactgroupbasketcondition

BasketCondition.objects = <polymorphic.managers.PolymorphicManager object>

BasketCondition.polymorphic_ctype

BasketCondition.polymorphic_primary_key_name = 'id'

BasketCondition.polymorphic_super_sub_accessors_replaced = False

BasketCondition.productsinbasketcondition
class shoop.campaigns.models.basket_conditions.BasketTotalProductAmountCondition (id,
                                                                                     poly-
                                                                                     mor-
                                                                                     phic_ctype,
                                                                                     ac-
                                                                                     tive,
                                                                                     bas-
                                                                                     ket-
                                                                                     con-
                                                                                     di-
                                                                                     tion_ptr,
                                                                                     prod-
                                                                                     uct_count)

Bases: shoop.campaigns.models.basket_conditions.BasketCondition

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions.
We monkey patch them until a patch can be added to Django (which would probably be very small and make all
of this obsolete).

If we have inheritance of the form ModelA -> ModelB ->ModelC then Django creates accessors like this: -
ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at
hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our
own accessors that use our appropriate base_objects manager.

identifier = 'basket_product_condition'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

matches (basket, lines)
```

```

description
value
exception DoesNotExist
    Bases: shoop.campaigns.models.basket_conditions.DoesNotExist
exception BasketTotalProductAmountCondition.MultipleObjectsReturned
    Bases: shoop.campaigns.models.basket_conditions.MultipleObjectsReturned
BasketTotalProductAmountCondition.base_objects = <django.db.models.manager.Manager object>
BasketTotalProductAmountCondition.basketcondition_ptr
BasketTotalProductAmountCondition.objects = <polymorphic.managers.PolymorphicManager object>
BasketTotalProductAmountCondition.polymorphic_primary_key_name = 'id'
BasketTotalProductAmountCondition.polymorphic_super_sub_accessors_replaced = False
class shoop.campaigns.models.basket_conditions.BasketTotalAmountCondition (id,
                                                                    poly-
                                                                    mor-
                                                                    phic_ctype,
                                                                    ac-
                                                                    tive,
                                                                    bas-
                                                                    ket-
                                                                    con-
                                                                    di-
                                                                    tion_ptr,
                                                                    amount_value)
    Bases: shoop.utils.properties.MoneyPropped, shoop.campaigns.models.basket_conditions.BasketTotalProductAmountCondition

    identifier = 'basket_amount_condition'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    amount
        Property for Price object.

        Similar to MoneyProperty but also has includes_tax field.

        Operators with TaxfulPrice and TaxlessPrice objects.
    matches (basket, lines)
    description
    value
    exception DoesNotExist
        Bases: shoop.campaigns.models.basket_conditions.DoesNotExist
    exception BasketTotalAmountCondition.MultipleObjectsReturned
        Bases: shoop.campaigns.models.basket_conditions.MultipleObjectsReturned
BasketTotalAmountCondition.base_objects = <django.db.models.manager.Manager object>
BasketTotalAmountCondition.basketcondition_ptr
BasketTotalAmountCondition.objects = <polymorphic.managers.PolymorphicManager object>
BasketTotalAmountCondition.polymorphic_primary_key_name = 'id'

```

```
BasketTotalAmountCondition.polymorphic_super_sub_accessors_replaced = False
class shoop.campaigns.models.basket_conditions.BasketMaxTotalProductAmountCondition (id,
                                                                                       poly-
                                                                                       mor-
                                                                                       phic_ctype,
                                                                                       ac-
                                                                                       tive,
                                                                                       bas-
                                                                                       ket-
                                                                                       con-
                                                                                       di-
                                                                                       tion_ptr,
                                                                                       prod-
                                                                                       uct_count)
```

Bases: *shoop.campaigns.models.basket_conditions.BasketCondition*

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

identifier = 'basket_max_product_condition'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

matches (*basket, lines*)

description

value

exception DoesNotExist

Bases: shoop.campaigns.models.basket_conditions.DoesNotExist

exception BasketMaxTotalProductAmountCondition.MultipleObjectsReturned

Bases: shoop.campaigns.models.basket_conditions.MultipleObjectsReturned

BasketMaxTotalProductAmountCondition.**base_objects** = <django.db.models.manager.Manager object>

BasketMaxTotalProductAmountCondition.**basketcondition_ptr**

BasketMaxTotalProductAmountCondition.**objects** = <polymorphic.managers.PolymorphicManager object>

BasketMaxTotalProductAmountCondition.**polymorphic_primary_key_name** = 'id'

BasketMaxTotalProductAmountCondition.**polymorphic_super_sub_accessors_replaced** = False


```

class shoop.campaigns.models.basket_conditions.BasketMaxTotalAmountCondition (id,
                                                                    poly-
                                                                    mor-
                                                                    phic_ctype,
                                                                    ac-
                                                                    tive,
                                                                    bas-
                                                                    ket-
                                                                    con-
                                                                    di-
                                                                    tion_ptr,
                                                                    amount_value)

Bases: shoop.utils.properties.MoneyPropped, shoop.campaigns.models.basket_conditions.BasketCondition

identifier = 'basket_max_amount_condition'
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
amount
    Property for Price object.
    Similar to MoneyProperty but also has includes_tax field.
    Operators with TaxfulPrice and TaxlessPrice objects.
matches (basket, lines)
description
value
exception DoesNotExist
    Bases: shoop.campaigns.models.basket_conditions.DoesNotExist
exception BasketMaxTotalAmountCondition.MultipleObjectsReturned
    Bases: shoop.campaigns.models.basket_conditions.MultipleObjectsReturned
BasketMaxTotalAmountCondition.base_objects = <django.db.models.manager.Manager object>
BasketMaxTotalAmountCondition.basketcondition_ptr
BasketMaxTotalAmountCondition.objects = <polymorphic.managers.PolymorphicManager object>
BasketMaxTotalAmountCondition.polymorphic_primary_key_name = 'id'
BasketMaxTotalAmountCondition.polymorphic_super_sub_accessors_replaced = False
class shoop.campaigns.models.basket_conditions.ProductsInBasketCondition (id,
                                                                    poly-
                                                                    mor-
                                                                    phic_ctype,
                                                                    ac-
                                                                    tive,
                                                                    bas-
                                                                    ket-
                                                                    con-
                                                                    di-
                                                                    tion_ptr,
                                                                    quan-
                                                                    tity)

Bases: shoop.campaigns.models.basket_conditions.BasketCondition

```

Replace Django's inheritance accessor member functions for our model (`self.__class__`) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB ->ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

```
identifier = 'basket_products_condition'
```

```
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
```

```
model
```

```
    alias of Product
```

```
products
```

```
matches (basket, lines)
```

```
description
```

```
values
```

```
exception DoesNotExist
```

```
    Bases: shoop.campaigns.models.basket_conditions.DoesNotExist
```

```
exception ProductsInBasketCondition.MultipleObjectsReturned
```

```
    Bases: shoop.campaigns.models.basket_conditions.MultipleObjectsReturned
```

```
ProductsInBasketCondition.base_objects = <django.db.models.manager.Manager object>
```

```
ProductsInBasketCondition.basketcondition_ptr
```

```
ProductsInBasketCondition.objects = <polymorphic.managers.PolymorphicManager object>
```

```
ProductsInBasketCondition.polymorphic_primary_key_name = 'id'
```

```
ProductsInBasketCondition.polymorphic_super_sub_accessors_replaced = False
```

```
class shoop.campaigns.models.basket_conditions.ContactGroupBasketCondition (id,  
                                                                           poly-  
                                                                           mor-  
                                                                           phic_ctype,  
                                                                           ac-  
                                                                           tive,  
                                                                           bas-  
                                                                           ket-  
                                                                           con-  
                                                                           di-  
                                                                           tion_ptr)
```

```
    Bases: shoop.campaigns.models.basket_conditions.BasketCondition
```

Replace Django's inheritance accessor member functions for our model (`self.__class__`) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB ->ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

```

model
    alias of ContactGroup

identifier = 'basket_contact_group_condition'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

contact_groups

matches (basket, lines=[])

description

values

exception DoesNotExist
    Bases: shoop.campaigns.models.basket_conditions.DoesNotExist

exception ContactGroupBasketCondition.MultipleObjectsReturned
    Bases: shoop.campaigns.models.basket_conditions.MultipleObjectsReturned

ContactGroupBasketCondition.base_objects = <django.db.models.manager.Manager object>

ContactGroupBasketCondition.basketcondition_ptr

ContactGroupBasketCondition.objects = <polymorphic.managers.PolymorphicManager object>

ContactGroupBasketCondition.polymorphic_primary_key_name = 'id'

ContactGroupBasketCondition.polymorphic_super_sub_accessors_replaced = False

class shoop.campaigns.models.basket_conditions.ContactBasketCondition (id,
                                                                    polymor-
                                                                    phic_ctype,
                                                                    active,
                                                                    basket-
                                                                    condi-
                                                                    tion_ptr)

Bases: shoop.campaigns.models.basket_conditions.BasketCondition

```

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

```

model
    alias of Contact

exception DoesNotExist
    Bases: shoop.campaigns.models.basket_conditions.DoesNotExist

```

exception `ContactBasketCondition.MultipleObjectsReturned`

Bases: `shoop.campaigns.models.basket_conditions.MultipleObjectsReturned`

`ContactBasketCondition.base_objects = <django.db.models.manager.Manager object>`

`ContactBasketCondition.basketcondition_ptr`

`ContactBasketCondition.identifier = 'basket_contact_condition'`

`ContactBasketCondition.objects = <polymorphic.managers.PolymorphicManager object>`

`ContactBasketCondition.polymorphic_primary_key_name = 'id'`

`ContactBasketCondition.polymorphic_super_sub_accessors_replaced = False`

`ContactBasketCondition.name = <django.utils.functional.lazy.<locals>.__proxy__ object>`

`ContactBasketCondition.contacts`

`ContactBasketCondition.matches (basket, lines=[])`

`ContactBasketCondition.description`

`ContactBasketCondition.values`

shoop.campaigns.models.basket_effects module

class `shoop.campaigns.models.basket_effects.BasketDiscountEffect` (*id*, *polymorphic_ctype*, *campaign*)

Bases: `shoop.core.models.PolymorphicShoopModel`

Replace Django's inheritance accessor member functions for our model (`self.__class__`) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: `modelb` - ModelB: `modela_ptr`, `modelb`, `modelc` - ModelC: `modela_ptr`, `modelb`, `modelb_ptr`, `modelc`

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate `base_objects` manager.

identifier = `None`

model = `None`

admin_form_class = `None`

campaign

apply_for_basket (*order_source*)

Applies the effect based on given `order_source`

Returns amount of discount to accumulate for the product

Return type *Price*

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `BasketDiscountEffect.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`BasketDiscountEffect.base_objects = <django.db.models.manager.Manager object>`

```

BasketDiscountEffect.basketdiscountamount
BasketDiscountEffect.basketdiscountpercentage
BasketDiscountEffect.objects = <polymorphic.managers.PolymorphicManager object>
BasketDiscountEffect.polymorphic_ctype
BasketDiscountEffect.polymorphic_primary_key_name = 'id'
BasketDiscountEffect.polymorphic_super_sub_accessors_replaced = False
class shoop.campaigns.models.basket_effects.BasketDiscountAmount (id, polymorphic_ctype,
                                                                    campaign,
                                                                    basketdiscounteffect_ptr, discount_amount)

Bases: shoop.campaigns.models.basket_effects.BasketDiscountEffect

```

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

```

identifier = 'discount_amount_effect'
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
description
value
apply_for_basket (order_source)
exception DoesNotExist
    Bases: shoop.campaigns.models.basket_effects.DoesNotExist
exception BasketDiscountAmount.MultipleObjectsReturned
    Bases: shoop.campaigns.models.basket_effects.MultipleObjectsReturned
BasketDiscountAmount.base_objects = <django.db.models.manager.Manager object>
BasketDiscountAmount.basketdiscounteffect_ptr
BasketDiscountAmount.objects = <polymorphic.managers.PolymorphicManager object>
BasketDiscountAmount.polymorphic_primary_key_name = 'id'
BasketDiscountAmount.polymorphic_super_sub_accessors_replaced = False

```

```
class shoop.campaigns.models.basket_effects.BasketDiscountPercentage (id, poly-
                                                                    mor-
                                                                    phic_ctype,
                                                                    campaign,
                                                                    basketdis-
                                                                    countef-
                                                                    fect_ptr,
                                                                    dis-
                                                                    count_percentage)

Bases: shoop.campaigns.models.basket_effects.BasketDiscountEffect
```

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB ->ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

identifier = 'discount_percentage_effect'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

admin_form_class

alias of PercentageField

description

exception DoesNotExist

Bases: *shoop.campaigns.models.basket_effects.DoesNotExist*

exception BasketDiscountPercentage.MultipleObjectsReturned

Bases: *shoop.campaigns.models.basket_effects.MultipleObjectsReturned*

BasketDiscountPercentage.base_objects = <django.db.models.manager.Manager object>

BasketDiscountPercentage.basketdiscounteffect_ptr

BasketDiscountPercentage.objects = <polymorphic.managers.PolymorphicManager object>

BasketDiscountPercentage.polymorphic_primary_key_name = 'id'

BasketDiscountPercentage.polymorphic_super_sub_accessors_replaced = False

BasketDiscountPercentage.value

BasketDiscountPercentage.apply_for_basket (order_source)

shoop.campaigns.models.basket_line_effects module

```
class shoop.campaigns.models.basket_line_effects.BasketLineEffect (id, polymor-
                                                                    phic_ctype,
                                                                    campaign)

Bases: shoop.core.models.PolymorphicShoopModel
```

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB ->ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

```
identifier = None
```

```
model = None
```

```
admin_form_class = None
```

```
campaign
```

```
get_discount_lines (order_source, original_lines)
```

Applies the effect based on given order_source

Returns amount of discount to accumulate for the product

Return type Iterable[shoop.core.order_creator.SourceLine]

```
exception DoesNotExist
```

Bases: django.core.exceptions.ObjectDoesNotExist

```
exception BasketLineEffect.MultipleObjectsReturned
```

Bases: django.core.exceptions.MultipleObjectsReturned

```
BasketLineEffect.base_objects = <django.db.models.manager.Manager object>
```

```
BasketLineEffect.discountfromproduct
```

```
BasketLineEffect.freeproductline
```

```
BasketLineEffect.objects = <polymorphic.managers.PolymorphicManager object>
```

```
BasketLineEffect.polymorphic_ctype
```

```
BasketLineEffect.polymorphic_primary_key_name = 'id'
```

```
BasketLineEffect.polymorphic_super_sub_accessors_replaced = False
```

```
class shoop.campaigns.models.basket_line_effects.FreeProductLine (id,      polymor-
                                                                phic_ctype,
                                                                campaign, bas-
                                                                ketlineeffect_ptr,
                                                                quantity)
```

Bases: shoop.campaigns.models.basket_line_effects.BasketLineEffect

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB ->ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

```
identifier = 'free_product_line_effect'
```

```
model
    alias of Product

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

products

description

values

get_discount_lines (order_source, original_lines)

exception DoesNotExist
    Bases: shoop.campaigns.models.basket_line_effects.DoesNotExist

exception FreeProductLine.MultipleObjectsReturned
    Bases: shoop.campaigns.models.basket_line_effects.MultipleObjectsReturned

FreeProductLine.base_objects = <django.db.models.manager.Manager object>

FreeProductLine.basketlineeffect_ptr

FreeProductLine.objects = <polymorphic.managers.PolymorphicManager object>

FreeProductLine.polymorphic_primary_key_name = 'id'

FreeProductLine.polymorphic_super_sub_accessors_replaced = False

class shoop.campaigns.models.basket_line_effects.DiscountFromProduct (id, poly-
                                                                    mor-
                                                                    phic_ctype,
                                                                    campaign,
                                                                    basketline-
                                                                    effect_ptr,
                                                                    per_line_discount,
                                                                    dis-
                                                                    count_amount)

Bases: shoop.campaigns.models.basket_line_effects.BasketLineEffect

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions.
We monkey patch them until a patch can be added to Django (which would probably be very small and make all
of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: -
ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at
hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our
own accessors that use our appropriate base_objects manager.

identifier = 'discount_from_product_line_effect'

model
    alias of Product

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

products

description

exception DoesNotExist
    Bases: shoop.campaigns.models.basket_line_effects.DoesNotExist
```


exception DiscountFromProduct.**MultipleObjectsReturned**

Bases: `shoop.campaigns.models.basket_line_effects.MultipleObjectsReturned`

DiscountFromProduct.**base_objects** = <django.db.models.manager.Manager object>

DiscountFromProduct.**basketlineeffect_ptr**

DiscountFromProduct.**get_discount_lines** (*order_source, original_lines*)

DiscountFromProduct.**objects** = <polymorphic.managers.PolymorphicManager object>

DiscountFromProduct.**polymorphic_primary_key_name** = 'id'

DiscountFromProduct.**polymorphic_super_sub_accessors_replaced** = False

shoop.campaigns.models.campaigns module

class shoop.campaigns.models.campaigns.**CampaignType**

Bases: `enumfields.enums.Enum`

An enumeration.

CATALOG = <CampaignType.CATALOG: 1>

BASKET = <CampaignType.BASKET: 2>

class shoop.campaigns.models.campaigns.**Campaign** (**args, **kwargs*)

Bases: `shoop.utils.properties.MoneyPropped`, `parler.models.TranslatableModel`

admin_url_suffix = None

shop

created_by

modified_by

class Meta

Bases: `object`

abstract = False

verbose_name = <django.utils.functional.lazy.<locals>.__proxy__ object>

verbose_name_plural = <django.utils.functional.lazy.<locals>.__proxy__ object>

Campaign.**is_available**()

Campaign.**type**

Campaign.**get_next_by_created_on** (**moreargs, **morekwargs*)

Campaign.**get_next_by_modified_on** (**moreargs, **morekwargs*)

Campaign.**get_previous_by_created_on** (**moreargs, **morekwargs*)

Campaign.**get_previous_by_modified_on** (**moreargs, **morekwargs*)

Campaign.**objects**

class shoop.campaigns.models.campaigns.**CatalogCampaign** (*id, shop, name, identifier, active, start_datetime, end_datetime, created_by, modified_by, created_on, modified_on*)

Bases: `shoop.campaigns.models.campaigns.Campaign`

admin_url_suffix = 'catalog_campaigns'

```
conditions
filters
translations
rules_match (context, shop_product)
classmethod get_matching (context, product)
exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist
exception CatalogCampaign.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned
CatalogCampaign.created_by
CatalogCampaign.effects
CatalogCampaign.get_next_by_created_on (*moreargs, **morekwargs)
CatalogCampaign.get_next_by_modified_on (*moreargs, **morekwargs)
CatalogCampaign.get_previous_by_created_on (*moreargs, **morekwargs)
CatalogCampaign.get_previous_by_modified_on (*moreargs, **morekwargs)
CatalogCampaign.modified_by
CatalogCampaign.objects = <parler.managers.TranslatableManager object>
CatalogCampaign.public_name
    Descriptor for translated attributes.

    This attribute proxies all get/set calls to the translated model.
CatalogCampaign.shop
class shoop.campaigns.models.campaigns.BasketCampaign (id, shop, name, identifier, active,
start_datetime, end_datetime,
created_by, modified_by, created_on, modified_on, basket_line_text, coupon)
    Bases: shoop.campaigns.models.campaigns.Campaign
admin_url_suffix = 'basket_campaigns'
conditions
coupon
translations
classmethod get_matching (basket, lines)
rules_match (basket, lines)
    Check if basket rules match.

    They will not match if 1) The campaign is not active 2) The campaign has attached coupon
    Unexpected indentation.

        which doesn't match or is not active

    Block quote ends without a blank line; unexpected unindent.
```

3. Any of the attached rules doesn't match

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception BasketCampaign.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`BasketCampaign.created_by`

`BasketCampaign.discount_effects`

`BasketCampaign.get_next_by_created_on(*moreargs, **morekwargs)`

`BasketCampaign.get_next_by_modified_on(*moreargs, **morekwargs)`

`BasketCampaign.get_previous_by_created_on(*moreargs, **morekwargs)`

`BasketCampaign.get_previous_by_modified_on(*moreargs, **morekwargs)`

`BasketCampaign.line_effects`

`BasketCampaign.modified_by`

`BasketCampaign.objects = <parler.managers.TranslatableManager object>`

`BasketCampaign.public_name`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`BasketCampaign.shop`

class `shoop.campaigns.models.campaigns.CouponUsage` (*id, coupon, order, created_by, modified_by, created_on, modified_on*)

Bases: `django.db.models.base.Model`

`coupon`

`order`

`created_by`

`modified_by`

`classmethod add_usage` (*order, coupon*)

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception CouponUsage.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`CouponUsage.get_next_by_created_on(*moreargs, **morekwargs)`

`CouponUsage.get_next_by_modified_on(*moreargs, **morekwargs)`

`CouponUsage.get_previous_by_created_on(*moreargs, **morekwargs)`

`CouponUsage.get_previous_by_modified_on(*moreargs, **morekwargs)`

`CouponUsage.objects = <django.db.models.manager.Manager object>`

class `shoop.campaigns.models.campaigns.Coupon` (*id, code, usage_limit_customer, usage_limit, active, created_by, modified_by, created_on, modified_on*)

Bases: `django.db.models.base.Model`

```
admin_url_suffix = 'coupons'

created_by

modified_by

classmethod generate_code (length=6)

exhausted

attached

attach_to_campaign (campaign)

classmethod is_usable (code, customer)

can_use_code (customer)
    Check if customer can use the code

    Parameters customer (Contact or None) –
    Return type True|False

use (order)

increase_customer_usage_limit_by (amount)

increase_usage_limit_by (amount)

has_been_used (usage_count=1)
    See if code is used the times given

save (**kwargs)

exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception Coupon.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

Coupon.campaign

Coupon.get_next_by_created_on (*moreargs, **morekwargs)

Coupon.get_next_by_modified_on (*moreargs, **morekwargs)

Coupon.get_previous_by_created_on (*moreargs, **morekwargs)

Coupon.get_previous_by_modified_on (*moreargs, **morekwargs)

Coupon.objects = <django.db.models.manager.Manager object>

Coupon.usages

class shoop.campaigns.models.campaigns.BasketCampaignTranslation (id, language_code,
                                                                    public_name,
                                                                    master)

Bases: parler.models.TranslatedFieldsModel

exception DoesNotExist
    Bases: parler.models.TranslationDoesNotExist, shoop.campaigns.models.campaigns.DoesNotExist
    shoop.campaigns.models.campaigns.DoesNotExist

exception BasketCampaignTranslation.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned
```

```

BasketCampaignTranslation.get_language_code_display(*moreargs,      **morek-
                                                    wargs)

BasketCampaignTranslation.master

BasketCampaignTranslation.objects = <django.db.models.manager.Manager object>

class shoop.campaigns.models.campaigns.CatalogCampaignTranslation(id,      lan-
                                                                    guage_code,
                                                                    public_name,
                                                                    master)

Bases: parler.models.TranslatedFieldsModel

exception DoesNotExist
    Bases: parler.models.TranslationDoesNotExist, shoop.campaigns.models.campaigns.DoesNotExist
    shoop.campaigns.models.campaigns.DoesNotExist

exception CatalogCampaignTranslation.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

CatalogCampaignTranslation.get_language_code_display(*moreargs,      **morek-
                                                    wargs)

CatalogCampaignTranslation.master

CatalogCampaignTranslation.objects = <django.db.models.manager.Manager object>

```

shoop.campaigns.models.catalog_filters module

```

class shoop.campaigns.models.catalog_filters.CatalogFilter(id,      polymorphic_ctype,
                                                            active)

Bases: polymorphic.models.PolymorphicModel

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions.
We monkey patch them until a patch can be added to Django (which would probably be very small and make all
of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: -
ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at
hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our
own accessors that use our appropriate base_objects manager.

model = None

identifier = 'base_catalog_filter'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

filter_queryset(queryset)

exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception CatalogFilter.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

CatalogFilter.base_objects = <django.db.models.manager.Manager object>

CatalogFilter.campaign

CatalogFilter.categoryfilter

```

```
CatalogFilter.objects = <polymorphic.managers.PolymorphicManager object>
CatalogFilter.polymorphic_ctype
CatalogFilter.polymorphic_primary_key_name = 'id'
CatalogFilter.polymorphic_super_sub_accessors_replaced = False
CatalogFilter.productfilter
CatalogFilter.producttypefilter
class shoop.campaigns.models.catalog_filters.ProductTypeFilter(id,      polymor-
                                                                phic_ctype, active,
                                                                catalogfilter_ptr)
Bases: shoop.campaigns.models.catalog_filters.CatalogFilter
```

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

```
model
    alias of ProductType
identifier = 'product_type_filter'
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
product_types
filter_queryset (queryset)
description
values
exception DoesNotExist
    Bases: shoop.campaigns.models.catalog_filters.DoesNotExist
exception ProductTypeFilter.MultipleObjectsReturned
    Bases: shoop.campaigns.models.catalog_filters.MultipleObjectsReturned
ProductTypeFilter.base_objects = <django.db.models.manager.Manager object>
ProductTypeFilter.catalogfilter_ptr
ProductTypeFilter.objects = <polymorphic.managers.PolymorphicManager object>
ProductTypeFilter.polymorphic_primary_key_name = 'id'
ProductTypeFilter.polymorphic_super_sub_accessors_replaced = False
class shoop.campaigns.models.catalog_filters.ProductFilter(id,      polymorphic_ctype,
                                                            active, catalogfilter_ptr)
Bases: shoop.campaigns.models.catalog_filters.CatalogFilter
```

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

```

model
    alias of Product

identifier = 'product_filter'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

products

filter_queryset (queryset)

description

values

exception DoesNotExist
    Bases: shoop.campaigns.models.catalog_filters.DoesNotExist

exception ProductFilter.MultipleObjectsReturned
    Bases: shoop.campaigns.models.catalog_filters.MultipleObjectsReturned

ProductFilter.base_objects = <django.db.models.manager.Manager object>

ProductFilter.catalogfilter_ptr

ProductFilter.objects = <polymorphic.managers.PolymorphicManager object>

ProductFilter.polymorphic_primary_key_name = 'id'

ProductFilter.polymorphic_super_sub_accessors_replaced = False

```

```

class shoop.campaigns.models.catalog_filters.CategoryFilter (id, polymorphic_ctype,
                                                         active, catalogfil-
                                                         ter_ptr)

    Bases: shoop.campaigns.models.catalog_filters.CategoryFilter

```

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

```

model
    alias of Category

identifier = 'category_filter'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

categories

```

filter_queryset (*queryset*)

exception DoesNotExist

Bases: `shoop.campaigns.models.catalog_filters.DoesNotExist`

exception CategoryFilter.MultipleObjectsReturned

Bases: `shoop.campaigns.models.catalog_filters.MultipleObjectsReturned`

`CategoryFilter.base_objects` = `<django.db.models.manager.Manager object>`

`CategoryFilter.catalogfilter_ptr`

`CategoryFilter.description`

`CategoryFilter.objects` = `<polymorphic.managers.PolymorphicManager object>`

`CategoryFilter.polymorphic_primary_key_name` = `'id'`

`CategoryFilter.polymorphic_super_sub_accessors_replaced` = `False`

`CategoryFilter.values`

shoop.campaigns.models.contact_group_sales_ranges module

class `shoop.campaigns.models.contact_group_sales_ranges.SalesRangeQuerySet` (*model=None,*
query=None,
us-
ing=None,
hints=None)

Bases: `django.db.models.query.QuerySet`

active (*shop*)

class `shoop.campaigns.models.contact_group_sales_ranges.ContactGroupSalesRange` (*id,*
group,
shop,
min_value,
max_value)

Bases: `django.db.models.base.Model`

group

shop

objects = `<django.db.models.manager.ManagerFromSalesRangeQuerySet object>`

save (**args, **kwargs*)

clean ()

is_active ()

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception ContactGroupSalesRange.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

shoop.campaigns.models.context_conditions module

class `shoop.campaigns.models.context_conditions.ContextCondition` (*id,* *polymor-*
phic_ctype,
active)

Bases: `polymorphic.models.PolymorphicModel`

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

model = None

identifier = 'context_condition'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

description = <django.utils.functional.lazy.<locals>.__proxy__ object>

matches (context)

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception ContextCondition.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

ContextCondition.base_objects = <django.db.models.manager.Manager object>

ContextCondition.campaign

ContextCondition.contactcondition

ContextCondition.contactgroupcondition

ContextCondition.objects = <polymorphic.managers.PolymorphicManager object>

ContextCondition.polymorphic_ctype

ContextCondition.polymorphic_primary_key_name = 'id'

ContextCondition.polymorphic_super_sub_accessors_replaced = False

```
class shoop.campaigns.models.context_conditions.ContactGroupCondition(id,
                                                                    polymor-
                                                                    phic_ctype,
                                                                    active,
                                                                    con-
                                                                    textcon-
                                                                    di-
                                                                    tion_ptr)
    Bases: shoop.campaigns.models.context_conditions.ContextCondition
```

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate `base_objects` manager.

```
model
    alias of ContactGroup

identifier = 'contact_group_condition'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

contact_groups

matches (context)

description

values

exception DoesNotExist
    Bases: shoop.campaigns.models.context_conditions.DoesNotExist

exception ContactGroupCondition.MultipleObjectsReturned
    Bases: shoop.campaigns.models.context_conditions.MultipleObjectsReturned

ContactGroupCondition.base_objects = <django.db.models.manager.Manager object>

ContactGroupCondition.contextcondition_ptr

ContactGroupCondition.objects = <polymorphic.managers.PolymorphicManager object>

ContactGroupCondition.polymorphic_primary_key_name = 'id'

ContactGroupCondition.polymorphic_super_sub_accessors_replaced = False

class shoop.campaigns.models.context_conditions.ContactCondition (id, polymor-
                                                                    phic_ctype, ac-
                                                                    tive, contextcon-
                                                                    dition_ptr)

    Bases: shoop.campaigns.models.context_conditions.ContextCondition
```

Replace Django's inheritance accessor member functions for our model (`self.__class__`) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate `base_objects` manager.

```
model
    alias of Contact

identifier = 'contact_condition'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

contacts

matches (context)

description
```

exception DoesNotExist

Bases: `shoop.campaigns.models.context_conditions.DoesNotExist`

exception ContactCondition.MultipleObjectsReturned

Bases: `shoop.campaigns.models.context_conditions.MultipleObjectsReturned`

`ContactCondition.base_objects = <django.db.models.manager.Manager object>`

`ContactCondition.contextcondition_ptr`

`ContactCondition.objects = <polymorphic.managers.PolymorphicManager object>`

`ContactCondition.polymorphic_primary_key_name = 'id'`

`ContactCondition.polymorphic_super_sub_accessors_replaced = False`

`ContactCondition.values`

shoop.campaigns.models.product_effects module

class `shoop.campaigns.models.product_effects.ProductDiscountEffect` (*id, polymorphic_ctype, campaign*)

Bases: `shoop.core.models.PolymorphicShoopModel`

Replace Django's inheritance accessor member functions for our model (`self.__class__`) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: `modela_ptr`, `modela` - ModelB: `modela_ptr`, `modelb`, `modelc` - ModelC: `modela_ptr`, `modelb`, `modelb_ptr`, `modelc`

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate `base_objects` manager.

identifier = None

model = None

admin_form_class = None

campaign

apply_for_product (*context, product, price_info*)

Applies the effect for product

Returns amount of discount to accumulate for the product

Return type *Price*

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception ProductDiscountEffect.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`ProductDiscountEffect.base_objects = <django.db.models.manager.Manager object>`

`ProductDiscountEffect.objects = <polymorphic.managers.PolymorphicManager object>`

`ProductDiscountEffect.polymorphic_ctype`

`ProductDiscountEffect.polymorphic_primary_key_name = 'id'`

```
ProductDiscountEffect.polymorphic_super_sub_accessors_replaced = False
ProductDiscountEffect.productdiscountamount
ProductDiscountEffect.productdiscountpercentage
class shoop.campaigns.models.product_effects.ProductDiscountAmount (id, polymor-
                                                                    phic_ctype,
                                                                    campaign,
                                                                    product-
                                                                    discountef-
                                                                    fect_ptr, dis-
                                                                    count_amount)

Bases: shoop.campaigns.models.product_effects.ProductDiscountEffect
```

Replace Django's inheritance accessor member functions for our model (`self.__class__`) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

```
identifier = 'discount_amount_effect'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

description

value

apply_for_product (context, product, price_info)

exception DoesNotExist
    Bases: shoop.campaigns.models.product_effects.DoesNotExist

exception ProductDiscountAmount.MultipleObjectsReturned
    Bases: shoop.campaigns.models.product_effects.MultipleObjectsReturned

ProductDiscountAmount.base_objects = <django.db.models.manager.Manager object>
ProductDiscountAmount.objects = <polymorphic.managers.PolymorphicManager object>
ProductDiscountAmount.polymorphic_primary_key_name = 'id'
ProductDiscountAmount.polymorphic_super_sub_accessors_replaced = False
ProductDiscountAmount.productdiscounteffect_ptr
```

```
class shoop.campaigns.models.product_effects.ProductDiscountPercentage (id,
                                                                    poly-
                                                                    mor-
                                                                    phic_ctype,
                                                                    cam-
                                                                    paign,
                                                                    pro-
                                                                    ductdis-
                                                                    coun-
                                                                    tef-
                                                                    fect_ptr,
                                                                    dis-
                                                                    count_percentage)
```

Bases: *shoop.campaigns.models.product_effects.ProductDiscountEffect*

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

identifier = 'discount_percentage_effect'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

admin_form_class

alias of PercentageField

description

exception DoesNotExist

Bases: shoop.campaigns.models.product_effects.DoesNotExist

exception ProductDiscountPercentage.MultipleObjectsReturned

Bases: shoop.campaigns.models.product_effects.MultipleObjectsReturned

ProductDiscountPercentage.**base_objects** = <django.db.models.manager.Manager object>

ProductDiscountPercentage.**objects** = <polymorphic.managers.PolymorphicManager object>

ProductDiscountPercentage.**polymorphic_primary_key_name** = 'id'

ProductDiscountPercentage.**polymorphic_super_sub_accessors_replaced** = False

ProductDiscountPercentage.**productdiscounteffect_ptr**

ProductDiscountPercentage.**value**

ProductDiscountPercentage.**apply_for_product** (context, product, price_info)

Module contents

```
class shoop.campaigns.models.BasketLineEffect (id, polymorphic_ctype, campaign)
```

Bases: *shoop.core.models.PolymorphicShoopModel*

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception BasketLineEffect.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`BasketLineEffect.admin_form_class = None`

`BasketLineEffect.base_objects = <django.db.models.manager.Manager object>`

`BasketLineEffect.campaign`

`BasketLineEffect.discountfromproduct`

`BasketLineEffect.freeproductline`

`BasketLineEffect.get_discount_lines (order_source, original_lines)`

Applies the effect based on given order_source

Returns amount of discount to accumulate for the product

Return type `Iterable[shoop.core.order_creator.SourceLine]`

`BasketLineEffect.identifier = None`

`BasketLineEffect.model = None`

`BasketLineEffect.objects = <polymorphic.managers.PolymorphicManager object>`

`BasketLineEffect.polymorphic_ctype`

`BasketLineEffect.polymorphic_primary_key_name = 'id'`

`BasketLineEffect.polymorphic_super_sub_accessors_replaced = False`

`class shoop.campaigns.models.BasketCampaign (id, shop, name, identifier, active, start_datetime, end_datetime, created_by, modified_by, created_on, modified_on, basket_line_text, coupon)`

Bases: `shoop.campaigns.models.campaigns.Campaign`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception BasketCampaign.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`BasketCampaign.admin_url_suffix = 'basket_campaigns'`

`BasketCampaign.conditions`

`BasketCampaign.coupon`

`BasketCampaign.created_by`

BasketCampaign.**discount_effects**

classmethod BasketCampaign.**get_matching** (*basket, lines*)

BasketCampaign.**get_next_by_created_on** (**moreargs, **morekwargs*)

BasketCampaign.**get_next_by_modified_on** (**moreargs, **morekwargs*)

BasketCampaign.**get_previous_by_created_on** (**moreargs, **morekwargs*)

BasketCampaign.**get_previous_by_modified_on** (**moreargs, **morekwargs*)

BasketCampaign.**line_effects**

BasketCampaign.**modified_by**

BasketCampaign.**objects** = <parler.managers.TranslatableManager object>

BasketCampaign.**public_name**

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

BasketCampaign.**rules_match** (*basket, lines*)

Check if basket rules match.

They will not match if 1) The campaign is not active 2) The campaign has attached coupon

Unexpected indentation.

which doesn't match or is not active

Block quote ends without a blank line; unexpected unindent.

3.Any of the attached rules doesn't match

BasketCampaign.**shop**

BasketCampaign.**translations**

class shoop.campaigns.models.**BasketDiscountEffect** (*id, polymorphic_ctype, campaign*)

Bases: [shoop.core.models.PolymorphicShoopModel](#)

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB ->ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

exception DoesNotExist

Bases: [django.core.exceptions.ObjectDoesNotExist](#)

exception BasketDiscountEffect.**MultipleObjectsReturned**

Bases: [django.core.exceptions.MultipleObjectsReturned](#)

BasketDiscountEffect.**admin_form_class** = None

`BasketDiscountEffect.apply_for_basket(order_source)`

Applies the effect based on given `order_source`

Returns amount of discount to accumulate for the product

Return type *Price*

`BasketDiscountEffect.base_objects = <django.db.models.manager.Manager object>`

`BasketDiscountEffect.basketdiscountamount`

`BasketDiscountEffect.basketdiscountpercentage`

`BasketDiscountEffect.campaign`

`BasketDiscountEffect.identifier = None`

`BasketDiscountEffect.model = None`

`BasketDiscountEffect.objects = <polymorphic.managers.PolymorphicManager object>`

`BasketDiscountEffect.polymorphic_ctype`

`BasketDiscountEffect.polymorphic_primary_key_name = 'id'`

`BasketDiscountEffect.polymorphic_super_sub_accessors_replaced = False`

class `shoop.campaigns.models.BasketCondition(id, polymorphic_ctype, active)`

Bases: `polymorphic.models.PolymorphicModel`

Replace Django's inheritance accessor member functions for our model (`self.__class__`) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: `modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc`

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate `base_objects` manager.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `BasketCondition.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`BasketCondition.base_objects = <django.db.models.manager.Manager object>`

`BasketCondition.basketmaxtotalamountcondition`

`BasketCondition.basketmaxtotalproductamountcondition`

`BasketCondition.basketttotalamountcondition`

`BasketCondition.basketttotalproductamountcondition`

`BasketCondition.campaign`

`BasketCondition.contactbasketcondition`

`BasketCondition.contactgroupbasketcondition`

`BasketCondition.matches(basket, lines)`

`BasketCondition.model = None`


```

BasketCondition.name = <django.utils.functional.lazy.<locals>.__proxy__ object>
BasketCondition.objects = <polymorphic.managers.PolymorphicManager object>
BasketCondition.polymorphic_ctype
BasketCondition.polymorphic_primary_key_name = 'id'
BasketCondition.polymorphic_super_sub_accessors_replaced = False
BasketCondition.productsinbasketcondition

class shoop.campaigns.models.Campaign(*args, **kwargs)
    Bases: shoop.utils.properties.MoneyPropped, parler.models.TranslatableModel

    class Meta
        Bases: object
        abstract = False
        verbose_name = <django.utils.functional.lazy.<locals>.__proxy__ object>
        verbose_name_plural = <django.utils.functional.lazy.<locals>.__proxy__ object>

    Campaign.admin_url_suffix = None
    Campaign.created_by
    Campaign.get_next_by_created_on(*moreargs, **morekwargs)
    Campaign.get_next_by_modified_on(*moreargs, **morekwargs)
    Campaign.get_previous_by_created_on(*moreargs, **morekwargs)
    Campaign.get_previous_by_modified_on(*moreargs, **morekwargs)
    Campaign.is_available()
    Campaign.modified_by
    Campaign.objects
    Campaign.shop
    Campaign.type

class shoop.campaigns.models.ProductDiscountEffect(id, polymorphic_ctype, campaign)
    Bases: shoop.core.models.PolymorphicShoopModel

    Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions.
    We monkey patch them until a patch can be added to Django (which would probably be very small and make all
    of this obsolete).

    If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: -
    ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

    These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at
    hand.

    The original Django accessors use our polymorphic manager. But they should not. So we replace them with our
    own accessors that use our appropriate base_objects manager.

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception ProductDiscountEffect.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

```

```
ProductDiscountEffect.admin_form_class = None
ProductDiscountEffect.apply_for_product(context, product, price_info)
    Applies the effect for product

    Returns amount of discount to accumulate for the product

    Return type Price

ProductDiscountEffect.base_objects = <django.db.models.manager.Manager object>
ProductDiscountEffect.campaign
ProductDiscountEffect.identifier = None
ProductDiscountEffect.model = None
ProductDiscountEffect.objects = <polymorphic.managers.PolymorphicManager object>
ProductDiscountEffect.polymorphic_ctype
ProductDiscountEffect.polymorphic_primary_key_name = 'id'
ProductDiscountEffect.polymorphic_super_sub_accessors_replaced = False
ProductDiscountEffect.productdiscountamount
ProductDiscountEffect.productdiscountpercentage

class shoop.campaigns.models.CatalogCampaign(id, shop, name, identifier, active,
                                             start_datetime, end_datetime, created_by,
                                             modified_by, created_on, modified_on)
    Bases: shoop.campaigns.models.campaigns.Campaign

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception CatalogCampaign.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

CatalogCampaign.admin_url_suffix = 'catalog_campaigns'
CatalogCampaign.conditions
CatalogCampaign.created_by
CatalogCampaign.effects
CatalogCampaign.filters
classmethod CatalogCampaign.get_matching(context, product)
CatalogCampaign.get_next_by_created_on(*moreargs, **morekwargs)
CatalogCampaign.get_next_by_modified_on(*moreargs, **morekwargs)
CatalogCampaign.get_previous_by_created_on(*moreargs, **morekwargs)
CatalogCampaign.get_previous_by_modified_on(*moreargs, **morekwargs)
CatalogCampaign.modified_by
CatalogCampaign.objects = <parler.managers.TranslatableManager object>
CatalogCampaign.public_name
    Descriptor for translated attributes.

    This attribute proxies all get/set calls to the translated model.

CatalogCampaign.rules_match(context, shop_product)
```

CatalogCampaign.**shop**

CatalogCampaign.**translations**

class shoop.campaigns.models.**CatalogFilter** (*id, polymorphic_ctype, active*)

Bases: `polymorphic.models.PolymorphicModel`

Replace Django's inheritance accessor member functions for our model (`self.__class__`) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB ->ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception CatalogFilter.**MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

CatalogFilter.**base_objects** = <django.db.models.manager.Manager object>

CatalogFilter.**campaign**

CatalogFilter.**categoryfilter**

CatalogFilter.**filter_queryset** (*queryset*)

CatalogFilter.**identifier** = 'base_catalog_filter'

CatalogFilter.**model** = None

CatalogFilter.**name** = <django.utils.functional.lazy.<locals>.__proxy__ object>

CatalogFilter.**objects** = <polymorphic.managers.PolymorphicManager object>

CatalogFilter.**polymorphic_ctype**

CatalogFilter.**polymorphic_primary_key_name** = 'id'

CatalogFilter.**polymorphic_super_sub_accessors_replaced** = False

CatalogFilter.**productfilter**

CatalogFilter.**producttypefilter**

class shoop.campaigns.models.**ContactGroupSalesRange** (*id, group, shop, min_value, max_value*)

Bases: `django.db.models.base.Model`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception ContactGroupSalesRange.**MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

ContactGroupSalesRange.**clean** ()

ContactGroupSalesRange.**group**

ContactGroupSalesRange.**is_active** ()

```
ContactGroupSalesRange.objects = <django.db.models.manager.ManagerFromSalesRangeQuerySet object>
ContactGroupSalesRange.save(*args, **kwargs)
ContactGroupSalesRange.shop
```

```
class shoop.campaigns.models.ContextCondition(id, polymorphic_ctype, active)
    Bases: polymorphic.models.PolymorphicModel
```

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception ContextCondition.MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

```
ContextCondition.base_objects = <django.db.models.manager.Manager object>
```

```
ContextCondition.campaign
```

```
ContextCondition.contactcondition
```

```
ContextCondition.contactgroupcondition
```

```
ContextCondition.description = <django.utils.functional.lazy.<locals>.__proxy__ object>
```

```
ContextCondition.identifier = 'context_condition'
```

```
ContextCondition.matches(context)
```

```
ContextCondition.model = None
```

```
ContextCondition.name = <django.utils.functional.lazy.<locals>.__proxy__ object>
```

```
ContextCondition.objects = <polymorphic.managers.PolymorphicManager object>
```

```
ContextCondition.polymorphic_ctype
```

```
ContextCondition.polymorphic_primary_key_name = 'id'
```

```
ContextCondition.polymorphic_super_sub_accessors_replaced = False
```

```
class shoop.campaigns.models.Coupon(id, code, usage_limit_customer, usage_limit, active, created_by, modified_by, created_on, modified_on)
```

Bases: django.db.models.base.Model

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception Coupon.MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

```
Coupon.admin_url_suffix = 'coupons'
```

```
Coupon.attach_to_campaign(campaign)
```

`Coupon.attached`

`Coupon.campaign`

`Coupon.can_use_code` (*customer*)
Check if customer can use the code

Parameters `customer` (Contact or None) –

Return type True|False

`Coupon.created_by`

`Coupon.exhausted`

classmethod `Coupon.generate_code` (*length=6*)

`Coupon.get_next_by_created_on` (**moreargs, **morekwargs*)

`Coupon.get_next_by_modified_on` (**moreargs, **morekwargs*)

`Coupon.get_previous_by_created_on` (**moreargs, **morekwargs*)

`Coupon.get_previous_by_modified_on` (**moreargs, **morekwargs*)

`Coupon.has_been_used` (*usage_count=1*)
See if code is used the times given

`Coupon.increase_customer_usage_limit_by` (*amount*)

`Coupon.increase_usage_limit_by` (*amount*)

classmethod `Coupon.is_usable` (*code, customer*)

`Coupon.modified_by`

`Coupon.objects` = <django.db.models.manager.Manager object>

`Coupon.save` (***kwargs*)

`Coupon.usages`

`Coupon.use` (*order*)

shoop.campaigns.templates package

Module contents

Submodules

shoop.campaigns.apps module

class `shoop.campaigns.apps.CampaignAppConfig` (**args, **kwargs*)

Bases: `shoop.apps.AppConfig`

name = 'shoop.campaigns'

verbose_name = 'Shoop Campaigns'

label = 'campaigns'

provides = {'admin_contact_group_form_part': ['shoop.campaigns.admin_module.form_parts:SalesRangesFormPart']

ready ()

shoop.campaigns.modules module

class shoop.campaigns.modules.**CatalogCampaignModule**

Bases: *shoop.core.pricing.DiscountModule*

identifier = 'catalog_campaigns'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

discount_price (*context, product, price_info*)

Get the discounted price for context.

Best discount is selected. Minimum price will be selected if the cheapest price is under that.

class shoop.campaigns.modules.**BasketCampaignModule**

Bases: *shoop.core.order_creator.OrderSourceModifierModule*

identifier = 'basket_campaigns'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

get_new_lines (*order_source, lines*)

can_use_code (*order_source, code*)

use_code (*order, code*)

clear_codes (*order*)

shoop.campaigns.signal_handlers module

shoop.campaigns.signal_handlers.**update_customers_groups** (*sender,* *instance,*
***kwargs*)

shoop.campaigns.utils module

shoop.campaigns.utils.**get_total_sales** (*shop, customer*)

shoop.campaigns.utils.**assign_to_group_based_on_sales** (*cls,* *shop,* *customer,*
sales_range=None)

shoop.campaigns.utils.**get_contacts_in_sales_range** (*shop, min_value, max_value*)

Module contents

shoop.core package

Subpackages

shoop.core.api package

Submodules

shoop.core.api.orders module

class shoop.core.api.orders.**OrderLineSerializer** (*instance=None,* *data=<class*
'rest_framework.fields.empty'>,
***kwargs*)

Bases: *rest_framework.serializers.ModelSerializer*

```

class Meta
    Bases: object

    model
        alias of OrderLine
class shoop.core.api.orders.AddressSerializer (instance=None, data=<class
        'rest_framework.fields.empty'>, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

class Meta
    Bases: object

    model
        alias of MutableAddress
class shoop.core.api.orders.PaymentSerializer (instance=None, data=<class
        'rest_framework.fields.empty'>, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

class Meta
    Bases: object

    model
        alias of Payment

    fields = ('payment_identifier', 'amount_value', 'description')
class shoop.core.api.orders.OrderSerializer (instance=None, data=<class
        'rest_framework.fields.empty'>, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    get_fields ()

    create (validated_data)

class Meta
    Bases: object

    model
        alias of Order
class shoop.core.api.orders.OrderViewSet (**kwargs)
    Bases: rest_framework.viewsets.ModelViewSet

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    serializer_class
        alias of OrderSerializer

    queryset

    create_payment (request, pk=None)

    set_fully_paid (request, pk=None)

    suffix = None

```

shoop.core.api.products module

```

class shoop.core.api.products.ShopProductSerializer (instance=None, data=<class
        'rest_framework.fields.empty'>, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

```

```
class Meta
    Bases: object

    model
        alias of ShopProduct

    extra_kwargs = {'categories': {'required': False}, 'shipping_methods': {'required': False}, 'suppliers': {'required': False}}

class shoop.core.api.products.ProductSerializer(instance=None, data=<class 'rest_framework.fields.empty'>, **kwargs)
    Bases: parler_rest.serializers.TranslatableModelSerializer

class Meta
    Bases: object

    model
        alias of Product

class shoop.core.api.products.ProductViewSet(**kwargs)
    Bases: rest_framework.viewsets.ModelViewSet

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    queryset = []

    serializer_class
        alias of ProductSerializer

    get_queryset()

    suffix = None

class shoop.core.api.products.ShopProductViewSet(**kwargs)
    Bases: rest_framework.viewsets.ModelViewSet

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    queryset = []

    serializer_class
        alias of ShopProductSerializer

    get_queryset()

    suffix = None
```

Module contents

`shoop.core.api.populate_core_api(router)`

Parameters **router** (`rest_framework.routers.DefaultRouter`) – Router

shoop.core.cache package

Submodules

shoop.core.cache.impl module

`shoop.core.cache.impl.get_cache_duration(cache_key)`

Determine a cache duration for the given cache key.

Parameters **cache_key** (*str*) – Cache key string

Returns Timeout seconds

Return type `int`

class `shoop.core.cache.impl.VersionedCache` (*using*)

Bases: `object`

Parameters `using` (*str*) – Cache alias

bump_version (*cache_key*)

Bump up the cache version for the given cache key/namespace.

Parameters `cache_key` (*str*) – Cache key or namespace

get_version (*cache_key*)

Get the cache version (or None) for the given cache key/namespace.

The cache version is stored in thread-local storage for the current request, so unless bumped in-request, all gets within a single request should get coherently versioned data from the cache.

Parameters `cache_key` (*str*) – Cache key or namespace

Returns Version ID or none

Return type `str|None`

set (*key, value, timeout=None, version=None*)

Set the value for key *key* in the cache.

Unlike `django.core.caches[using].set()`, this also derives timeout and versioning information from the key (and cached version data) if the key begins with a colon-separated namespace, such as `foo:bar`.

Parameters

- **key** (*str*) – Cache key
- **value** (`object`) – Value to cache
- **timeout** (*int|None*) – Timeout seconds or None (for auto-determination)
- **version** (*str|None*) – Version string or None (for auto-determination)
- **using** (*str*) – Cache alias

get (*key, version=None, default=None*)

Get the value for key *key* in the cache.

Unlike `django.core.caches[using].get()`, versioning information can be auto-derived from the key (and cached version data) if the key begins with a colon-separated namespace, such as `foo:bar`.

Parameters

- **key** (*str*) – Cache key
- **version** (*str|None*) – Version string or None (for auto-determination)
- **default** (`object`) – Default value, if the key is not found
- **using** (*str*) – Cache alias

Returns cached value

Return type `object`

clear ()

Module contents Utilities for versioned caching and automatic timeout determination.

Versioning works by way of namespaces. Namespaces are the first colon-separated part of cache keys.

For instance, the cache keys `price:10`, `price:20`, and `price` all belong to the `price` namespace and can be invalidated with one `bump_version("price")` call.

The versions themselves are stored within the cache, within the `_version` namespace. (As an implementation detail, this allows one to invalidate `_all_` versioned keys by bumping the version of `_version`. Very meta!)

class `shoop.core.cache.VersionedCache` (*using*)

Bases: `object`

Parameters `using` (*str*) – Cache alias

bump_version (*cache_key*)

Bump up the cache version for the given cache key/namespace.

Parameters `cache_key` (*str*) – Cache key or namespace

clear ()

get (*key*, *version=None*, *default=None*)

Get the value for key `key` in the cache.

Unlike `django.core.caches[using].get()`, versioning information can be auto-derived from the key (and cached version data) if the key begins with a colon-separated namespace, such as `foo:bar`.

Parameters

- **key** (*str*) – Cache key
- **version** (*str/None*) – Version string or None (for auto-determination)
- **default** (*object*) – Default value, if the key is not found
- **using** (*str*) – Cache alias

Returns cached value

Return type *object*

get_version (*cache_key*)

Get the cache version (or None) for the given cache key/namespace.

The cache version is stored in thread-local storage for the current request, so unless bumped in-request, all gets within a single request should get coherently versioned data from the cache.

Parameters `cache_key` (*str*) – Cache key or namespace

Returns Version ID or none

Return type `str/None`

set (*key*, *value*, *timeout=None*, *version=None*)

Set the value for key `key` in the cache.

Unlike `django.core.caches[using].set()`, this also derives timeout and versioning information from the key (and cached version data) if the key begins with a colon-separated namespace, such as `foo:bar`.

Parameters

- **key** (*str*) – Cache key
- **value** (*object*) – Value to cache
- **timeout** (*int/None*) – Timeout seconds or None (for auto-determination)

- **version** (*str* / *None*) – Version string or None (for auto-determination)
- **using** (*str*) – Cache alias

shoop.core.defaults package

Submodules

shoop.core.defaults.order_statuses module

shoop.core.defaults.order_statuses.**create_default_order_statuses** ()

Module contents

shoop.core.fields package

Submodules

shoop.core.fields.tagged_json module “Tagged JSON” encoder/decoder.

Objects that are normally not unambiguously representable via JSON are encoded into special objects of the form {tag: val}; the encoding and decoding process can be customized however necessary.

shoop.core.fields.tagged_json.**isoformat** (*obj*)

shoop.core.fields.tagged_json.**encode_enum** (*enum_val*)

shoop.core.fields.tagged_json.**decode_enum** (*val*)

class shoop.core.fields.tagged_json.**TagRegistry**

Bases: *object*

register (*tag, classes, encoder=<class 'str'>, decoder=None*)

encode (*obj, default*)

decode (*obj*)

shoop.core.fields.tagged_json.**tag_registry** = <shoop.core.fields.tagged_json.TagRegistry object>

The default tag registry.

class shoop.core.fields.tagged_json.**TaggedJSONEncoder** (*skipkeys=False, ensure_ascii=True, check_circular=True, allow_nan=True, sort_keys=False, indent=None, separators=None, default=None*)

Bases: *jsonfield.encoder.JSONEncoder*

Constructor for JSONEncoder, with sensible defaults.

If skipkeys is false, then it is a *TypeError* to attempt encoding of keys that are not str, int, float or None. If skipkeys is True, such items are simply skipped.

If ensure_ascii is true, the output is guaranteed to be str objects with all incoming non-ASCII characters escaped. If ensure_ascii is false, the output can contain non-ASCII characters.

If `check_circular` is true, then lists, dicts, and custom encoded objects will be checked for circular references during encoding to prevent an infinite recursion (which would cause an `OverflowError`). Otherwise, no such check takes place.

If `allow_nan` is true, then NaN, Infinity, and -Infinity will be encoded as such. This behavior is not JSON specification compliant, but is consistent with most JavaScript based encoders and decoders. Otherwise, it will be a `ValueError` to encode such floats.

If `sort_keys` is true, then the output of dictionaries will be sorted by key; this is useful for regression tests to ensure that JSON serializations can be compared on a day-to-day basis.

If `indent` is a non-negative integer, then JSON array elements and object members will be pretty-printed with that indent level. An indent level of 0 will only insert newlines. None is the most compact representation.

If specified, separators should be an (item_separator, key_separator) tuple. The default is (', ', ': ') if `indent` is None and (',', ': ') otherwise. To get the most compact JSON representation, you should specify (',', ':') to eliminate whitespace.

If specified, `default` is a function that gets called for objects that can't otherwise be serialized. It should return a JSON encodable version of the object or raise a `TypeError`.

registry = <shoop.core.fields.tagged_json.TagRegistry object>

default (*obj*)

Module contents

class `shoop.core.fields.InternalIdentifierField` (***kwargs*)

Bases: `django.db.models.fields.CharField`

get_prep_value (*value*)

deconstruct ()

class `shoop.core.fields.CurrencyField` (***kwargs*)

Bases: `django.db.models.fields.CharField`

class `shoop.core.fields.FormattedDecimalFormField` (*max_value=None, min_value=None, max_digits=None, decimal_places=None, *args, **kwargs*)

Bases: `django.forms.fields.DecimalField`

MAX_DECIMAL_PLACES_FOR_STEP = 5

widget_attrs (*widget*)

class `shoop.core.fields.FormattedDecimalField` (*verbose_name=None, name=None, max_digits=None, decimal_places=None, **kwargs*)

Bases: `django.db.models.fields.DecimalField`

`DecimalField` subclass to display decimal values in non-scientific format.

value_from_object (*obj*)

format_decimal (*value, max_digits=100, exponent_limit=100*)

formfield (***kwargs*)

class `shoop.core.fields.MoneyValueField` (***kwargs*)

Bases: `shoop.core.fields.FormattedDecimalField`

class `shoop.core.fields.QuantityField` (***kwargs*)

Bases: `shoop.core.fields.FormattedDecimalField`

```

class shoop.core.fields.MeasurementField(unit, **kwargs)
    Bases: shoop.core.fields.FormattedDecimalField

    KNOWN_UNITS = ('mm', 'm', 'kg', 'g', 'm3')

    deconstruct()

class shoop.core.fields.LanguageField(*args, **kwargs)
    Bases: django.db.models.fields.CharField

    LANGUAGE_CODES = {'fa_AF', 'nn', 'jrb', 'pt', 'wal', 'nv', 'sdh', 'ur', 'arp', 'war', 'bo', 'naq', 'asa', 'ak', 'vls', 'bas', 'rn'}

    get_choices(include_blank=True, blank_choice=[(' ', '—')])

class shoop.core.fields.UnsavedForeignKey(to, to_field=None, rel_class=<class
    'django.db.models.fields.related.ManyToOneRel'>,
    db_constraint=True, **kwargs)
    Bases: django.db.models.fields.related.ForeignKey

    allow_unsaved_instance_assignment = True

class shoop.core.fields.TaggedJSONField(*args, **kwargs)
    Bases: jsonfield.fields.JSONField

    contribute_to_class(cls, name)

```

shoop.core.management package

Subpackages

shoop.core.management.commands package

Submodules

shoop.core.management.commands.compute_bought_with_relations module

```

class shoop.core.management.commands.compute_bought_with_relations.Command(stdout=None,
                                                                              stderr=None,
                                                                              no_color=False)

    Bases: django.core.management.base.BaseCommand

    handle(*args, **options)

```

shoop.core.management.commands.makemessages module Patched version of Django's Makemessages that works with Jinja2.

Works by monkey patching django.utils.translation.trans_real.templatize with our version.

```

class shoop.core.management.commands.makemessages.Command(stdout=None,
                                                            stderr=None,
                                                            no_color=False)

    Bases: django.core.management.commands.makemessages.Command

    add_arguments(parser)

    handle(*args, **options)

```

build_potfiles()

Build PO Template files and return paths to them.

Extends base classes version of this method by adding the “Remove POT-Creation-Date” feature.

write_po_file(*potfile, locale*)

Writo PO file of given locale to disk.

Extends base classes version of this method by adding a feature to not change those PO files at all that have only the “POT-Creation-Date” header changed.

```
shoop.core.management.commands.makemessages.jinja_messages_to_python(src,  
                                                                    ori-  
                                                                    gin=None)
```

Convert Jinja2 file to Python preserving only messages.

```
shoop.core.management.commands.makemessages.extract_jinja(contents, origin=None)
```

shoop.core.management.commands.shoop_init module

```
shoop.core.management.commands.shoop_init.schema(model, identifier, **info)
```

```
class shoop.core.management.commands.shoop_init.Initializer
```

Bases: `object`

```
    schemata = [{‘identifier’: ‘default’, ‘info’: {‘status’: <ShopStatus.ENABLED: 1>, ‘name’: ‘Default Shop’, ‘model’: <cl
```

```
    process_schema(schema)
```

```
    run()
```

```
class shoop.core.management.commands.shoop_init.Command(stdout=None, stderr=None,  
                                                         no_color=False)
```

Bases: `django.core.management.base.BaseCommand`

```
    leave_locale_alone = True
```

```
    handle(*args, **options)
```

shoop.core.management.commands.shoop_makemessages module Makemessages helper for Shoop projects.

Runs Django’s makemessages for django and djangojs domains with sane defaults for Shoop projects (ignores and extensions).

```
class shoop.core.management.commands.shoop_makemessages.Command(stdout=None,  
                                                                stderr=None,  
                                                                no_color=False)
```

Bases: `shoop.core.management.commands.makemessages.Command`

```
    help = “\nMakemessages helper for Shoop projects.\n\nRuns Django’s makemessages for django and djangojs domains v
```

```
    add_arguments(parser)
```

```
    handle(*args, **options)
```

shoop.core.management.commands.shoop_show_settings module Show known Shoop settings and their values.

```
class shoop.core.management.commands.shoop_show_settings.Command(stdout=None,  
                                                                stderr=None,  
                                                                no_color=False)
```

Bases: `django.core.management.base.BaseCommand`

```
    help = ‘Show known Shoop settings and their values.’
```

```
option_list = (<Option at 0x7f713fc39f28: --only-changed>,)
handle (*args, **options)
```

Module contents

Module contents

shoop.core.models package

Module contents

```
class shoop.core.models.AnonymousContact (id, polymorphic_ctype, created_on, identifier, is_active, default_shipping_address, default_billing_address, default_shipping_method, default_payment_method, language, marketing_permission, phone, www, timezone, prefix, name, suffix, name_ext, email, tax_group, merchant_notes, account_manager, contact_ptr)
```

Bases: *shoop.core.models.Contact*

exception DoesNotExist

Bases: shoop.core.models._contacts.DoesNotExist

exception AnonymousContact.MultipleObjectsReturned

Bases: shoop.core.models._contacts.MultipleObjectsReturned

AnonymousContact.**base_objects** = <django.db.models.manager.Manager object>

AnonymousContact.**contact_ptr**

AnonymousContact.**default_contact_group_identifier** = 'default_anonymous_group'

AnonymousContact.**default_contact_group_name** = <django.utils.functional.lazy.<locals>.__proxy__ object>

AnonymousContact.**delete** (*args, **kwargs)

AnonymousContact.groups

Contact groups accessor for anonymous contact.

The base class already has a *groups* property via *ContactGroup* *related_name*, but this overrides it for *AnonymousContact* so that it will return a queryset containing just the anonymous contact group rather than returning the original related manager, which cannot work since *AnonymousContact* is not in the database.

This allows to use statements like this for all kinds of contacts, even *AnonymousContact*:

```
some_contact.groups.all()
```

Return type django.db.QuerySet

AnonymousContact.**id** = None

AnonymousContact.**is_anonymous** = True

AnonymousContact.**objects** = <polymorphic.managers.PolymorphicManager object>

AnonymousContact.**pk** = None

AnonymousContact.**polymorphic_primary_key_name** = 'id'

```
AnonymousContact.polymorphic_super_sub_accessors_replaced = False
AnonymousContact.save(*args, **kwargs)
class shoop.core.models.Attribute(id, identifier, searchable, type, visibility_mode)
    Bases: parler.models.TranslatableModel

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception Attribute.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    Attribute.formfield(**kwargs)
        Get a form field for this attribute.

        Parameters kwargs – Kwargs to pass for the form field class.

        Returns Form field.

        Return type forms.Field

    Attribute.get_type_display(*moreargs, **morekwargs)

    Attribute.get_visibility_mode_display(*moreargs, **morekwargs)

    Attribute.is_null_value(value)
        Find out whether the given value is null from this attribute's point of view.

        Parameters value (object) – A value

        Returns Nulliness boolean

        Return type bool

    Attribute.is_numeric

    Attribute.is_stringy

    Attribute.is_temporal

    Attribute.is_translated

    Attribute.name
        Descriptor for translated attributes.

        This attribute proxies all get/set calls to the translated model.

    Attribute.objects = <django.db.models.manager.ManagerFromAttributeQuerySet object>

    Attribute.product_types

    Attribute.productattribute_set

    Attribute.save(*args, **kwargs)

    Attribute.translations

    Attribute.type
        A placeholder class that provides a way to set the attribute on the model.

    Attribute.visibility_mode
        A placeholder class that provides a way to set the attribute on the model.

class shoop.core.models.AttributeType
    Bases: enumfields.enums.Enum

    An enumeration.
```



```

BOOLEAN = <AttributeType.BOOLEAN: 2>
DATE = <AttributeType.DATE: 11>
DATETIME = <AttributeType.DATETIME: 10>
DECIMAL = <AttributeType.DECIMAL: 3>
INTEGER = <AttributeType.INTEGER: 1>
TIMEDELTA = <AttributeType.TIMEDELTA: 4>
TRANSLATED_STRING = <AttributeType.TRANSLATED_STRING: 20>
UNTRANSLATED_STRING = <AttributeType.UNTRANSLATED_STRING: 21>

```

```
class shoop.core.models.AttributeVisibility
```

```
Bases: enumfields.enums.Enum
```

An enumeration.

```

HIDDEN = <AttributeVisibility.HIDDEN: 0>
NOT_VISIBLE = <AttributeVisibility.NOT_VISIBLE: 3>
SEARCHABLE_FIELD = <AttributeVisibility.SEARCHABLE_FIELD: 2>
SHOW_ON_PRODUCT_PAGE = <AttributeVisibility.SHOW_ON_PRODUCT_PAGE: 1>

```

```
class shoop.core.models.Carrier(*args, **kwargs)
```

```
Bases: shoop.core.models.ServiceProvider
```

Service provider interface for shipment processing.

Services provided by a carrier are *shipping methods*. To create a new shipping method for a carrier, use the `create_service` method.

Implementers of this interface will provide provide a list of shipping service choices and each related shipping method should have one of those service choices assigned to it.

Note: *Carrier* objects should never be created on their own but rather through a concrete subclass.

Replace Django's inheritance accessor member functions for our model (`self.__class__`) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate `base_objects` manager.

```
exception DoesNotExist
```

```
Bases: shoop.core.models._service_base.DoesNotExist
```

```
exception Carrier.MultipleObjectsReturned
```

```
Bases: shoop.core.models._service_base.MultipleObjectsReturned
```

```
Carrier.base_objects = <parler.managers.TranslatableManager object>
```

```
Carrier.customcarrier
```

```
Carrier.delete(*args, **kwargs)
```

```
Carrier.objects = <shoop.core.models._base._PolyTransManager object>
```

```
Carrier.polymorphic_primary_key_name = 'id'
Carrier.polymorphic_super_sub_accessors_replaced = False
Carrier.service_model
    alias of ShippingMethod
Carrier.serviceprovider_ptr
Carrier.shippingmethod_set

class shoop.core.models.Category(id, parent, identifier, status, image, ordering, visibility)
    Bases: mptt.models.MPTTModel, parler.models.TranslatableModel

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception Category.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    Category.add_log_entry(message, identifier=None, kind=<LogEntryKind.OTHER: 0>,
                           user=None, extra=None, save=True)

    Category.categoryfilter_set
    Category.children
    Category.description
        Descriptor for translated attributes.

        This attribute proxies all get/set calls to the translated model.

    Category.get_status_display(*moreargs, **morekwargs)
    Category.get_visibility_display(*moreargs, **morekwargs)
    Category.image
    Category.is_visible(customer)
    Category.log_entries
    Category.name
        Descriptor for translated attributes.

        This attribute proxies all get/set calls to the translated model.

    Category.objects = <shoop.core.models._categories.CategoryManager object>
    Category.parent
    Category.primary_products
    Category.primary_shop_products
    Category.save(*args, **kwargs)
    Category.shop_products
    Category.shops
    Category.slug
        Descriptor for translated attributes.

        This attribute proxies all get/set calls to the translated model.

    Category.status
        A placeholder class that provides a way to set the attribute on the model.
```

`Category.translations`

`Category.visibility`

A placeholder class that provides a way to set the attribute on the model.

`Category.visibility_groups`

class `shoop.core.models.CategoryStatus`

Bases: `enumfields.enums.Enum`

An enumeration.

DELETED = `<CategoryStatus.DELETED: 2>`

INVISIBLE = `<CategoryStatus.INVISIBLE: 0>`

VISIBLE = `<CategoryStatus.VISIBLE: 1>`

class `shoop.core.models.CategoryVisibility`

Bases: `enumfields.enums.Enum`

An enumeration.

VISIBLE_TO_ALL = `<CategoryVisibility.VISIBLE_TO_ALL: 1>`

VISIBLE_TO_GROUPS = `<CategoryVisibility.VISIBLE_TO_GROUPS: 3>`

VISIBLE_TO_LOGGED_IN = `<CategoryVisibility.VISIBLE_TO_LOGGED_IN: 2>`

class `shoop.core.models.CompanyContact` (*id, polymorphic_ctype, created_on, identifier, is_active, default_shipping_address, default_billing_address, default_shipping_method, default_payment_method, language, marketing_permission, phone, www, time-zone, prefix, name, suffix, name_ext, email, tax_group, merchant_notes, account_manager, contact_ptr, tax_number*)

Bases: `shoop.core.models.Contact`

exception `DoesNotExist`

Bases: `shoop.core.models._contacts.DoesNotExist`

exception `CompanyContact.MultipleObjectsReturned`

Bases: `shoop.core.models._contacts.MultipleObjectsReturned`

`CompanyContact.base_objects` = `<django.db.models.manager.Manager object>`

`CompanyContact.contact_ptr`

`CompanyContact.default_contact_group_identifier` = `'default_company_group'`

`CompanyContact.default_contact_group_name` = `<django.utils.functional.lazy._locals>.__proxy__ object>`

`CompanyContact.default_tax_group_getter()`

`CompanyContact.members`

`CompanyContact.objects` = `<polymorphic.managers.PolymorphicManager object>`

`CompanyContact.polymorphic_primary_key_name` = `'id'`

`CompanyContact.polymorphic_super_sub_accessors_replaced` = `False`

class `shoop.core.models.ConfigurationItem` (*id, shop, key, value*)

Bases: `shoop.core.models.ShoopModel`

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `ConfigurationItem.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`ConfigurationItem.objects` = <`django.db.models.manager.Manager` object>

`ConfigurationItem.shop`

`ConfigurationItem.value`

A placeholder class that provides a way to set the attribute on the model.

class `shoop.core.models.Contact` (*id, polymorphic_ctype, created_on, identifier, is_active, default_shipping_address, default_billing_address, default_shipping_method, default_payment_method, language, marketing_permission, phone, www, timezone, prefix, name, suffix, name_ext, email, tax_group, merchant_notes, account_manager*)

Bases: `shoop.core.models.PolymorphicShoopModel`

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `Contact.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`Contact.account_manager`

`Contact.anonymouscontact`

`Contact.base_objects` = <`django.db.models.manager.Manager` object>

`Contact.company_memberships`

`Contact.companycontact`

`Contact.contactbasketcondition_set`

`Contact.contactcondition_set`

`Contact.customer_baskets`

`Contact.customer_orders`

`Contact.default_billing_address`

`Contact.default_contact_group_identifier` = None

`Contact.default_contact_group_name` = None

`Contact.default_payment_method`

`Contact.default_shipping_address`

`Contact.default_shipping_method`

`Contact.default_tax_group_getter` = None

`Contact.full_name`

classmethod `Contact.get_default_group()`

Get or create default contact group for the class.

Identifier of the group is specified by the class property `default_contact_group_identifier`.

If new group is created, its name is set to value of `default_contact_group_name` class property.

Return type `core.models.ContactGroup`

`Contact.get_language_display` (*moreargs, **morekwargs)

`Contact.get_next_by_created_on` (*moreargs, **morekwargs)

`Contact.get_previous_by_created_on(*moreargs, **morekwargs)`

`Contact.get_price_display_options()`

Get price display options of the contact.

If the default group (`get_default_group`) defines price display options and the contact is member of it, return it.

If contact is not (anymore) member of the default group or the default group does not define options, return one of the groups which defines options. If there is more than one such groups, it is undefined which options will be used.

If contact is not a member of any group that defines price display options, return default constructed `PriceDisplayOptions`.

Subclasses may still override this default behavior.

Return type *PriceDisplayOptions*

`Contact.get_timezone_display(*moreargs, **morekwargs)`

`Contact.groups`

`Contact.is_all_seeing = False`

`Contact.is_anonymous = False`

`Contact.objects = <polymorphic.managers.PolymorphicManager object>`

`Contact.personcontact`

`Contact.polymorphic_ctype`

`Contact.polymorphic_primary_key_name = 'id'`

`Contact.polymorphic_super_sub_accessors_replaced = False`

`Contact.save(*args, **kwargs)`

`Contact.savedaddress_set`

`Contact.shop_set`

`Contact.tax_group`

`Contact.timezone`

A placeholder class that provides a way to set the attribute on the model.

class `shoop.core.models.ContactGroup(id, identifier, show_pricing, show_prices_including_taxes, hide_prices)`

Bases: *shoop.core.models.TranslatableShoopModel*

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `ContactGroup.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`ContactGroup.can_delete()`

`ContactGroup.cgpprice_set`

`ContactGroup.contactgroupbasketcondition_set`

`ContactGroup.contactgroupcondition_set`

`ContactGroup.delete(*args, **kwargs)`

`ContactGroup.get_price_display_options()`

```
ContactGroup.groupavailabilitybehaviorcomponent_set
ContactGroup.members
ContactGroup.name
    Descriptor for translated attributes.

    This attribute proxies all get/set calls to the translated model.
ContactGroup.objects = <django.db.models.manager.ManagerFromContactGroupQuerySet object>
ContactGroup.translations
ContactGroup.visible_categories
ContactGroup.visible_products

class shoop.core.models.GroupAvailabilityBehaviorComponent (id, polymorphic_ctype,
                                                            servicebehaviorcomponent_ptr)

Bases: shoop.core.models.ServiceBehaviorComponent

exception DoesNotExist
    Bases: shoop.core.models._service_base.DoesNotExist

exception GroupAvailabilityBehaviorComponent.MultipleObjectsReturned
    Bases: shoop.core.models._service_base.MultipleObjectsReturned

GroupAvailabilityBehaviorComponent.base_objects = <django.db.models.manager.Manager object>
GroupAvailabilityBehaviorComponent.get_unavailability_reasons (service,
                                                                source)

GroupAvailabilityBehaviorComponent.groups
GroupAvailabilityBehaviorComponent.help_text = <django.utils.functional.lazy.<locals>.__proxy__ object>
GroupAvailabilityBehaviorComponent.name = <django.utils.functional.lazy.<locals>.__proxy__ object>
GroupAvailabilityBehaviorComponent.objects = <polymorphic.managers.PolymorphicManager object>
GroupAvailabilityBehaviorComponent.polymorphic_primary_key_name = 'id'
GroupAvailabilityBehaviorComponent.polymorphic_super_sub_accessors_replaced = False
GroupAvailabilityBehaviorComponent.servicebehaviorcomponent_ptr

class shoop.core.models.Counter (id, value)
    Bases: django.db.models.base.Model

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception Counter.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    classmethod Counter.get_and_increment (id)

    Counter.get_id_display (*moreargs, **morekwargs)

    Counter.id
        A placeholder class that provides a way to set the attribute on the model.

    Counter.objects = <django.db.models.manager.Manager object>

class shoop.core.models.CounterType
    Bases: enumfields.enums.Enum
```

An enumeration.

ORDER_REFERENCE = <CounterType.ORDER_REFERENCE: 1>

class shoop.core.models.**CustomCarrier** (*args, **kwargs)

Bases: *shoop.core.models.Carrier*

Carrier without any integration or special processing.

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

exception DoesNotExist

Bases: shoop.core.models._service_shipping.DoesNotExist

exception CustomCarrier.MultipleObjectsReturned

Bases: shoop.core.models._service_shipping.MultipleObjectsReturned

CustomCarrier.**base_objects** = <parler.managers.TranslatableManager object>

CustomCarrier.**carrier_ptr**

CustomCarrier.**carrierwithcheckoutphase**

CustomCarrier.**get_service_choices**()

CustomCarrier.**objects** = <shoop.core.models._base._PolyTransManager object>

CustomCarrier.**polymorphic_primary_key_name** = 'id'

CustomCarrier.**polymorphic_super_sub_accessors_replaced** = False

class shoop.core.models.**CustomerTaxGroup** (id, identifier, enabled)

Bases: *shoop.core.models.TranslatableShoopModel*

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception CustomerTaxGroup.MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

CustomerTaxGroup.**contact_set**

classmethod CustomerTaxGroup.**get_default_company_group**()

classmethod CustomerTaxGroup.**get_default_person_group**()

CustomerTaxGroup.**name**

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

CustomerTaxGroup.**objects** = <parler.managers.TranslatableManager object>

CustomerTaxGroup.**taxrule_set**

CustomerTaxGroup.**translations**

```
class shoop.core.models.CustomPaymentProcessor(*args, **kwargs)
```

```
    Bases: shoop.core.models.PaymentProcessor
```

Payment processor without any integration or special processing.

Can be used for payment methods whose payments are processed manually.

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

```
exception DoesNotExist
```

```
    Bases: shoop.core.models._service_payment.DoesNotExist
```

```
exception CustomPaymentProcessor.MultipleObjectsReturned
```

```
    Bases: shoop.core.models._service_payment.MultipleObjectsReturned
```

```
CustomPaymentProcessor.base_objects = <parler.managers.TranslatableManager object>
```

```
CustomPaymentProcessor.get_service_choices()
```

```
CustomPaymentProcessor.objects = <shoop.core.models._base._PolyTransManager object>
```

```
CustomPaymentProcessor.paymentprocessor_ptr
```

```
CustomPaymentProcessor.paymentwithcheckoutphase
```

```
CustomPaymentProcessor.polymorphic_primary_key_name = 'id'
```

```
CustomPaymentProcessor.polymorphic_super_sub_accessors_replaced = False
```

```
CustomPaymentProcessor.process_payment_return_request(service, order, request)
```

```
class shoop.core.models.FixedCostBehaviorComponent(id, polymorphic_ctype, servicebehaviorcomponent_ptr, price_value)
```

```
    Bases: shoop.core.models._service_base.TranslatableServiceBehaviorComponent
```

```
exception DoesNotExist
```

```
    Bases: django.core.exceptions.ObjectDoesNotExist
```

```
exception FixedCostBehaviorComponent.MultipleObjectsReturned
```

```
    Bases: django.core.exceptions.MultipleObjectsReturned
```

```
FixedCostBehaviorComponent.base_objects = <parler.managers.TranslatableManager object>
```

```
FixedCostBehaviorComponent.description
```

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

```
FixedCostBehaviorComponent.get_costs(service, source)
```

```
FixedCostBehaviorComponent.help_text = <django.utils.functional.lazy.<locals>.__proxy__ object>
```

```
FixedCostBehaviorComponent.name = <django.utils.functional.lazy.<locals>.__proxy__ object>
```

```
FixedCostBehaviorComponent.objects = <shoop.core.models._base._PolyTransManager object>
```

```
FixedCostBehaviorComponent.polymorphic_primary_key_name = 'id'
```



```
FixedCostBehaviorComponent.polymorphic_super_sub_accessors_replaced = False
```

```
FixedCostBehaviorComponent.servicebehaviorcomponent_ptr
```

```
FixedCostBehaviorComponent.translations
```

```
shoop.core.models.get_company_contact(user)
```

Get preferred CompanyContact of given user.

If user has associated PersonContact which is member of CompanyContact, return CompanyContact. Otherwise, return None.

Parameters *user* (*django.contrib.auth.models.User* | *None*) – User object (or None)
to get contact for

Returns CompanyContact (or none) of which user’s PersonContact is a member

Return type CompanyContact|None

```
shoop.core.models.get_person_contact(user)
```

Get PersonContact of given user.

If given user is non-zero (evaluates true as bool) and not anonymous, return the PersonContact of the user. If there is no PersonContact for the user yet, create it first. When this creation happens, details (name, email, is_active) are copied from the user.

If given user is None (or otherwise evaluates as false) or anonymous, return the AnonymousContact.

Parameters *user* (*django.contrib.auth.models.User* | *None*) – User object (or None)
to get contact for

Returns PersonContact of the user or AnonymousContact

Return type PersonContact|AnonymousContact

```
class shoop.core.models.Gender
```

Bases: *enumfields.enums.Enum*

An enumeration.

FEMALE = <Gender.FEMALE: ‘f’>

MALE = <Gender.MALE: ‘m’>

OTHER = <Gender.OTHER: ‘o’>

UNDISCLOSED = <Gender.UNDISCLOSED: ‘u’>

```
class shoop.core.models.ImmutableAddress(*args, **kwargs)
```

Bases: *shoop.core.models._base.ChangeProtected*, *shoop.core.models._addresses.Address*

An address that can not be changed.

Immutable addresses are used for orders, etc., where subsequent edits to the original address (for example an user’s default address) must not affect past business data.

Immutable addresses can be created directly, with the *from_data()* method, or by creating an immutable copy of an existing *MutableAddress* with the *Address.to_immutable()* method.

exception DoesNotExist

Bases: *django.core.exceptions.ObjectDoesNotExist*

exception ImmutableAddress.MultipleObjectsReturned

Bases: *django.core.exceptions.MultipleObjectsReturned*

ImmutableAddress.billing_orders

`ImmutableAddress.country`

A descriptor for country fields on a model instance. Returns a `Country` when accessed so you can do things like:

```
>>> from people import Person
>>> person = Person.object.get(name='Chris')

>>> person.country.name
'New Zealand'

>>> person.country.flag
'/static/flags/nz.gif'
```

classmethod `ImmutableAddress.from_data(data)`

Get or create immutable address with given data.

Parameters `data` (`dict[str, str]`) – data for address

Returns Saved immutable address

Return type `ImmutableAddress`

`ImmutableAddress.get_country_display(*moreargs, **morekwargs)`

`ImmutableAddress.objects = <django.db.models.manager.Manager object>`

`ImmutableAddress.shipping_orders`

`ImmutableAddress.to_immutable()`

class `shoop.core.models.Manufacturer(id, created_on, identifier, name, url)`

Bases: `django.db.models.base.Model`

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `Manufacturer.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`Manufacturer.get_next_by_created_on(*moreargs, **morekwargs)`

`Manufacturer.get_previous_by_created_on(*moreargs, **morekwargs)`

`Manufacturer.objects = <django.db.models.manager.Manager object>`

`Manufacturer.product_set`

class `shoop.core.models.MutableAddress(*args, **kwargs)`

Bases: `shoop.core.models._addresses.Address`

An address that can be changed.

Mutable addresses are used for e.g. contact's saved addresses. They are saved as new immutable addresses when used in e.g. orders.

Mutable addresses can be created with `MutableAddress.from_data` or with the `to_mutable` method of `Address` objects.

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `MutableAddress.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

MutableAddress.country

A descriptor for country fields on a model instance. Returns a Country when accessed so you can do things like:

```
>>> from people import Person
>>> person = Person.object.get(name='Chris')

>>> person.country.name
'New Zealand'

>>> person.country.flag
'/static/flags/nz.gif'
```

classmethod MutableAddress.from_data(data)

Construct mutable address from a data dictionary.

Parameters *data* (*dict[str, str]*) – data for address

Returns Unsaved mutable address

Return type *MutableAddress*

MutableAddress.get_country_display (*moreargs, **morekwargs)

MutableAddress.objects = <django.db.models.manager.Manager object>

MutableAddress.saved_addresses

MutableAddress.shop_set

class shoop.core.models.**Order**(*id, shop, created_on, modified_on, identifier, label, key, reference_number, customer, orderer, billing_address, shipping_address, tax_number, phone, email, creator, modified_by, deleted, status, payment_status, shipping_status, payment_method, payment_method_name, payment_data, shipping_method, shipping_method_name, shipping_data, extra_data, taxful_total_price_value, taxless_total_price_value, currency, prices_include_tax, display_currency, display_currency_rate, ip_address, order_date, payment_date, language, customer_comment, admin_comment, require_verification, all_verified, marketing_permission, _codes*)

Bases: *shoop.utils.properties.MoneyPropped*, *django.db.models.base.Model*

exception DoesNotExist

Bases: *django.core.exceptions.ObjectDoesNotExist*

exception Order.MultipleObjectsReturned

Bases: *django.core.exceptions.MultipleObjectsReturned*

Order.add_log_entry (*message, identifier=None, kind=<LogEntryKind.OTHER: 0>, user=None, extra=None, save=True*)

Order.billing_address

Order.cache_prices ()

Order.can_create_refund ()

Order.can_edit ()

Order.can_set_canceled ()

Order.can_set_complete ()

Order.check_all_verified ()

`Order.codes`

`Order.common_select_related = ('billing_address',)`

`Order.coupon_usages`

`Order.create_full_refund(restock_products=False)`

Create a full for entire order contents, with the option of restocking stocked products.

Parameters `restock_products` (*bool* | *False*) – Boolean indicating whether to restock products

`Order.create_payment(amount, payment_identifier=None, description='')`

Create a payment with given amount for this order.

If the order already has payments and sum of their amounts is equal or greater than `self.taxful_total_price`, an exception is raised.

If the end sum of all payments is equal or greater than `self.taxful_total_price`, then the order is marked as paid.

Parameters

- **amount** (*Money*) – Amount of the payment to be created
- **payment_identifier** (*str* | *None*) – Identifier of the created payment. If not set, default value of “gateway_id:order_id:number” will be used (where number is number of payments in the order).
- **description** (*str*) – Description of the payment. Will be set to `method` property of the created payment.

Returns The created Payment object

Return type *shoop.core.models.Payment*

`Order.create_refund(refund_data, created_by=None)`

Create a refund if passed a list of refund line data.

Refund line data is simply a list of dictionaries where each dictionary contains data for a particular refund line.

If refund line data includes a parent line, the refund is associated with that line and cannot exceed the line amount.

Additionally, if the parent line is of enum type *OrderLineType.PRODUCT* and the `restock_products` boolean flag is set to `True`, the products will be restocked with the order's supplier the exact amount of the value of the `quantity` field.

Parameters

- **refund_data** (*[dict]*) – List of dicts containing refund data.
- **created_by** (*django.contrib.auth.User* | *None*) – Refund creator's user instance, used for adjusting supplier stock.

`Order.create_shipment(product_quantities, supplier=None, shipment=None)`

Create a shipment for this order from `product_quantities`. `product_quantities` is expected to be a dict mapping Product instances to quantities.

Only quantities over 0 are taken into account, and if the mapping is empty or has no quantity value over 0, `NoProductsToShipException` will be raised.

Parameters

- **product_quantities** (*dict[shoop.shop.models.Product, decimal.Decimal]*) – a dict mapping Product instances to quantities to ship
- **supplier** – Optional Supplier for this product. No validation is made as to whether the given supplier supplies the products.
- **shipment** – Optional unsaved Shipment for ShipmentProduct's. If not given Shipment is created based on supplier parameter.

Raises NoProductsToShipException

Returns Saved, complete Shipment object

Return type *shoop.core.models.Shipment*

`Order.create_shipment_of_all_products (supplier=None)`

Create a shipment of all the products in this Order, no matter whether or not any have been previously marked as shipped or not.

See the documentation for *create_shipment*.

Parameters **supplier** – The Supplier to use. If *None*, the first supplier in the order is used. (If several are in the order, this fails.)

Returns Saved, complete Shipment object

Return type *shoop.shop.models.Shipment*

`Order.creator`

`Order.customer`

`Order.delete (using=None)`

`Order.extra_data`

A placeholder class that provides a way to set the attribute on the model.

`Order.full_clean (exclude=None, validate_unique=True)`

`Order.get_known_additional_data ()`

Get a list of “known additional data” in this order’s *payment_data*, *shipping_data* and *extra_data*. The list is returned in the order the fields are specified in the settings entries for said known keys. *dict(that_list)* can of course be used to “flatten” the list into a dict. :return: list of 2-tuples.

`Order.get_language_display (*moreargs, **morekwargs)`

`Order.get_next_by_created_on (*moreargs, **morekwargs)`

`Order.get_next_by_modified_on (*moreargs, **morekwargs)`

`Order.get_next_by_order_date (*moreargs, **morekwargs)`

`Order.get_payment_status_display (*moreargs, **morekwargs)`

`Order.get_previous_by_created_on (*moreargs, **morekwargs)`

`Order.get_previous_by_modified_on (*moreargs, **morekwargs)`

`Order.get_previous_by_order_date (*moreargs, **morekwargs)`

`Order.get_product_ids_and_quantities ()`

`Order.get_product_summary ()`

Return a dict of product IDs -> {ordered, unshipped, shipped}

`Order.get_purchased_attachments ()`

`Order.get_shipping_status_display (*moreargs, **morekwargs)`

```
Order.get_status_display()
Order.get_tax_summary()
    Return type taxing.TaxSummary
Order.get_total_paid_amount()
Order.get_total_refunded_amount()
Order.get_total_unpaid_amount()
Order.get_total_unrefunded_amount()
Order.get_tracking_codes()
Order.get_unshipped_products()
Order.has_refunds()
Order.is_canceled()
Order.is_complete()
Order.is_paid()
Order.lines
Order.log_entries
Order.modified_by
Order.objects = <django.db.models.manager.ManagerFromOrderQuerySet object>
Order.orderer
Order.payment_data
    A placeholder class that provides a way to set the attribute on the model.
Order.payment_method
Order.payment_status
    A placeholder class that provides a way to set the attribute on the model.
Order.payments
Order.save(*args, **kwargs)
Order.set_canceled()
Order.shipments
Order.shipping_address
Order.shipping_data
    A placeholder class that provides a way to set the attribute on the model.
Order.shipping_method
Order.shipping_status
    A placeholder class that provides a way to set the attribute on the model.
Order.shop
Order.status
Order.taxful_total_price
Order.taxless_total_price
```

```

    Order.update_shipping_status()
class shoop.core.models.OrderLine(id, order, product, supplier, parent_line, ordering, type,
                                   sku, text, accounting_identifier, require_verification, ver-
                                   ified, extra_data, quantity, base_unit_price_value, dis-
                                   count_amount_value)
Bases:      shoop.utils.properties.MoneyPropped,      django.db.models.base.Model,
            shoop.core.pricing.Priceful

exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception OrderLine.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

OrderLine.base_unit_price
    Property for Price object.

    Similar to MoneyProperty but also has includes_tax field.

    Operators with TaxfulPrice and TaxlessPrice objects.

OrderLine.child_lines

OrderLine.discount_amount
    Property for Price object.

    Similar to MoneyProperty but also has includes_tax field.

    Operators with TaxfulPrice and TaxlessPrice objects.

OrderLine.extra_data
    A placeholder class that provides a way to set the attribute on the model.

OrderLine.get_type_display(*moreargs, **morekwargs)

OrderLine.max_refundable_amount
    Return type shoop.utils.money.Money

OrderLine.objects = <shoop.core.models_order_lines.OrderLineManager object>

OrderLine.order

OrderLine.parent_line

OrderLine.product

OrderLine.save(*args, **kwargs)

OrderLine.supplier

OrderLine.tax_amount
    Return type shoop.utils.money.Money

OrderLine.taxes

OrderLine.type
    A placeholder class that provides a way to set the attribute on the model.
class shoop.core.models.OrderLineTax(id, order_line, tax, name, amount_value,
                                       base_amount_value, ordering)
Bases:      shoop.utils.properties.MoneyPropped,      shoop.core.models.ShoopModel,
            shoop.core.taxing.LineTax

```

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception OrderLineTax.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

OrderLineTax.amount

Property for a Money amount.

Will return `Money` objects when the property is being get and accepts `Money` objects on set. Value and currency are read/written from/to other fields.

Fields are given as locators, that is a string in dotted format, e.g. locator `"foo.bar"` points to `instance.foo.bar` where `instance` is an instance of the class owning the `MoneyProperty`.

Setting value of this property to a `Money` object with different currency that is currently set (in the field pointed by the currency locator), will raise an `UnitMixupError`.

OrderLineTax.base_amount

Property for a Money amount.

Will return `Money` objects when the property is being get and accepts `Money` objects on set. Value and currency are read/written from/to other fields.

Fields are given as locators, that is a string in dotted format, e.g. locator `"foo.bar"` points to `instance.foo.bar` where `instance` is an instance of the class owning the `MoneyProperty`.

Setting value of this property to a `Money` object with different currency that is currently set (in the field pointed by the currency locator), will raise an `UnitMixupError`.

OrderLineTax.objects = <django.db.models.manager.Manager object>

OrderLineTax.order_line

OrderLineTax.tax

class shoop.core.models.OrderLineType

Bases: `enumfields.enums.Enum`

An enumeration.

DISCOUNT = <OrderLineType.DISCOUNT: 4>

OTHER = <OrderLineType.OTHER: 5>

PAYMENT = <OrderLineType.PAYMENT: 3>

PRODUCT = <OrderLineType.PRODUCT: 1>

REFUND = <OrderLineType.REFUND: 6>

SHIPPING = <OrderLineType.SHIPPING: 2>

class shoop.core.models.OrderLogEntry (*id, created_on, user, message, identifier, kind, extra, target*)

Bases: `shoop.utils.analog.BaseLogEntry`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception OrderLogEntry.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

OrderLogEntry.extra

A placeholder class that provides a way to set the attribute on the model.

OrderLogEntry.get_kind_display (**moreargs, **morekwargs*)


```

OrderLogEntry.get_next_by_created_on(*moreargs, **morekwargs)
OrderLogEntry.get_previous_by_created_on(*moreargs, **morekwargs)
OrderLogEntry.kind
    A placeholder class that provides a way to set the attribute on the model.
OrderLogEntry.logged_model
    alias of Order
OrderLogEntry.objects = <django.db.models.manager.Manager object>
OrderLogEntry.target
OrderLogEntry.user

class shoop.core.models.OrderStatus(id, identifier, ordering, role, default)
    Bases: parler.models.TranslatableModel

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception OrderStatus.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    OrderStatus.get_role_display(*moreargs, **morekwargs)

    OrderStatus.name
        Descriptor for translated attributes.

        This attribute proxies all get/set calls to the translated model.

    OrderStatus.objects = <django.db.models.manager.ManagerFromOrderStatusQuerySet object>
    OrderStatus.order_set
    OrderStatus.role
        A placeholder class that provides a way to set the attribute on the model.

    OrderStatus.save(*args, **kwargs)
    OrderStatus.translations

class shoop.core.models.OrderStatusRole
    Bases: enumfields.enums.Enum

    An enumeration.

    CANCELED = <OrderStatusRole.CANCELED: 3>
    COMPLETE = <OrderStatusRole.COMPLETE: 2>
    INITIAL = <OrderStatusRole.INITIAL: 1>
    NONE = <OrderStatusRole.NONE: 0>

class shoop.core.models.Payment(id, order, created_on, gateway_id, payment_identifier,
                                amount_value, foreign_amount_value, foreign_currency, de-
                                scription)
    Bases: shoop.utils.properties.MoneyPropped, django.db.models.base.Model

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception Payment.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

```

`Payment.amount`

Property for a Money amount.

Will return `Money` objects when the property is being get and accepts `Money` objects on set. Value and currency are read/written from/to other fields.

Fields are given as locators, that is a string in dotted format, e.g. locator `"foo.bar"` points to `instance.foo.bar` where `instance` is an instance of the class owning the `MoneyProperty`.

Setting value of this property to a `Money` object with different currency that is currently set (in the field pointed by the currency locator), will raise an `UnitMixupError`.

`Payment.foreign_amount`

Property for a Money amount.

Will return `Money` objects when the property is being get and accepts `Money` objects on set. Value and currency are read/written from/to other fields.

Fields are given as locators, that is a string in dotted format, e.g. locator `"foo.bar"` points to `instance.foo.bar` where `instance` is an instance of the class owning the `MoneyProperty`.

Setting value of this property to a `Money` object with different currency that is currently set (in the field pointed by the currency locator), will raise an `UnitMixupError`.

`Payment.get_next_by_created_on(*moreargs, **morekwargs)`

`Payment.get_previous_by_created_on(*moreargs, **morekwargs)`

`Payment.objects = <django.db.models.manager.Manager object>`

`Payment.order`

class `shoop.core.models.PaymentMethod`(*id, identifier, enabled, shop, choice_identifier, old_module_identifier, old_module_data, logo, tax_class, payment_processor*)

Bases: `shoop.core.models.Service`

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `PaymentMethod.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`PaymentMethod.behavior_components`

`PaymentMethod.can_delete()`

`PaymentMethod.contact_set`

`PaymentMethod.description`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`PaymentMethod.get_payment_process_response(order, urls)`

`PaymentMethod.line_type = <OrderLineType.PAYMENT: 3>`

`PaymentMethod.logo`

`PaymentMethod.name`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`PaymentMethod.objects = <django.db.models.manager.ManagerFromServiceQuerySet object>`

`PaymentMethod.old_module_data`

A placeholder class that provides a way to set the attribute on the model.

`PaymentMethod.payment_orders`

`PaymentMethod.payment_processor`

`PaymentMethod.payment_products`

`PaymentMethod.process_payment_return_request` (*order, request*)

`PaymentMethod.provider_attr = 'payment_processor'`

`PaymentMethod.shop`

`PaymentMethod.shop_product_m2m = 'payment_methods'`

`PaymentMethod.tax_class`

`PaymentMethod.translations`

class `shoop.core.models.PaymentProcessor` (**args, **kwargs*)

Bases: `shoop.core.models.ServiceProvider`

Service provider interface for payment processing.

Services provided by a payment processor are *payment methods*. To create a new payment method for a payment processor, use the `create_service` method.

Implementers of this interface will provide provide a list of payment service choices and each related payment method should have one of those service choices assigned to it.

Payment processing is handled with *get_payment_process_response* and *process_payment_return_request* methods.

Note: *PaymentProcessor* objects should never be created on their own but rather through a concrete subclass.

Replace Django's inheritance accessor member functions for our model (`self.__class__`) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: `modelb` - ModelB: `modela_ptr`, `modelb`, `modelc` - ModelC: `modela_ptr`, `modelb`, `modelb_ptr`, `modelc`

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate `base_objects` manager.

exception `DoesNotExist`

Bases: `shoop.core.models._service_base.DoesNotExist`

exception `PaymentProcessor.MultipleObjectsReturned`

Bases: `shoop.core.models._service_base.MultipleObjectsReturned`

`PaymentProcessor.base_objects = <parler.managers.TranslatableManager object>`

`PaymentProcessor.custompaymentprocessor`

`PaymentProcessor.delete` (**args, **kwargs*)

`PaymentProcessor.get_payment_process_response` (*service, order, urls*)

Get payment process response for given order.

Return type `django.http.HttpResponseNone`

```
PaymentProcessor.objects = <shoop.core.models._base._PolyTransManager object>
PaymentProcessor.paymentmethod_set
PaymentProcessor.polymorphic_primary_key_name = 'id'
PaymentProcessor.polymorphic_super_sub_accessors_replaced = False
PaymentProcessor.process_payment_return_request (service, order, request)
    Process payment return request for given order.

    Should set order.payment_status. Default implementation just sets it to DEFERRED if it is
    NOT_PAID.

    Return type None
PaymentProcessor.pseudopaymentprocessor
PaymentProcessor.service_model
    alias of PaymentMethod
PaymentProcessor.serviceprovider_ptr
class shoop.core.models.PaymentStatus
    Bases: enumfields.enums.Enum

    An enumeration.

    CANCELED = <PaymentStatus.CANCELED: 3>
    DEFERRED = <PaymentStatus.DEFERRED: 4>
    FULLY_PAID = <PaymentStatus.FULLY_PAID: 2>
    NOT_PAID = <PaymentStatus.NOT_PAID: 0>
    PARTIALLY_PAID = <PaymentStatus.PARTIALLY_PAID: 1>
class shoop.core.models.PaymentUrls (payment_url, return_url, cancel_url)
    Bases: object

    Container for URLs used in payment processing.
class shoop.core.models.PersistentCacheEntry (id, module, key, time, data)
    Bases: django.db.models.base.Model

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist
    exception PersistentCacheEntry.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned
PersistentCacheEntry.data
    A placeholder class that provides a way to set the attribute on the model.
PersistentCacheEntry.get_next_by_time (*moreargs, **morekwargs)
PersistentCacheEntry.get_previous_by_time (*moreargs, **morekwargs)
PersistentCacheEntry.objects = <django.db.models.manager.Manager object>
```

```

class shoop.core.models.PersonContact (id, polymorphic_ctype, created_on, identifier, is_active,
                                         default_shipping_address, default_billing_address, de-
                                         fault_shipping_method, default_payment_method, lan-
                                         guage, marketing_permission, phone, www, timezone,
                                         prefix, name, suffix, name_ext, email, tax_group, mer-
                                         chant_notes, account_manager, contact_ptr, user, gen-
                                         der, birth_date, first_name, last_name)

Bases: shoop.core.models.Contact

exception DoesNotExist
    Bases: shoop.core.models._contacts.DoesNotExist

exception PersonContact.MultipleObjectsReturned
    Bases: shoop.core.models._contacts.MultipleObjectsReturned

PersonContact.base_objects = <django.db.models.manager.Manager object>

PersonContact.contact_ptr

PersonContact.contact_set

PersonContact.default_contact_group_identifier = 'default_person_group'

PersonContact.default_contact_group_name = <django.utils.functional.lazy.<locals>.__proxy__ object>

PersonContact.default_tax_group_getter()

PersonContact.gender
    A placeholder class that provides a way to set the attribute on the model.

PersonContact.get_gender_display(*moreargs, **morekwargs)

PersonContact.is_all_seeing

PersonContact.name

PersonContact.objects = <polymorphic.managers.PolymorphicManager object>

PersonContact.orderer_baskets

PersonContact.orderer_orders

PersonContact.polymorphic_primary_key_name = 'id'

PersonContact.polymorphic_super_sub_accessors_replaced = False

PersonContact.save(*args, **kwargs)

PersonContact.user

class shoop.core.models.PolymorphicShoopModel (*args, **kwargs)
    Bases: polymorphic.models.PolymorphicModel, shoop.core.models.ShoopModel

    Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions.
    We monkey patch them until a patch can be added to Django (which would probably be very small and make all
    of this obsolete).

    If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: -
    ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

    These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at
    hand.

    The original Django accessors use our polymorphic manager. But they should not. So we replace them with our
    own accessors that use our appropriate base_objects manager.

```

```
class Meta
    Bases: object

    abstract = False

PolymorphicShoopModel.base_objects

PolymorphicShoopModel.objects

PolymorphicShoopModel.polymorphic_ctype

PolymorphicShoopModel.polymorphic_super_sub_accessors_replaced = False
```

class `shoop.core.models.PolymorphicTranslatableShoopModel` (**args, **kwargs*)
Bases: *shoop.core.models.PolymorphicShoopModel, shoop.core.models.TranslatableShoopModel*

Replace Django's inheritance accessor member functions for our model (`self.__class__`) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

```
class Meta
    Bases: object

    abstract = False

PolymorphicTranslatableShoopModel.base_objects

PolymorphicTranslatableShoopModel.objects

PolymorphicTranslatableShoopModel.polymorphic_ctype

PolymorphicTranslatableShoopModel.polymorphic_super_sub_accessors_replaced = False
```

class `shoop.core.models.Product` (*id, created_on, modified_on, deleted, mode, variation_parent, stock_behavior, shipping_mode, sales_unit, tax_class, type, sku, gtin, barcode, accounting_identifier, profit_center, cost_center, category, width, height, depth, net_weight, gross_weight, manufacturer, primary_image*)
Bases: *shoop.core.taxing.TaxableItem, shoop.core.models._attributes.AttributableMixin, parler.models.TranslatableModel*

COMMON_SELECT_RELATED = ('type', 'primary_image', 'tax_class')

exception `DoesNotExist`
Bases: *django.core.exceptions.ObjectDoesNotExist*

exception `Product.MultipleObjectsReturned`
Bases: *django.core.exceptions.MultipleObjectsReturned*

`Product.add_log_entry` (*message, identifier=None, kind=<LogEntryKind.OTHER: 0>, user=None, extra=None, save=True*)

`Product.attributes`

`Product.category`

`Product.clear_variation()`
Fully remove variation information.

Make this product a non-variation parent.

`Product.cross_sell_1`

`Product.cross_sell_2`

`Product.delete (using=None)`

`Product.description`
Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Product.discountfromproduct_set`

`Product.freeproductline_set`

`Product.get_all_available_combinations()`
Generate all available combinations of variation variables.

If the product is not a variable variation parent, the iterator is empty.

Because of possible combinatorial explosion this is a generator function. (For example 6 variables with 5 options each explodes to 15,625 combinations.)

Returns Iterable of combination information dicts.

Return type Iterable[dict]

`Product.get_all_package_children()`

`Product.get_all_package_parents()`

`Product.get_available_attribute_queryset()`

`Product.get_available_variation_results()`
Get a dict of `combination_hash` to product ID of variable variation results.

Returns Mapping of combination hashes to product IDs

Return type dict[str, int]

`Product.get_base_price (context, quantity=1)`
Get base price of the product within given context.

Base price differs from the (effective) price when there are discounts in effect.

Return type [*shoop.core.pricing.Price*](#)

`Product.get_cheapest_child_price (context, quantity=1)`

`Product.get_cheapest_child_price_info (context, quantity=1)`
Get the `PriceInfo` of the cheapest variation child

The attribute used for sorting is `PriceInfo.price`.

Return `None` if `self.variation_children` do not exist. This is because we cannot return anything sensible.

Return type [*shoop.core.pricing.PriceInfo*](#)

`Product.get_child_price_range (context, quantity=1)`
Get the prices for cheapest and the most expensive child

The attribute used for sorting is `PriceInfo.price`.

Return (`None`, `None`) if `self.variation_children` do not exist. This is because we cannot return anything sensible.

Returns a tuple of prices

Return type (`shoop.core.pricing.Price`, `shoop.core.pricing.Price`)

`Product.get_mode_display(*moreargs, **morekwargs)`

`Product.get_next_by_created_on(*moreargs, **morekwargs)`

`Product.get_next_by_modified_on(*moreargs, **morekwargs)`

`Product.get_package_child_to_quantity_map()`

`Product.get_previous_by_created_on(*moreargs, **morekwargs)`

`Product.get_previous_by_modified_on(*moreargs, **morekwargs)`

`Product.get_price(context, quantity=1)`

Get price of the product within given context.

Note: When the current pricing module implements pricing steps, it is possible that `p.get_price(ctx) * 123` is not equal to `p.get_price(ctx, quantity=123)`, since there could be quantity discounts in effect, but usually they are equal.

Return type *shoop.core.pricing.Price*

`Product.get_price_info(context, quantity=1)`

Get `PriceInfo` object for the product in given context.

Returned `PriceInfo` object contains calculated price and `base_price`. The calculation of prices is handled in the current pricing module.

Return type *shoop.core.pricing.PriceInfo*

`Product.get_priced_children(context, quantity=1)`

Get child products with price infos sorted by price.

Return type `list[(Product, PriceInfo)]`

Returns List of products and their price infos sorted from cheapest to most expensive.

`Product.get_public_media()`

`Product.get_shipping_mode_display(*moreargs, **morekwargs)`

`Product.get_shop_instance(shop)`

Return type *shoop.core.models.ShopProduct*

`Product.get_stock_behavior_display(*moreargs, **morekwargs)`

`Product.get_variation_siblings()`

`Product.is_package_child()`

`Product.is_package_parent()`

`Product.is_stocked()`

`Product.is_variation_child()`

`Product.is_variation_parent()`

`Product.keywords`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Product.link_to_parent` (*parent, variables=None, combination_hash=None*)

Parameters

- **parent** (`Product`) – The parent to link to.
- **variables** (*dict/None*) – Optional dict of {variable identifier: value identifier} for complex variable linkage
- **combination_hash** (*str/None*) – Optional combination hash (for variable variations), if precomputed. Mutually exclusive with `variables`

`Product.log_entries`

`Product.make_package` (*package_def*)

`Product.manufacturer`

`Product.media`

`Product.mode`

A placeholder class that provides a way to set the attribute on the model.

`Product.name`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Product.objects` = `<django.db.models.manager.ManagerFromProductQuerySet object>`

`Product.order_lines`

`Product.primary_image`

`Product.productfilter_set`

`Product.productsinbasketcondition_set`

`Product.sales_unit`

`Product.save` (**args, **kwargs*)

`Product.shipments`

`Product.shipping_mode`

A placeholder class that provides a way to set the attribute on the model.

`Product.shop_products`

`Product.simplify_variation()`

Remove variation variables from the given variation parent, turning it into a simple variation (or a normal product, if it has no children).

Parameters `product` (`shoop.core.models.Product`) – Variation parent to not be variable any longer.

`Product.slug`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Product.soft_delete` (*user=None*)

`Product.status_text`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Product.stock_behavior`

A placeholder class that provides a way to set the attribute on the model.

`Product.storedbasket_set`

`Product.suppliedproduct_set`

`Product.tax_class`

`Product.translations`

`Product.type`

`Product.unlink_from_parent()`

`Product.variation_children`

`Product.variation_name`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Product.variation_parent`

`Product.variation_result_subs`

`Product.variation_result_supers`

`Product.variation_variables`

`Product.verify_mode()`

```
class shoop.core.models.Product(id, created_on, modified_on, deleted, mode, variation_parent,
                                stock_behavior, shipping_mode, sales_unit, tax_class, type, sku,
                                gtin, barcode, accounting_identifier, profit_center, cost_center,
                                category, width, height, depth, net_weight, gross_weight, manu-
                                facturer, primary_image)
```

Bases: `shoop.core.taxing.TaxableItem`, `shoop.core.models._attributes.AttributableMixin`,
`parler.models.TranslatableModel`

COMMON_SELECT_RELATED = ('type', 'primary_image', 'tax_class')

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception Product.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`Product.add_log_entry`(message, identifier=None, kind=<LogEntryKind.OTHER: 0>,
user=None, extra=None, save=True)

`Product.attributes`

`Product.category`

`Product.clear_variation()`

Fully remove variation information.

Make this product a non-variation parent.

`Product.cross_sell_1`

`Product.cross_sell_2`

`Product.delete (using=None)`

`Product.description`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Product.discountfromproduct_set`

`Product.freeproductline_set`

`Product.get_all_available_combinations ()`

Generate all available combinations of variation variables.

If the product is not a variable variation parent, the iterator is empty.

Because of possible combinatorial explosion this is a generator function. (For example 6 variables with 5 options each explodes to 15,625 combinations.)

Returns Iterable of combination information dicts.

Return type Iterable[dict]

`Product.get_all_package_children ()`

`Product.get_all_package_parents ()`

`Product.get_available_attribute_queryset ()`

`Product.get_available_variation_results ()`

Get a dict of `combination_hash` to product ID of variable variation results.

Returns Mapping of combination hashes to product IDs

Return type dict[str, int]

`Product.get_base_price (context, quantity=1)`

Get base price of the product within given context.

Base price differs from the (effective) price when there are discounts in effect.

Return type *shoop.core.pricing.Price*

`Product.get_cheapest_child_price (context, quantity=1)`

`Product.get_cheapest_child_price_info (context, quantity=1)`

Get the PriceInfo of the cheapest variation child

The attribute used for sorting is PriceInfo.price.

Return `None` if `self.variation_children` do not exist. This is because we cannot return anything sensible.

Return type *shoop.core.pricing.PriceInfo*

`Product.get_child_price_range (context, quantity=1)`

Get the prices for cheapest and the most expensive child

The attribute used for sorting is PriceInfo.price.

Return `(None, None)` if `self.variation_children` do not exist. This is because we cannot return anything sensible.

Returns a tuple of prices

Return type (*shoop.core.pricing.Price*, *shoop.core.pricing.Price*)

`Product.get_mode_display (*moreargs, **morekwargs)`

`Product.get_next_by_created_on(*moreargs, **morekwargs)`

`Product.get_next_by_modified_on(*moreargs, **morekwargs)`

`Product.get_package_child_to_quantity_map()`

`Product.get_previous_by_created_on(*moreargs, **morekwargs)`

`Product.get_previous_by_modified_on(*moreargs, **morekwargs)`

`Product.get_price(context, quantity=1)`

Get price of the product within given context.

Note: When the current pricing module implements pricing steps, it is possible that `p.get_price(ctx) * 123` is not equal to `p.get_price(ctx, quantity=123)`, since there could be quantity discounts in effect, but usually they are equal.

Return type *shoop.core.pricing.Price*

`Product.get_price_info(context, quantity=1)`

Get PriceInfo object for the product in given context.

Returned PriceInfo object contains calculated price and base_price. The calculation of prices is handled in the current pricing module.

Return type *shoop.core.pricing.PriceInfo*

`Product.get_priced_children(context, quantity=1)`

Get child products with price infos sorted by price.

Return type `list[(Product, PriceInfo)]`

Returns List of products and their price infos sorted from cheapest to most expensive.

`Product.get_public_media()`

`Product.get_shipping_mode_display(*moreargs, **morekwargs)`

`Product.get_shop_instance(shop)`

Return type *shoop.core.models.ShopProduct*

`Product.get_stock_behavior_display(*moreargs, **morekwargs)`

`Product.get_variation_siblings()`

`Product.is_package_child()`

`Product.is_package_parent()`

`Product.is_stocked()`

`Product.is_variation_child()`

`Product.is_variation_parent()`

`Product.keywords`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Product.link_to_parent(parent, variables=None, combination_hash=None)`

Parameters

- **parent** (`Product`) – The parent to link to.
- **variables** (`dict/None`) – Optional dict of {variable identifier: value identifier} for complex variable linkage
- **combination_hash** (`str/None`) – Optional combination hash (for variable variations), if precomputed. Mutually exclusive with `variables`

`Product.log_entries`

`Product.make_package` (`package_def`)

`Product.manufacturer`

`Product.media`

`Product.mode`

A placeholder class that provides a way to set the attribute on the model.

`Product.name`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Product.objects` = <django.db.models.manager.ManagerFromProductQuerySet object>

`Product.order_lines`

`Product.primary_image`

`Product.productfilter_set`

`Product.productsinbasketcondition_set`

`Product.sales_unit`

`Product.save` (`*args, **kwargs`)

`Product.shipments`

`Product.shipping_mode`

A placeholder class that provides a way to set the attribute on the model.

`Product.shop_products`

`Product.simplify_variation` ()

Remove variation variables from the given variation parent, turning it into a simple variation (or a normal product, if it has no children).

Parameters `product` (`shoop.core.models.Product`) – Variation parent to not be variable any longer.

`Product.slug`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Product.soft_delete` (`user=None`)

`Product.status_text`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Product.stock_behavior`

A placeholder class that provides a way to set the attribute on the model.

`Product.storedbasket_set`

```
Product.suppliedproduct_set
Product.tax_class
Product.translations
Product.type
Product.unlink_from_parent()
Product.variation_children
Product.variation_name
    Descriptor for translated attributes.
    This attribute proxies all get/set calls to the translated model.
Product.variation_parent
Product.variation_result_subs
Product.variation_result_supers
Product.variation_variables
Product.verify_mode()
class shoop.core.models.ProductAttribute(id, attribute, numeric_value, datetime_value, un-
                                         translated_string_value, product)
    Bases: shoop.core.models._attributes.AppliedAttribute
exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist
exception ProductAttribute.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned
ProductAttribute.attribute
ProductAttribute.objects = <parler.managers.TranslatableManager object>
ProductAttribute.product
ProductAttribute.translated_string_value
    Descriptor for translated attributes.
    This attribute proxies all get/set calls to the translated model.
ProductAttribute.translations
class shoop.core.models.ProductCrossSell(id, product1, product2, weight, type)
    Bases: django.db.models.base.Model
exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist
exception ProductCrossSell.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned
ProductCrossSell.get_type_display(*moreargs, **morekwargs)
ProductCrossSell.objects = <django.db.models.manager.Manager object>
ProductCrossSell.product1
ProductCrossSell.product2
ProductCrossSell.type
    A placeholder class that provides a way to set the attribute on the model.
```

```

class shoop.core.models.ProductCrossSellType
    Bases: enumfields.enums.Enum

    An enumeration.

    BOUGHT_WITH = <ProductCrossSellType.BUGHT_WITH: 4>
    COMPUTED = <ProductCrossSellType.COMPUTED: 3>
    RECOMMENDED = <ProductCrossSellType.RECOMMENDED: 1>
    RELATED = <ProductCrossSellType.RELATED: 2>

class shoop.core.models.ProductMedia (id, identifier, product, kind, file, external_url, ordering, enabled, public, purchased)
    Bases: parler.models.TranslatableModel

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception ProductMedia.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    ProductMedia.description
        Descriptor for translated attributes.

        This attribute proxies all get/set calls to the translated model.

    ProductMedia.easy_thumbnails_thumbnailer
        Get Thumbnailer instance.

        Will return None if file cannot be thumbnailed.

        :rtype:easy_thumbnails.files.Thumbnailer|None

    ProductMedia.effective_title

    ProductMedia.file

    ProductMedia.get_kind_display (*moreargs, **morekwargs)

    ProductMedia.get_thumbnail (**kwargs)
        Get thumbnail for image

        This will return None if there is no file or kind is not ProductMediaKind.IMAGE

        Return type easy_thumbnails.files.ThumbnailFile|None

    ProductMedia.kind
        A placeholder class that provides a way to set the attribute on the model.

    ProductMedia.objects = <parler.managers.TranslatableManager object>

    ProductMedia.primary_image_for_products

    ProductMedia.primary_image_for_shop_products

    ProductMedia.product

    ProductMedia.shops

    ProductMedia.title
        Descriptor for translated attributes.

        This attribute proxies all get/set calls to the translated model.

    ProductMedia.translations

    ProductMedia.url

```

```
class shoop.core.models.ProductMediaKind
    Bases: enumfields.enums.Enum

    An enumeration.

    DOCUMENTATION = <ProductMediaKind.DOCUMENTATION: 3>
    GENERIC_FILE = <ProductMediaKind.GENERIC_FILE: 1>
    IMAGE = <ProductMediaKind.IMAGE: 2>
    SAMPLE = <ProductMediaKind.SAMPLE: 4>

class shoop.core.models.ProductMode
    Bases: enumfields.enums.Enum

    An enumeration.

    NORMAL = <ProductMode.NORMAL: 0>
    PACKAGE_PARENT = <ProductMode.PACKAGE_PARENT: 1>
    SIMPLE_VARIATION_PARENT = <ProductMode.SIMPLE_VARIATION_PARENT: 2>
    VARIABLE_VARIATION_PARENT = <ProductMode.VARIABLE_VARIATION_PARENT: 3>
    VARIATION_CHILD = <ProductMode.VARIATION_CHILD: 4>

class shoop.core.models.ProductPackageLink(id, parent, child, quantity)
    Bases: django.db.models.base.Model

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception ProductPackageLink.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    ProductPackageLink.child

    ProductPackageLink.objects = <django.db.models.manager.Manager object>

    ProductPackageLink.parent

class shoop.core.models.ProductType(id, identifier)
    Bases: parler.models.TranslatableModel

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception ProductType.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    ProductType.attributes

    ProductType.name
        Descriptor for translated attributes.

        This attribute proxies all get/set calls to the translated model.

    ProductType.objects = <parler.managers.TranslatableManager object>

    ProductType.products

    ProductType.producttypefilter_set

    ProductType.translations
```



```

class shoop.core.models.ProductVariationLinkStatus
    Bases: enumfields.enums.Enum

    An enumeration.

    INVISIBLE = <ProductVariationLinkStatus.INVISIBLE: 0>
    VISIBLE = <ProductVariationLinkStatus.VISIBLE: 1>

class shoop.core.models.ProductVariationResult (id, product, combination_hash, result, status)
    Bases: django.db.models.base.Model

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception ProductVariationResult.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    ProductVariationResult.get_status_display (*moreargs, **morekwargs)

    ProductVariationResult.objects = <django.db.models.manager.Manager object>

    ProductVariationResult.product

    classmethod ProductVariationResult.resolve (parent_product, combination)

    ProductVariationResult.result

    ProductVariationResult.status
        A placeholder class that provides a way to set the attribute on the model.

class shoop.core.models.ProductVariationVariable (id, product, identifier)
    Bases: parler.models.TranslatableModel

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception ProductVariationVariable.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    ProductVariationVariable.name
        Descriptor for translated attributes.

        This attribute proxies all get/set calls to the translated model.

    ProductVariationVariable.objects = <parler.managers.TranslatableManager object>

    ProductVariationVariable.product

    ProductVariationVariable.translations

    ProductVariationVariable.values

class shoop.core.models.ProductVariationVariableValue (id, variable, identifier)
    Bases: parler.models.TranslatableModel

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception ProductVariationVariableValue.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    ProductVariationVariableValue.objects = <parler.managers.TranslatableManager object>

    ProductVariationVariableValue.translations

```

`ProductVariationVariableValue.value`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`ProductVariationVariableValue.variable`

class `shoop.core.models.ProductVisibility`

Bases: `enumfields.enums.Enum`

An enumeration.

VISIBLE_TO_ALL = `<ProductVisibility.VISIBLE_TO_ALL: 1>`

VISIBLE_TO_GROUPS = `<ProductVisibility.VISIBLE_TO_GROUPS: 3>`

VISIBLE_TO_LOGGED_IN = `<ProductVisibility.VISIBLE_TO_LOGGED_IN: 2>`

class `shoop.core.models.RoundingMode`

Bases: `enumfields.enums.Enum`

An enumeration.

ROUND_DOWN = `<RoundingMode.ROUND_DOWN: 'ROUND_DOWN'>`

ROUND_HALF_DOWN = `<RoundingMode.ROUND_HALF_DOWN: 'ROUND_HALF_DOWN'>`

ROUND_HALF_UP = `<RoundingMode.ROUND_HALF_UP: 'ROUND_HALF_UP'>`

ROUND_UP = `<RoundingMode.ROUND_UP: 'ROUND_UP'>`

class `shoop.core.models.RoundingBehaviorComponent` (*id, polymorphic_ctype, servicebehaviorcomponent_ptr, quant, mode*)

Bases: `shoop.core.models.ServiceBehaviorComponent`

exception `DoesNotExist`

Bases: `shoop.core.models._service_base.DoesNotExist`

exception `RoundingBehaviorComponent.MultipleObjectsReturned`

Bases: `shoop.core.models._service_base.MultipleObjectsReturned`

`RoundingBehaviorComponent.base_objects` = `<django.db.models.manager.Manager object>`

`RoundingBehaviorComponent.get_costs` (*service, source*)

`RoundingBehaviorComponent.get_mode_display` (**moreargs, **morekwargs*)

`RoundingBehaviorComponent.help_text` = `<django.utils.functional.lazy.<locals>.__proxy__ object>`

`RoundingBehaviorComponent.mode`

A placeholder class that provides a way to set the attribute on the model.

`RoundingBehaviorComponent.name` = `<django.utils.functional.lazy.<locals>.__proxy__ object>`

`RoundingBehaviorComponent.objects` = `<polymorphic.managers.PolymorphicManager object>`

`RoundingBehaviorComponent.polymorphic_primary_key_name` = `'id'`

`RoundingBehaviorComponent.polymorphic_super_sub_accessors_replaced` = `False`

`RoundingBehaviorComponent.servicebehaviorcomponent_ptr`

class `shoop.core.models.SalesUnit` (*id, identifier, decimals*)

Bases: `parler.models.TranslatableModel`

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `SalesUnit.MultipleObjectsReturned`
 Bases: `django.core.exceptions.MultipleObjectsReturned`

`SalesUnit.allow_fractions`

`SalesUnit.name`
 Descriptor for translated attributes.
 This attribute proxies all get/set calls to the translated model.

`SalesUnit.objects = <parler.managers.TranslatableManager object>`

`SalesUnit.product_set`

`SalesUnit.quantity_step`
 Get the quantity increment for the amount of decimals this unit allows.
 For 0 decimals, this will be 1; for 1 decimal, 0.1; etc.
Returns Decimal in (0..1]
Return type *Decimal*

`SalesUnit.round(value)`

`SalesUnit.short_name`
 Descriptor for translated attributes.
 This attribute proxies all get/set calls to the translated model.

`SalesUnit.translations`

class `shoop.core.models.SavedAddress(*args, **kwargs)`
 Bases: `shoop.core.models.ShoopModel`
 Model for saving multiple addresses in an ‘address book’ of sorts.

exception `DoesNotExist`
 Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `SavedAddress.MultipleObjectsReturned`
 Bases: `django.core.exceptions.MultipleObjectsReturned`

`SavedAddress.address`

`SavedAddress.get_role_display(*moreargs, **morekwargs)`

`SavedAddress.get_status_display(*moreargs, **morekwargs)`

`SavedAddress.get_title()`
 Returns the display title for this *SavedAddress* instance. Defaults to a short representation of the address.
 This method should be used instead of accessing the `title` field directly when displaying *SavedAddress* objects.

`SavedAddress.objects = <shoop.core.models._addresses.SavedAddressManager object>`

`SavedAddress.owner`

`SavedAddress.role`
 A placeholder class that provides a way to set the attribute on the model.

`SavedAddress.status`
 A placeholder class that provides a way to set the attribute on the model.

```
class shoop.core.models.SavedAddressRole
```

```
    Bases: enumfields.enums.Enum
```

An enumeration.

```
    BILLING = <SavedAddressRole.BILLING: 2>
```

```
    SHIPPING = <SavedAddressRole.SHIPPING: 1>
```

```
class shoop.core.models.SavedAddressStatus
```

```
    Bases: enumfields.enums.Enum
```

An enumeration.

```
    DISABLED = <SavedAddressStatus.DISABLED: 0>
```

```
    ENABLED = <SavedAddressStatus.ENABLED: 1>
```

```
class shoop.core.models.Service(*args, **kwargs)
```

```
    Bases: shoop.core.models.TranslatableShoopModel
```

Abstract base model for services.

Each enabled service should be linked to a service provider and should have a choice identifier specified in its `choice_identifier` field. The choice identifier should be valid for the service provider, i.e. it should be one of the `ServiceChoice.identifier` values returned by the `ServiceProvider.get_service_choices` method.

```
class Meta
```

```
    Bases: object
```

```
    abstract = False
```

```
Service.behavior_components
```

```
Service.description
```

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

```
Service.get_costs(source)
```

Get costs of this service for items in given source.

Returns description, price and tax class of the costs

Return type Iterable[ServiceCost]

```
Service.get_effective_name(source)
```

Get effective name of the service for given order source.

By default, effective name is the same as name of this service, but if there is a service provider with a custom implementation for `get_effective_name` method, then this can be different.

Return type str

```
Service.get_lines(source)
```

Get lines for given source.

Lines are created based on costs. Costs without description are combined to single line.

Return type Iterable[shoop.core.order_creator.SourceLine]

```
Service.get_total_cost(source)
```

Get total cost of this service for items in given source.

Return type PriceInfo

```

Service.get_unavailability_reasons (source)
    Get reasons of being unavailable for given source.

    Return type Iterable[ValidationError]

Service.is_available_for (source)
    Return true if service is available for given source.

    Return type bool

Service.logo

Service.name
    Descriptor for translated attributes.

    This attribute proxies all get/set calls to the translated model.

Service.objects

Service.old_module_data
    A placeholder class that provides a way to set the attribute on the model.

Service.provider

    Return type shoop.core.models.ServiceProvider

Service.shop

Service.tax_class

class shoop.core.models.ServiceBehaviorComponent (id, polymorphic_ctype)
    Bases: shoop.core.models.PolymorphicShoopModel

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception ServiceBehaviorComponent.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    ServiceBehaviorComponent.base_objects = <django.db.models.manager.Manager object>

    ServiceBehaviorComponent.expensiveswedenbehaviorcomponent

    ServiceBehaviorComponent.fixedcostbehaviorcomponent

    ServiceBehaviorComponent.get_costs (service, source)
        Return costs for for this object. This should be implemented in subclass. This method is used to calculate
        price for ShippingMethod and PaymentMethod objects.

        Return type Iterable[ServiceCost]

    ServiceBehaviorComponent.get_delivery_time (service, source)

        Return type shoop.utils.dates.DurationRangelNone

    ServiceBehaviorComponent.get_unavailability_reasons (service, source)

        Return type Iterable[ValidationError]

    ServiceBehaviorComponent.groupavailabilitybehaviorcomponent

    ServiceBehaviorComponent.help_text = None

    ServiceBehaviorComponent.name = None

    ServiceBehaviorComponent.objects = <polymorphic.managers.PolymorphicManager object>

    ServiceBehaviorComponent.paymentmethod_set

```

```
ServiceBehaviorComponent.polymorphic_ctype
ServiceBehaviorComponent.polymorphic_primary_key_name = 'id'
ServiceBehaviorComponent.polymorphic_super_sub_accessors_replaced = False
ServiceBehaviorComponent.roundingbehaviorcomponent
ServiceBehaviorComponent.shippingmethod_set
ServiceBehaviorComponent.staffonlybehaviorcomponent
ServiceBehaviorComponent.waivingcostbehaviorcomponent
ServiceBehaviorComponent.weightbasedpricingbehaviorcomponent
ServiceBehaviorComponent.weightlimitsbehaviorcomponent
```

```
class shoop.core.models.ServiceChoice (identifier, name)
```

Bases: `object`

Choice of service provided by a service provider.

Initialize service choice.

Parameters

- **identifier** (*str*) – Internal identifier for the service. Should be unique within a single *ServiceProvider*.
- **name** (*str*) – Descriptive name of the service in currently active language.

```
class shoop.core.models.ServiceCost (price, description=None, tax_class=None,
                                     base_price=None)
```

Bases: `object`

A cost of a service.

One service might have several costs.

Initialize cost from values.

Note: If `tax_class` is specified, also `description` must be given.

price_info

```
class shoop.core.models.ServiceProvider (*args, **kwargs)
```

Bases: `shoop.core.models.PolymorphicTranslatableShoopModel`

Entity that provides services.

Good examples of service providers are *Carrier* and *PaymentProcessor*.

When subclassing *ServiceProvider*, set value for *service_model* class attribute. It should be a model class which is subclass of *Service*.

Replace Django's inheritance accessor member functions for our model (`self.__class__`) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate `base_objects` manager.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception ServiceProvider.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`ServiceProvider.base_objects = <parler.managers.TranslatableManager object>`

`ServiceProvider.base_translations`

`ServiceProvider.carrier`

`ServiceProvider.create_service(choice_identifier, **kwargs)`

Create a service for given choice identifier.

Subclass implementation may attach some *behavior components* to the created service.

Subclasses should provide implementation for `_create_service` or override this. Base class implementation calls the `_create_service` method with resolved `choice_identifier`.

Parameters `choice_identifier` (*str/None*) – Identifier of the service choice to use. If

None, use the default service choice.

Return type *shoop.core.models.Service*

`ServiceProvider.get_effective_name(service, source)`

Get effective name of the service for given order source.

Base class implementation will just return name of the given service, but that may be changed in a subclass.

Return type *str*

`ServiceProvider.get_service_choices()`

Get all service choices of this provider.

Subclasses should implement this method.

Return type `list[ServiceChoice]`

`ServiceProvider.logo`

`ServiceProvider.name`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`ServiceProvider.objects = <shoop.core.models._base._PolyTransManager object>`

`ServiceProvider.paymentprocessor`

`ServiceProvider.polymorphic_ctype`

`ServiceProvider.polymorphic_primary_key_name = 'id'`

`ServiceProvider.polymorphic_super_sub_accessors_replaced = False`

`ServiceProvider.service_model = None`

class `shoop.core.models.Shipment` (*id, order, supplier, created_on, status, tracking_code, description, volume, weight, identifier*)

Bases: `django.db.models.base.Model`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

```
exception Shipment.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

Shipment.cache_values()
    (Re)cache volume and weight for this Shipment from the ShipmentProducts within.

Shipment.get_next_by_created_on(*moreargs, **morekwargs)

Shipment.get_previous_by_created_on(*moreargs, **morekwargs)

Shipment.get_status_display(*moreargs, **morekwargs)

Shipment.objects = <django.db.models.manager.Manager object>

Shipment.order

Shipment.products

Shipment.save(*args, **kwargs)

Shipment.status
    A placeholder class that provides a way to set the attribute on the model.

Shipment.supplier

Shipment.total_products

class shoop.core.models.ShipmentProduct(id, shipment, product, quantity, unit_volume,
                                         unit_weight)
    Bases: django.db.models.base.Model

exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception ShipmentProduct.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

ShipmentProduct.cache_values()

ShipmentProduct.objects = <django.db.models.manager.Manager object>

ShipmentProduct.product

ShipmentProduct.shipment

class shoop.core.models.ShippingMethod(id, identifier, enabled, shop, choice_identifier,
                                         old_module_identifier, old_module_data, logo,
                                         tax_class, carrier)
    Bases: shoop.core.models.Service

exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception ShippingMethod.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

ShippingMethod.behavior_components

ShippingMethod.can_delete()

ShippingMethod.carrier

ShippingMethod.contact_set

ShippingMethod.description
    Descriptor for translated attributes.

    This attribute proxies all get/set calls to the translated model.
```



```

ShippingMethod.get_shipping_time (source)
    Get shipping time for items in given source.

    Return type shoop.utils.dates.DurationRange|None

ShippingMethod.line_type = <OrderLineType.SHIPPING: 2>

ShippingMethod.logo

ShippingMethod.name
    Descriptor for translated attributes.

    This attribute proxies all get/set calls to the translated model.

ShippingMethod.objects = <django.db.models.manager.ManagerFromServiceQuerySet object>

ShippingMethod.old_module_data
    A placeholder class that provides a way to set the attribute on the model.

ShippingMethod.provider_attr = 'carrier'

ShippingMethod.shipping_orders

ShippingMethod.shipping_products

ShippingMethod.shop

ShippingMethod.shop_product_m2m = 'shipping_methods'

ShippingMethod.tax_class

ShippingMethod.translations

class shoop.core.models.ShippingMode
    Bases: enumfields.enums.Enum

    An enumeration.

    NOT_SHIPPED = <ShippingMode.NOT_SHIPPED: 0>

    SHIPPED = <ShippingMode.SHIPPED: 1>

class shoop.core.models.ShippingStatus
    Bases: enumfields.enums.Enum

    An enumeration.

    FULLY_SHIPPED = <ShippingStatus.FULLY_SHIPPED: 2>

    NOT_SHIPPED = <ShippingStatus.NOT_SHIPPED: 0>

    PARTIALLY_SHIPPED = <ShippingStatus.PARTIALLY_SHIPPED: 1>

class shoop.core.models.ShoopModel (*args, **kwargs)
    Bases: django.db.models.base.Model

    Shoop Model.

    class Meta
        Bases: object

        abstract = False

    ShoopModel.identifier_attr = 'identifier'

class shoop.core.models.Shop (id, identifier, domain, status, owner, options, currency,
                             prices_include_tax, logo, maintenance_mode, contact_address)
    Bases: shoop.core.models._base.ChangeProtected, shoop.core.models.TranslatableShoopModel

```

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception Shop.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`Shop.basketcampaign_set`

`Shop.catalogcampaign_set`

`Shop.categories`

`Shop.cgpprice_set`

`Shop.change_protect_message` = `<django.utils.functional.lazy.<locals>.__proxy__ object>`

`Shop.contact_address`

`Shop.create_price` (*value*)

Create a price with given value and settings of this shop.

Takes the `prices_include_tax` and `currency` settings of this Shop into account.

Return type *shoop.core.pricing.Price*

`Shop.get_status_display` (**moreargs*, ***morekwargs*)

`Shop.logo`

`Shop.maintenance_message`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Shop.name`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Shop.objects` = `<parler.managers.TranslatableManager object>`

`Shop.options`

A placeholder class that provides a way to set the attribute on the model.

`Shop.order_set`

`Shop.owner`

`Shop.paymentmethod_set`

`Shop.product_media`

`Shop.protected_fields` = `['currency', 'prices_include_tax']`

`Shop.public_name`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Shop.shippingmethod_set`

`Shop.shop_products`

`Shop.status`

A placeholder class that provides a way to set the attribute on the model.

`Shop.storedbasket_set`

`Shop.translations`

```
class shoop.core.models.ShopProduct(id, shop, product, visible, listed, purchasable,
                                     searchable, visibility_limit, purchase_multiple, min-
                                     imum_purchase_quantity, limit_shipping_methods,
                                     limit_payment_methods, primary_category,
                                     shop_primary_image, default_price_value, mini-
                                     mum_price_value)
```

Bases: `shoop.utils.properties.MoneyPropped`, `django.db.models.base.Model`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception ShopProduct.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`ShopProduct.categories`

`ShopProduct.default_price`

Property for Price object.

Similar to `MoneyProperty` but also has `includes_tax` field.

Operators with `TaxfulPrice` and `TaxlessPrice` objects.

`ShopProduct.get_orderability_errors(supplier, quantity, customer, ig-
nore_minimum=False)`

Yield `ValidationErrors` that would cause this product to not be orderable.

Parameters

- **supplier** (`shoop.core.models.Supplier`) – Supplier to order this product from. May be None.
- **quantity** (`int/Decimal`) – Quantity to order.
- **customer** (`shoop.core.models.Contact`) – Customer contact.
- **ignore_minimum** (`bool`) – Ignore any limitations caused by quantity minimums.

Returns `Iterable[ValidationError]`

`ShopProduct.get_visibility_errors(customer)`

`ShopProduct.get_visibility_limit_display(*moreargs, **morekwargs)`

`ShopProduct.images`

`ShopProduct.is_list_visible()`

Return True if this product should be visible in listings in general, without taking into account any other visibility limitations. :rtype: bool

`ShopProduct.is_orderable(supplier, customer, quantity)`

`ShopProduct.is_visible(customer)`

`ShopProduct.minimum_price`

Property for Price object.

Similar to `MoneyProperty` but also has `includes_tax` field.

Operators with `TaxfulPrice` and `TaxlessPrice` objects.

`ShopProduct.objects = <django.db.models.manager.Manager object>`

`ShopProduct.payment_methods`

`ShopProduct.primary_category`

`ShopProduct.primary_image`

`ShopProduct.product`

`ShopProduct.quantity_step`

Quantity step for purchasing this product.

Return type `decimal.Decimal`

Example: `<input type="number" step="{{ shop_product.quantity_step }}">`

`ShopProduct.raise_if_not_orderable (supplier, customer, quantity, ignore_minimum=False)`

`ShopProduct.raise_if_not_visible (customer)`

`ShopProduct.rounded_minimum_purchase_quantity`

The minimum purchase quantity, rounded to the sales unit's precision.

Return type `decimal.Decimal`

Example:

`<input type="number" min="{{ shop_product.rounded_minimum_purchase_quantity }}" value="{{ shop_product.rounded_minimum_purchase_quantity }}">`

`ShopProduct.save (*args, **kwargs)`

`ShopProduct.shipping_methods`

`ShopProduct.shop`

`ShopProduct.shop_primary_image`

`ShopProduct.suppliers`

`ShopProduct.visibility_groups`

`ShopProduct.visibility_limit`

A placeholder class that provides a way to set the attribute on the model.

class `shoop.core.models.ShopStatus`

Bases: `enumfields.enums.Enum`

An enumeration.

DISABLED = `<ShopStatus.DISABLED: 0>`

ENABLED = `<ShopStatus.ENABLED: 1>`

class `shoop.core.models.StaffOnlyBehaviorComponent (id, polymorphic_ctype, servicebehaviorcomponent_ptr)`

Bases: `shoop.core.models.ServiceBehaviorComponent`

exception `DoesNotExist`

Bases: `shoop.core.models._service_base.DoesNotExist`

exception `StaffOnlyBehaviorComponent.MultipleObjectsReturned`

Bases: `shoop.core.models._service_base.MultipleObjectsReturned`

`StaffOnlyBehaviorComponent.base_objects` = `<django.db.models.manager.Manager object>`

`StaffOnlyBehaviorComponent.get_unavailability_reasons (service, source)`

`StaffOnlyBehaviorComponent.help_text` = `<django.utils.functional.lazy.<locals>.__proxy__ object>`

```

StaffOnlyBehaviorComponent.name = <django.utils.functional.lazy.<locals>.__proxy__ object>
StaffOnlyBehaviorComponent.objects = <polymorphic.managers.PolymorphicManager object>
StaffOnlyBehaviorComponent.polymorphic_primary_key_name = 'id'
StaffOnlyBehaviorComponent.polymorphic_super_sub_accessors_replaced = False
StaffOnlyBehaviorComponent.servicebehaviorcomponent_ptr

class shoop.core.models.StockBehavior
    Bases: enumfields.enums.Enum

    An enumeration.

    STOCKED = <StockBehavior.STOCKED: 1>
    UNSTOCKED = <StockBehavior.UNSTOCKED: 0>

class shoop.core.models.SuppliedProduct (id, supplier, product, sku, alert_limit, physical_count,
                                         logical_count)
    Bases: django.db.models.base.Model

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception SuppliedProduct.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    SuppliedProduct.objects = <django.db.models.manager.Manager object>

    SuppliedProduct.product
    SuppliedProduct.supplier

class shoop.core.models.Supplier (id, identifier, name, type, stock_managed, module_identifier,
                                  module_data)
    Bases: shoop.core.modules.interface.ModuleInterface, shoop.core.models.ShoopModel

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception Supplier.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    Supplier.adjust_stock (product_id, delta, created_by=None)

    Supplier.default_module_spec = 'shoop.core.suppliers:BaseSupplierModule'

    Supplier.get_orderability_errors (shop_product, quantity, customer)

        Parameters
        • shop_product (shoop.core.models.ShopProduct) – Shop Product
        • quantity (decimal.Decimal) – Quantity to order
        • contact (shoop.core.models.Contact) – Ordering contact.

        Return type iterable[ValidationError]

    Supplier.get_stock_status (product_id)

        Parameters product_id (int) – Product ID

        Return type shoop.core.stocks.ProductStockStatus

    Supplier.get_stock_statuses (product_ids)

```

Parameters `product_ids` – Iterable of product IDs

Returns Dict of {product_id: ProductStockStatus}

Return type dict[int, shoop.core.stocks.ProductStockStatus]

`Supplier.get_suppliable_products (shop, customer)`

Parameters

- `shop` (`shoop.core.models.Shop`) – Shop to check for suppliability
- `customer` (`shoop.core.models.Contact`) – Customer contact to check for suppliability

Return type list[int]

`Supplier.get_type_display (*moreargs, **morekwargs)`

`Supplier.module_data`

A placeholder class that provides a way to set the attribute on the model.

`Supplier.module_provides_key = 'supplier_module'`

`Supplier.objects = <django.db.models.manager.Manager object>`

`Supplier.order_lines`

`Supplier.shipments`

`Supplier.shop_products`

`Supplier.stockadjustment_set`

`Supplier.stockcount_set`

`Supplier.suppliedproduct_set`

`Supplier.type`

A placeholder class that provides a way to set the attribute on the model.

`Supplier.update_stock (product_id)`

`Supplier.update_stocks (product_ids)`

class `shoop.core.models.SupplierType`

Bases: `enumfields.enums.Enum`

An enumeration.

EXTERNAL = `<SupplierType.EXTERNAL: 2>`

INTERNAL = `<SupplierType.INTERNAL: 1>`

class `shoop.core.models.Tax (id, code, rate, amount_value, currency, enabled)`

Bases: `shoop.utils.properties.MoneyPropped`, `shoop.core.models._base.ChangeProtected`, `shoop.core.models.TranslatableShoopModel`

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `Tax.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`Tax.amount`

Property for a Money amount.

Will return `Money` objects when the property is being get and accepts `Money` objects on set. Value and currency are read/written from/to other fields.

Fields are given as locators, that is a string in dotted format, e.g. locator `"foo.bar"` points to `instance.foo.bar` where `instance` is an instance of the class owning the `MoneyProperty`.

Setting value of this property to a `Money` object with different currency that is currently set (in the field pointed by the currency locator), will raise an `UnitMixupError`.

`Tax.calculate_amount` (*base_amount*)

Calculate tax amount with this tax for given base amount.

Return type *shoop.utils.money.Money*

`Tax.change_protect_message` = <django.utils.functional.lazy.<locals>.__proxy__ object>

`Tax.clean` ()

`Tax.identifier_attr` = 'code'

`Tax.name`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Tax.objects` = <parler.managers.TranslatableManager object>

`Tax.order_line_taxes`

`Tax.taxrule_set`

`Tax.translations`

`Tax.unprotected_fields` = ['enabled']

class `shoop.core.models.TaxClass` (*id, identifier, enabled*)

Bases: *shoop.core.models.TranslatableShoopModel*

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `TaxClass.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`TaxClass.name`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`TaxClass.objects` = <parler.managers.TranslatableManager object>

`TaxClass.paymentmethod_set`

`TaxClass.product_set`

`TaxClass.shippingmethod_set`

`TaxClass.taxrule_set`

`TaxClass.translations`

class `shoop.core.models.TranslatableShoopModel` (**args, **kwargs*)

Bases: *shoop.core.models.ShoopModel*, `parler.models.TranslatableModel`

class `Meta`

Bases: `object`

abstract = False

```
TranslatableShoopModel.name_attr = 'name'
TranslatableShoopModel.objects

class shoop.core.models.WaivingCostBehaviorComponent (id, polymorphic_ctype, servicebehaviorcomponent_ptr, price_value, waive_limit_value)
    Bases: shoop.core.models._service_base.TranslatableServiceBehaviorComponent
    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist
    exception WaivingCostBehaviorComponent.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned
    WaivingCostBehaviorComponent.base_objects = <parler.managers.TranslatableManager object>
    WaivingCostBehaviorComponent.description
        Descriptor for translated attributes.
        This attribute proxies all get/set calls to the translated model.
    WaivingCostBehaviorComponent.get_costs (service, source)
    WaivingCostBehaviorComponent.help_text = <django.utils.functional.lazy.<locals>.__proxy__ object>
    WaivingCostBehaviorComponent.name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    WaivingCostBehaviorComponent.objects = <shoop.core.models._base.PolyTransManager object>
    WaivingCostBehaviorComponent.polymorphic_primary_key_name = 'id'
    WaivingCostBehaviorComponent.polymorphic_super_sub_accessors_replaced = False
    WaivingCostBehaviorComponent.servicebehaviorcomponent_ptr
    WaivingCostBehaviorComponent.translations

class shoop.core.models.WeightBasedPriceRange (id, component, min_value, max_value, price_value)
    Bases: parler.models.TranslatableModel
    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist
    exception WeightBasedPriceRange.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned
    WeightBasedPriceRange.component
    WeightBasedPriceRange.description
        Descriptor for translated attributes.
        This attribute proxies all get/set calls to the translated model.
    WeightBasedPriceRange.matches_to_value (value)
    WeightBasedPriceRange.objects = <parler.managers.TranslatableManager object>
    WeightBasedPriceRange.translations

class shoop.core.models.WeightBasedPricingBehaviorComponent (id, polymorphic_ctype, servicebehaviorcomponent_ptr)
    Bases: shoop.core.models.ServiceBehaviorComponent
```


exception DoesNotExist

Bases: `shoop.core.models._service_base.DoesNotExist`

exception WeightBasedPricingBehaviorComponent.MultipleObjectsReturned

Bases: `shoop.core.models._service_base.MultipleObjectsReturned`

`WeightBasedPricingBehaviorComponent.base_objects` = <django.db.models.manager.Manager object>

`WeightBasedPricingBehaviorComponent.get_costs` (*service*, *source*)

`WeightBasedPricingBehaviorComponent.get_unavailability_reasons` (*service*,
source)

`WeightBasedPricingBehaviorComponent.help_text` = <django.utils.functional.lazy.<locals>.__proxy__ object>

`WeightBasedPricingBehaviorComponent.name` = <django.utils.functional.lazy.<locals>.__proxy__ object>

`WeightBasedPricingBehaviorComponent.objects` = <polymorphic.managers.PolymorphicManager object>

`WeightBasedPricingBehaviorComponent.polymorphic_primary_key_name` = 'id'

`WeightBasedPricingBehaviorComponent.polymorphic_super_sub_accessors_replaced` = False

`WeightBasedPricingBehaviorComponent.ranges`

`WeightBasedPricingBehaviorComponent.servicebehaviorcomponent_ptr`

class `shoop.core.models.WeightLimitsBehaviorComponent` (*id*, *polymorphic_ctype*, *servicebehaviorcomponent_ptr*,
min_weight, *max_weight*)

Bases: `shoop.core.models.ServiceBehaviorComponent`

exception DoesNotExist

Bases: `shoop.core.models._service_base.DoesNotExist`

exception WeightLimitsBehaviorComponent.MultipleObjectsReturned

Bases: `shoop.core.models._service_base.MultipleObjectsReturned`

`WeightLimitsBehaviorComponent.base_objects` = <django.db.models.manager.Manager object>

`WeightLimitsBehaviorComponent.get_unavailability_reasons` (*service*, *source*)

`WeightLimitsBehaviorComponent.help_text` = <django.utils.functional.lazy.<locals>.__proxy__ object>

`WeightLimitsBehaviorComponent.name` = <django.utils.functional.lazy.<locals>.__proxy__ object>

`WeightLimitsBehaviorComponent.objects` = <polymorphic.managers.PolymorphicManager object>

`WeightLimitsBehaviorComponent.polymorphic_primary_key_name` = 'id'

`WeightLimitsBehaviorComponent.polymorphic_super_sub_accessors_replaced` = False

`WeightLimitsBehaviorComponent.servicebehaviorcomponent_ptr`

shoop.core.modules package**Submodules****shoop.core.modules.interface module****exception** `shoop.core.modules.interface.ModuleNotFound`

Bases: `ValueError`

class `shoop.core.modules.interface.ModuleInterface`

Bases: `object`

```
default_module_spec = None
module_identifier = None
module_options_field = 'module_data'
module_provides_key = None
module
classmethod get_module_choices (empty_label=None)
classmethod get_module_implementation_map ()
    Get a dict that maps module spec identifiers (short strings) into actual spec names.
    As an example:
```

```
{ "Eggs": "foo_package.bar_module:EggsClass" }
```

Return type dict[str, str]

Module contents

```
class shoop.core.modules.ModuleInterface
    Bases: object
    default_module_spec = None
    classmethod get_module_choices (empty_label=None)
    classmethod get_module_implementation_map ()
        Get a dict that maps module spec identifiers (short strings) into actual spec names.
        As an example:
```

```
{ "Eggs": "foo_package.bar_module:EggsClass" }
```

Return type dict[str, str]

```
module
module_identifier = None
module_options_field = 'module_data'
module_provides_key = None
```

shoop.core.order_creator package

Submodules

shoop.core.order_creator.signals module

Module contents

```
shoop.core.order_creator.get_order_source_modifier_modules ()
    Get a list of configured order source modifier module instances.
```

Return type list[OrderSourceModifierModule]

```
shoop.core.order_creator.is_code_usable (order_source, code)
```

class `shoop.core.order_creator.OrderCreator` (*request=None*)
 Bases: `shoop.core.order_creator._creator.OrderProcessor`

Initialize order creator.

Parameters **request** (*django.http.HttpRequest/None*) – Optional request object for backward compatibility. Passing non-None value is DEPRECATED.

create_order (*order_source*)

class `shoop.core.order_creator.OrderModifier`
 Bases: `shoop.core.order_creator._creator.OrderProcessor`

update_order_from_source (*order_source, order*)

class `shoop.core.order_creator.OrderSource` (*shop*)
 Bases: `object`

A “provisional order” object.

Contains data that’s not strictly about a basket’s contents, but is useful for things that need to calculate something based on the basket’s contents and extra data, such as shipping/billing addresses.

The core API of `OrderCreator` reads an `OrderSource`.

No objects held here need be saved, but they may be.

add_code (*code*)
 Add code to this OrderSource.

At this point it is expected that the customers permission to use the code has already been checked by the caller.

The code will be converted to text.

Parameters **code** (*str*) – The code to add

Returns True if code was added, False if it was already there

Return type `bool`

add_line (***kwargs*)

calculate_taxes (*force_recalculate=False*)

calculate_taxes_or_raise ()

clear_codes ()
 Remove all codes from this OrderSource.

Returns True iff there was codes before clearing

Return type `bool`

codes

create_line (***kwargs*)

creator

customer

get_final_lines (*with_taxes=False*)
 Get lines with processed lines added.

This implementation includes the all lines returned by `get_lines` and in addition, lines from shipping and payment methods, but these lines can be extended, deleted or replaced by a subclass (by overriding `_compute_processed_lines` method) and with the `post_compute_source_lines` signal.

Note: By default, taxes for the returned lines are not calculated when `self.calculate_taxes_automatically` is false. Pass in True to `with_taxes` argument or use `calculate_taxes` method to force tax calculation.

get_lines()

Get unprocessed lines in this OrderSource.

See also `get_final_lines`.

get_product_lines()

Get lines with a product.

This does not use `get_final_lines` because it will be called when final lines is being computed (for example to determine shipping discounts based on the total price of all products).

get_validation_errors()

has_shippable_lines()

is_empty

modified_by

orderer

payment_method

product_count

Get the total number of products in this OrderSource.

Return type decimal.Decimal

product_ids

remove_code(*code*)

Remove given code from this OrderSource.

Parameters *code* (*str*) – The code to remove

Returns True if code was removed, False if code was not there

Return type bool

shipping_method

status

taxful_total_discount

Property that calculates sum of prices.

Used to implement various total price propties to OrderSource.

taxful_total_discount_or_none

Property that turns TaxesNotCalculated exception to None.

Used to implement the OrderSource taxful/taxless total price properties with the “_or_none” suffix.

taxful_total_price

Property that calculates sum of prices.

Used to implement various total price propties to OrderSource.

taxful_total_price_or_none

Property that turns TaxesNotCalculated exception to None.

Used to implement the OrderSource taxful/taxless total price properties with the “_or_none” suffix.

taxless_total_discount

Property that calculates sum of prices.

Used to implement various total price proprties to OrderSource.

taxless_total_discount_or_none

Property that turns TaxesNotCalculated exception to None.

Used to implement the OrderSource taxful/taxless total price properties with the “_or_none” suffix.

taxless_total_price

Property that calculates sum of prices.

Used to implement various total price proprties to OrderSource.

taxless_total_price_or_none

Property that turns TaxesNotCalculated exception to None.

Used to implement the OrderSource taxful/taxless total price properties with the “_or_none” suffix.

total_discount

Property that calculates sum of prices.

Used to implement various total price proprties to OrderSource.

total_gross_weight**total_price**

Property that calculates sum of prices.

Used to implement various total price proprties to OrderSource.

total_price_of_products

Property that calculates sum of prices.

Used to implement various total price proprties to OrderSource.

uncache()

Uncache processed lines.

Should be called after changing the contents before (re)accessing lines with *get_final_lines*.

update(values)****update_from_order(order)**

```
class shoop.core.order_creator.OrderSourceModifierModule
```

Bases: *object*

can_use_code(order_source, code)

clear_codes(order)

get_new_lines(order_source, lines)

Get new lines to be added to order source.

Return type Iterable[shoop.core.order_creator.SourceLine]

use_code(order, code)

class `shoop.core.order_creator.SourceLine` (*source*, ***kwargs*)

Bases: `shoop.core.taxing.TaxableItem`, `shoop.core.pricing.Priceful`

Line of OrderSource.

Note: Properties like `price`, `taxful_price`, `tax_rate`, etc. are inherited from the `Priceful` mixin.

Initialize `SourceLine` with given source and data.

Parameters

- **source** (`OrderSource`) – The `OrderSource` this `SourceLine` belongs to.
- **kwargs** – Data for the `SourceLine`.

base_unit_price = `None`

discount_amount = `None`

classmethod `from_dict` (*source*, *data*)

Create `SourceLine` from given `OrderSource` and dict.

Return type `cls`

get (*key*, *default=None*)

parent_line

quantity = `None`

tax_amount

Return type `shoop.utils.money.Money`

tax_class

taxes

Taxes of this line.

Determined by a `TaxModule` in `OrderSource.calculate_taxes`.

Return type `list[shoop.core.taxing.LineTax]`

to_dict ()

update (***kwargs*)

exception `shoop.core.order_creator.TaxesNotCalculated`

Bases: `TypeError`

Requested tax calculated price but taxes are not calculated.

Raised when requesting a price with `taxful/taxless` mismatching with `shop.prices_include_tax` and taxes are not yet calculated.

shoop.core.pricing package

Submodules

shoop.core.pricing.default_pricing module

class `shoop.core.pricing.default_pricing.DefaultPricingModule`

Bases: `shoop.core.pricing.PricingModule`

identifier = `'default_pricing'`

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

get_price_info (*context, product, quantity=1*)

Return a `PriceInfo` calculated from `ShopProduct.default_price`

Since `ShopProduct.default_price` can be `None` it will be set to zero (0) if `None`.

Module contents Shoop modular product pricing functionality.

The pricing module in use is declared by the `SHOOP_PRICING_MODULE` setting. The default is a pricing module that always prices everything to be free. The base distribution contains `shoop.customer_group_pricing`, which is an useful pricing module for many cases.

To acquire an instance of the current pricing module, use `get_pricing_module`.

In brief, a pricing module is able to price a product based on a *context*; what exactly a context contains is determined by the module in question. You can construct a context from a request by calling the module's `get_context_from_request` method, or for more advanced uses, when you do not have access to an HTTP request, `get_context_from_data`.

After you have acquired the module and a context, you can calculate prices for a product with the module's `get_price_info` method. (*Product* objects contain the convenience methods `get_price_info`, `get_price`, and `get_base_price` which do these steps for you.)

If you have multiple products, it will likely be more efficient – depending on the implementation of the module – to use the `get_price_infos` method.

TODO: document the concepts of base price and the pricing steps API. TODO: caching.

class `shoop.core.pricing.DiscountModule`

Bases: `object`

discount_price (*context, product, price_info*)

Discount given price of given product.

Parameters

- **context** (`shoop.core.pricing.PricingContext`) – Pricing context to operate in
- **product** (`shoop.core.models.Product` | `int`) – Product in question or its id
- **price_info** (`shoop.core.pricing.PriceInfo`) – Price to discount

Returns Discounted price

Return type `shoop.core.pricing.PriceInfo`

discount_prices (*context, products, price_infos*)

Discount a bunch of prices.

Parameters

- **context** (`shoop.core.pricing.PricingContext`) – Pricing context to operate in
- **products** (`Iterable[shoop.core.models.Product | int]`) – Products in question or their ids

Return type `dict[int, PriceInfo]`

get_pricing_steps (*context, product, steps*)

Get discounted pricing steps for given product.

Base class version just discounts all the given steps with `discount_price`, but another module could add more steps and should do so, if the module introduces any pricing steps.

Parameters

- **context** (`shoop.core.pricing.PricingContext`) – Pricing context to operate in
- **product** (`shoop.core.models.Product | int`) – Product in question or its id

Return type `list[PriceInfo]`

get_pricing_steps_for_products (`context, products, steps`)

Get discounted pricing steps for a bunch of products.

Parameters

- **context** (`shoop.core.pricing.PricingContext`) – Pricing context to operate in
- **products** (`Iterable[shoop.core.models.Product | int]`) – Products in question or their ids

Return type `dict[int, list[PriceInfo]]`

`shoop.core.pricing.get_discount_modules()`

Get a list of configured discount module instances.

Return type `list[DiscountModule]`

`shoop.core.pricing.get_price_info(context, product, quantity=1)`

Get price info of product for given quantity.

Returned `PriceInfo` object contains calculated price and `base_price`. The calculation of prices is handled in the current pricing module and possibly configured discount modules.

Parameters **product** (`shoop.core.models.Product | int`) – Product object or id of Product

Return type `shoop.core.pricing.PriceInfo`

`shoop.core.pricing.get_price_infos(context, products, quantity=1)`

Get `PriceInfo` objects for a bunch of products.

Returns a dict with product id as key and `PriceInfo` as value.

May be faster than doing `get_price_info` for each product.

Parameters **products** (`Iterable[shoop.core.models.Product | int]`) – List of product objects or id's

Return type `dict[int, PriceInfo]`

`shoop.core.pricing.get_pricing_module()`

Return type `shoop.core.pricing.PricingModule`

`shoop.core.pricing.get_pricing_steps(context, product)`

Get context-specific list pricing steps for the given product.

Returns a list of `PriceInfos`, see `PricingModule.get_pricing_steps` for description of its format.

Parameters **product** (`shoop.core.models.Product | int`) – Product or product id

Return type `list[shoop.core.pricing.PriceInfo]`

`shoop.core.pricing.get_pricing_steps_for_products` (*context, products*)

Get pricing steps for a bunch of products.

Returns a dict with product id as key and step data (as list of PriceInfos) as values.

May be faster than doing `get_pricing_steps` for each product separately.

Parameters `products` (*Iterable[shoop.core.models.Product/int]*) – List of product objects or id's

Return type `dict[int,list[PriceInfo]]`

class `shoop.core.pricing.Price`

Bases: `shoop.utils.money.Money`

Money amount with taxful/taxless info.

Taxful and taxless prices cannot be mixed in comparison or in calculations, i.e. operations like `x < y` or `x + y` for two Prices `x` and `y` with `x.includes_tax != y.includes_tax` will raise an `UnitMixupError`.

In addition to `includes_tax` info, Prices are Money and know their `value` and currency. To get the bare Money amount of a `Price`, use the `amount` property.

amount

Money amount of this price.

Return type `Money`

classmethod `from_data` (*value, currency, includes_tax=None*)

includes_tax = `None`

unit_matches_with (*other*)

class `shoop.core.pricing.PriceDisplayOptions` (*include_taxes=None, show_prices=True*)

Bases: `object`

Price display options.

Parameters on how prices should be rendered.

Initialize price display options.

Parameters

- **include_taxes** (*bool/None*) – Whether include taxes to rendered prices or not. If `None`, show prices in their original taxness.
- **show_prices** (*bool*) – Whether show prices at all.

classmethod `from_context` (*context*)

Get price display options from context.

Return type `PriceDisplayOptions`

hide_prices

set_for_request (*request*)

Set price display options of given request to self.

class `shoop.core.pricing.Priceful`

Bases: `object`

Mixin to define price properties based on other price properties.

You must provide at least

- `quantity` (`Decimal`)

and both

- `base_unit_price` (*Price*) and
- `discount_amount` (*Price*)

or both

- `price` (*Price*) and
- `base_price` (*Price*).

You may also provide

- `tax_amount` (*Money*)

to get various tax related properties.

Provided `base_unit_price`, `discount_amount`, `price`, `base_price`, and `tax_amount` must have compatible units (i.e. same taxness and currency).

Invariants:

- $\text{price} = \text{base_unit_price} * \text{quantity} - \text{discount_amount}$
- $\text{discount_amount} = \text{base_price} - \text{price}$
- $\text{discount_rate} = 1 - (\text{price} / \text{base_price})$
- $\text{discount_percentage} = 100 * \text{discount_rate}$
- $\text{unit_discount_amount} = \text{discount_amount} / \text{quantity}$
- $\text{taxful_price} = \text{taxless_price} + \text{tax_amount}$
- $\text{tax_rate} = (\text{taxful_price}.\text{amount} / \text{taxless_price}.\text{amount}) - 1$
- $\text{tax_percentage} = 100 * \text{tax_rate}$

`base_price`

Total price for the specified quantity excluding discount.

Return type *shoop.core.pricing.Price*

`base_unit_price`

Undiscounted unit price.

Note: If quantity is 0, will return `base_price`.

Return type *shoop.core.pricing.Price*

`discount_amount`

Amount of discount for the total quantity.

Normally positive or zero, but might also be negative if product is being sold with higher price than its normal price.

Return type *shoop.core.pricing.Price*

`discount_percentage`

Discount percentage, 100 meaning totally discounted.

See *discount_rate*.

Return type *decimal.Decimal*

discount_rate

Discount rate, 1 meaning totally discounted.

Note: Could be negative, when base price is smaller than effective price. Could also be greater than 1, when effective price is negative.

If base price is 0, will return 0.

Return type `decimal.Decimal`

discounted_unit_price

Unit price with discount.

If quantity is 0, will return `base_unit_price - discount_amount`.

Return type `shoop.core.pricing.Price`

is_discounted

Check if there is a discount in effect.

Returns True, iff price < base price.

price

Total price for the specified quantity with discount.

Return type `shoop.core.pricing.Price`

tax_percentage

Return type `decimal.Decimal`

tax_rate

Return type `decimal.Decimal`

taxful_base_price

Taxful `base_price`

taxful_base_unit_price

Taxful `base_unit_price`

taxful_discount_amount

Taxful `discount_amount`

taxful_discounted_unit_price

Taxful `discounted_unit_price`

taxful_price

Return type `TaxfulPrice`

taxful_unit_discount_amount

Taxful `unit_discount_amount`

taxless_base_price

Taxless `base_price`

taxless_base_unit_price

Taxless `base_unit_price`

taxless_discount_amount

Taxless `discount_amount`

taxless_discounted_unit_price

Taxless `discounted_unit_price`

taxless_price

Return type *TaxlessPrice*

taxless_unit_discount_amount
Taxless *unit_discount_amount*

unit_discount_amount
Discount amount per unit.

If quantity is 0, will return `discount_amount`.

Return type *shoop.core.pricing.Price*

class `shoop.core.pricing.PriceInfo` (*price, base_price, quantity, expires_on=None*)
Bases: *shoop.core.pricing.Priceful*

Object for passing around pricing data of an item.

Initialize `PriceInfo` with prices and other parameters.

Prices can be taxful or taxless, but their types must match.

Parameters

- **price** (*Price*) – Effective price for the specified quantity.
- **base_price** (*Price*) – Base price for the specified quantity. Discounts are calculated based on this.
- **quantity** (*numbers.Number*) – Quantity that the given price is for. Unit price is calculated by `discounted_unit_price = price / quantity`. Note: Quantity could be non-integral (i.e. decimal).
- **expires_on** (*numbers.Number/None*) – Timestamp, comparable to values returned by `time.time`, determining the point in time when the prices are no longer valid, or `None` if no expire time is set (which could mean indefinitely, but in reality, it just means undefined).

base_price = `None`

price = `None`

quantity = `None`

class `shoop.core.pricing.PricingContext` (*shop, customer, time=None*)
Bases: *shoop.core.pricing.PricingContextable*

Context for pricing.

Initialize pricing context for shop and customer.

class `shoop.core.pricing.PricingContextable`
Bases: `object`

Object that is or can be converted to a pricing context.

Currently there exists two kind of *PricingContextable* objects: `PricingContext` (and its subclasses) and `HttpRequest`.

Note: Expression `isinstance(request, PricingContextable)` will return `True` for a request which is `HttpRequest`, because `HttpRequest` is registered as a subclass of this abstract base class.

This abstract base class is just a helper to allow writing simpler type specifiers, since we want to allow passing `HttpRequest` as a pricing context even though it is not a `PricingContext`.

class `shoop.core.pricing.PricingModule`

Bases: `object`

get_context (*context*)

Create pricing context from pricing contextable object.

Return type `PricingContext`

get_context_from_data (*shop, customer, time=None, **kwargs*)

Create pricing context from given arguments.

Return type `PricingContext`

get_context_from_request (*request*)

Create pricing context from HTTP request.

This base class implementation does not use `request` at all.

Return type `PricingContext`

get_price_info (*context, product, quantity=1*)

Get price info of product for given quantity.

Parameters **product** (`shoop.core.models.Product|int`) – Product object or id of Product

Return type `PriceInfo`

get_price_infos (*context, products, quantity=1*)

Get `PriceInfo` objects for a bunch of products.

Returns a dict with product id as key and `PriceInfo` as value.

May be faster than doing `get_price_info` for each product separately, since inheriting class may override this.

Parameters **products** (`Iterable[shoop.core.models.Product|int]`) – List of product objects or id's

Return type `dict[int,PriceInfo]`

get_pricing_steps (*context, product*)

Get context-specific list pricing steps for the given product.

Returns a list of `PriceInfos` [`pi0, pi1, pi2, ...`] where each `PriceInfo` object is at the border unit price change: unit price for $0 \leq \text{quantity} < \text{pi1.quantity1}$ is `pi0.discounted_unit_price`, and unit price for $\text{pi1.quantity} \leq \text{quantity} < \text{pi2.quantity}$ is `pi1.discounted_unit_price`, and so on.

If there are “no steps”, the return value will be a list of single `PriceInfo` object with the constant price, i.e. [`price_info`].

Parameters **product** (`shoop.core.models.Product|int`) – Product or product id

Return type `list[PriceInfo]`

get_pricing_steps_for_products (*context, products*)

Get pricing steps for a bunch of products.

Returns a dict with product id as key and step data (as list of `PriceInfos`) as values.

May be faster than doing `get_pricing_steps` for each product separately, since inheriting class may override this.

Parameters **products** (*Iterable[shoop.core.models.Product|int]*) – List of product objects or id's

Return type dict[int,list[PriceInfo]]

identifier = None

name = None

pricing_context_class
alias of *PricingContext*

class *shoop.core.pricing.TaxfulPrice*
Bases: *shoop.core.pricing.Price*

Price which includes taxes.

Check the base class, *Price*, for more info.

includes_tax = True

class *shoop.core.pricing.TaxlessPrice*
Bases: *shoop.core.pricing.Price*

Price which does not include taxes.

Check the base class, *Price*, for more info.

includes_tax = False

shoop.core.shortcuts package

Module contents

shoop.core.shortcuts.update_order_line_from_product (*pricing_context*, *order_line*,
product, *quantity=1*, *supplier=None*)

Update OrderLine data from a product.

This is a convenience method for simple applications.

Parameters **pricing_context** (*shoop.core.pricing.PricingContextable|None*)
– Pricing context to use for pricing the line. If None is given, the line will get zero price and zero discount amount.

shoop.core.stocks package

Module contents

class *shoop.core.stocks.ProductStockStatus* (*product=None*, *product_id=None*, *logical_count=0*, *physical_count=0*, *message=None*,
error=None)
Bases: *shoop.core.utils.product_caching_object.ProductCachingObject*

shoop.core.suppliers package

Submodules

shoop.core.suppliers.base module

class `shoop.core.suppliers.base.BaseSupplierModule` (*supplier, options*)

Bases: `object`

Base supplier module implementation.

identifier = `None`

name = `None`

get_stock_statuses (*product_ids*)

Parameters **product_ids** – Iterable of product IDs

Returns Dict of {product_id: ProductStockStatus}

Return type dict[int, shoop.core.stocks.ProductStockStatus]

get_stock_status (*product_id*)

Parameters **product_id** (*int*) – Product ID

Return type `shoop.core.stocks.ProductStockStatus`

get_orderability_errors (*shop_product, quantity, customer*)

Parameters

- **shop_product** (`shoop.core.models.ShopProduct`) – Shop Product
- **quantity** (`decimal.Decimal`) – Quantity to order
- **customer** – Contact

Return type iterable[ValidationError]

adjust_stock (*product_id, delta, created_by=None*)

update_stock (*product_id*)

update_stocks (*product_ids*)

Module contents

class `shoop.core.suppliers.BaseSupplierModule` (*supplier, options*)

Bases: `object`

Base supplier module implementation.

adjust_stock (*product_id, delta, created_by=None*)

get_orderability_errors (*shop_product, quantity, customer*)

Parameters

- **shop_product** (`shoop.core.models.ShopProduct`) – Shop Product
- **quantity** (`decimal.Decimal`) – Quantity to order
- **customer** – Contact

Return type iterable[ValidationError]

get_stock_status (*product_id*)

Parameters **product_id** (*int*) – Product ID

Return type *shoop.core.stocks.ProductStockStatus*

get_stock_statuses (*product_ids*)

Parameters *product_ids* – Iterable of product IDs

Returns Dict of {product_id: ProductStockStatus}

Return type dict[int, shoop.core.stocks.ProductStockStatus]

identifier = None

name = None

update_stock (*product_id*)

update_stocks (*product_ids*)

shoop.core.taxing package

Submodules

shoop.core.taxing.utils module

shoop.core.taxing.utils.get_tax_class_proportions (*lines*)

Generate tax class proportions from taxed lines.

Sum prices per tax class and return a list of (tax_class, factor) pairs, where factor is the proportion of total price of lines with given tax class from the total price of all lines.

Parameters *lines* (*list*[*shoop.core.order_creator.SourceLine*]) – List of taxed lines to generate proportions from

Return type list[(shoop.core.models.TaxClass, decimal.Decimal)]

Returns List of tax classes with a proportion, or empty list if total price is zero. Sum of proportions is 1.

shoop.core.taxing.utils.stacked_value_added_taxes (*price*, *taxes*)

Stack added taxes on the given price without compounding.

Note that this will not take compound taxation (Quebec) into account.

Parameters

- **price** (*shoop.core.pricing.Price*) – Taxful or taxless price to calculate taxes for
- **taxes** (*list*[*shoop.core.models.Tax*]) – List of Tax objects

Returns TaxedPrice with the calculated taxes.

Return type *TaxedPrice*

shoop.core.taxing.utils.calculate_compounded_added_taxes (*price*, *tax_groups*)

Calculate compounded and added taxes from given groups of taxes.

The *tax_groups* argument should be a list of tax groups, where each tax group is a list of Tax objects. Taxes in each tax group will be added together and finally each added tax group will be compounded over each other.

Parameters

- **price** (*shoop.core.pricing.Price*) – Taxful or taxless price to calculate taxes for
- **tax_groups** (*list*[*list*[*shoop.core.models.Tax*]]) – List of tax groups, each being a list of taxes

Returns TaxedPrice with the calculated taxes.

Return type *TaxedPrice*

Module contents

class `shoop.core.taxing.LineTax`

Bases: `object`

Tax of some line.

This is an interface for specifying taxes of an `OrderLine` or `SourceLine`.

tax

(*Tax*) The tax that this line is about.

name

(*str*) Name of the tax.

amount

(*Money*) Tax amount.

base_amount

(*Money*) Amount that this tax is calculated from.

classmethod `from_tax` (*tax*, *base_amount*, ***kwargs*)

Create tax line for given tax and base amount.

rate

class `shoop.core.taxing.SourceLineTax` (*tax*, *name*, *amount*, *base_amount*)

Bases: `shoop.core.taxing.LineTax`

Initialize line tax from given values.

class `shoop.core.taxing.TaxModule`

Bases: `object`

Module for calculating taxes.

add_taxes (*source*, *lines*)

Add taxes to given OrderSource lines.

Given lines are modified in-place, also new lines may be added (with `lines.extend` for example). If there is any existing taxes for the `lines`, they are simply replaced.

Parameters

- **source** (`shoop.core.order_creator.OrderSource`) – OrderSource of the lines
- **lines** (`list[shoop.core.order_creator.SourceLine]`) – List of lines to add taxes for

calculating_is_cheap = `True`

get_context_from_data (***context_data*)

get_context_from_order_source (*source*)

get_context_from_request (*request*)

get_taxed_price (*context, price, tax_class*)

Get TaxedPrice for price and tax class.

Parameters

- **context** (*TaxingContext*) – Taxing context to calculate in
- **price** (*shoop.core.pricing.Price*) – Price (taxful or taxless) to calculate taxes for
- **tax_class** (*shoop.core.models.TaxClass*) – Tax class of the item to get taxes for

Return type *shoop.core.taxing.TaxedPrice*

get_taxed_price_for (*context, item, price*)

Get TaxedPrice for taxable item.

Taxable items could be products (*Product*), services (*Service*), or lines (*SourceLine*).

Parameters

- **context** (*TaxingContext*) – Taxing context to calculate in
- **item** (*shoop.core.taxing.TaxableItem*) – Item to get taxes for
- **price** (*shoop.core.pricing.Price*) – Price (taxful or taxless) to calculate taxes for

Return type *shoop.core.taxing.TaxedPrice*

identifier = None

name = None

taxing_context_class
alias of *TaxingContext*

class *shoop.core.taxing.TaxSummary*

Bases: *list*

classmethod **from_line_taxes** (*line_taxes, untaxed*)

Create TaxSummary from LineTaxes.

Parameters

- **line_taxes** (*list [LineTax]*) – List of line taxes to summarize
- **untaxed** (*shoop.core.pricing.TaxlessPrice*) – Sum of taxless prices that have no taxes added

class *shoop.core.taxing.TaxableItem*

Bases: *object*

tax_class

Return type *shoop.core.models.TaxClass*

class *shoop.core.taxing.TaxedPrice* (*taxful, taxless, taxes=None*)

Bases: *object*

Price with calculated taxes.

taxful
(*TaxfulPrice*) Price including taxes.

taxless

(*TaxlessPrice*) Pretax price.

taxes

(list[shoop.core.taxing.LineTax]) List of taxes applied to the price.

Initialize from given prices and taxes.

Parameters

- **taxful** (shoop.core.pricing.TaxfulPrice) – Price including taxes.
- **taxless** (shoop.core.pricing.TaxlessPrice) – Pretax price.
- **taxes** (list[shoop.core.taxing.LineTax] | None) – List of taxes applied to the price.

tax_amount

Total amount of applied taxes.

tax_rate

Tax rate calculated from taxful and taxless amounts.

class shoop.core.taxing.**TaxingContext** (*customer_tax_group=None*, *customer_tax_number=None*, *location=None*)

Bases: `object`

shoop.core.taxing.**get_tax_module**()

Get the TaxModule specified in settings.

Return type *shoop.core.taxing.TaxModule*

shoop.core.taxing.**should_calculate_taxes_automatically**()

If `settings.SHOOP_CALCULATE_TAXES_AUTOMATICALLY_IF_POSSIBLE` is `False` taxes shouldn't be calculated automatically otherwise use current tax module value `TaxModule.calculating_is_cheap` to determine whether taxes should be calculated automatically.

Return type `bool`

shoop.core.templatetags package**Submodules**

shoop.core.templatetags.prices module Template tags for displaying prices correctly.

Prefer these filters for rendering prices of products and basket lines or total price of basket, since they will take price display options of current template context into account (see `PriceDisplayOptions`). Especially, they convert prices to correct taxness.

There is also a global context function `show_prices` which can be used to render certain price container elements conditionally.

shoop.core.templatetags.prices.**show_prices** (*context*)

Return true if price display options has show prices enabled.

shoop.core.templatetags.shoop_common module Common helpers for Shoop templates.

Note: In addition to these, also the price rendering tags from `shoop.core.templatetags.prices` module are available.

`shoop.core.templatetags.shoop_common.get_language_choices()`

Get language choices as code and text in two languages.

Returns Available language codes as tuples (code, name, local_name) where name is in the currently active language, and local_name is in the language of the item.

Return type Iterable[tuple[str, str, str]]

`shoop.core.templatetags.shoop_common.money(amount, digits=None, widen=0)`

Format money amount according to current locale settings.

Parameters

- **amount** (`shoop.utils.money.Money`) – Money or Price object to format
- **digits** (`int` / `None`) – Number of digits to use, by default use locale’s default
- **widen** (`int`) – Number of extra digits to add; for formatting with additional precision, e.g. `widen=3` will use 5 digits instead of 2

Returns Formatted string representing the given amount

Return type `str`

`shoop.core.templatetags.shoop_common.percent(value, ndigits=0)`

`shoop.core.templatetags.shoop_common.number(value)`

`shoop.core.templatetags.shoop_common.datetime(value, kind='datetime', format='medium', tz=True)`

Format a datetime for human consumption.

The currently active locale’s formatting rules are used. The output of this function is probably not machine-parseable.

Parameters

- **value** (`datetime.datetime`) – datetime object to format
- **kind** (`str`) – Format as ‘datetime’, ‘date’ or ‘time’
- **format** (`str`) – Format specifier or one of ‘full’, ‘long’, ‘medium’ or ‘short’
- **tz** (`bool` / `str`) – Convert to current or given timezone. Accepted values are:
 - True (default)** convert to currently active timezone (as reported by `django.utils.timezone.get_current_timezone()`)
 - False (or other false value like empty string)** do no convert to any timezone (use UTC)
 - Other values (as str)** convert to given timezone (e.g. "US/Hawaii")

`shoop.core.templatetags.shoop_common.json(value)`

Module contents

shoop.core.utils package

Submodules

shoop.core.utils.form_mixins module

class shoop.core.utils.form_mixins.**ProtectedFieldsMixin** (**kwargs)

Bases: `object`

change_protect_field_text = <django.utils.functional.lazy.<locals>.__proxy__ object>

disable_protected_fields()

clean_protected_fields(*cleaned_data*)

Ignore protected fields (they are set to disabled, so they will not be in the form data).

As a side effect, this removes the fields from `changed_data` too.

Parameters *cleaned_data* (*dict*) – Cleaned data

Returns Cleaned data without protected field data

Return type *dict*

clean()

shoop.core.utils.maintenance module

shoop.core.utils.maintenance.**maintenance_mode_exempt**(*view_func*)

Make view ignore shop maintenance mode

Parameters *view_func* – view attached to this decorator

Returns view added with `maintenance_mode_exempt` attribute

shoop.core.utils.model_caching_descriptor module

class shoop.core.utils.model_caching_descriptor.**ModelCachingDescriptor**(*name*,
query-set)

Bases: `object`

set_id(*instance*, *value*)

get_id(*instance*)

set_object(*instance*, *value*)

get_object(*instance*)

shoop.core.utils.name_mixin module

class shoop.core.utils.name_mixin.**NameMixin**

Bases: `object`

split_name

first_name

last_name

full_name

shoop.core.utils.price_display module Utilities for displaying prices correctly.

Contents:

- Class `PriceDisplayOptions` for storing the display options.
- Helper function `render_price_property` for rendering prices correctly from Python code.
- Various filter classes for implementing Jinja2 filters.

`shoop.core.utils.price_display.render_price_property(request, item, priceful, property_name='price')`

Render price property of a `Priceful` object.

Return type `str`

```
class shoop.core.utils.price_display.PriceDisplayFilter(name, property_name=None)
    Bases: shoop.core.utils.price_display._ContextFilter
```

```
class shoop.core.utils.price_display.PricePropertyFilter(name, property_name=None)
    Bases: shoop.core.utils.price_display._ContextFilter
```

```
class shoop.core.utils.price_display.PricePercentPropertyFilter(name, property_name=None)
    Bases: shoop.core.utils.price_display._ContextFilter
```

```
class shoop.core.utils.price_display.TotalPriceDisplayFilter(name, property_name=None)
    Bases: shoop.core.utils.price_display._ContextFilter
```

```
class shoop.core.utils.price_display.PriceRangeDisplayFilter(name, property_name=None)
    Bases: shoop.core.utils.price_display._ContextFilter
```

shoop.core.utils.prices module

`shoop.core.utils.prices.convert_taxness(request, item, priceful, with_taxes)`

Convert taxness of a `priceful` object.

Return a `Priceful` object `result` satisfying `result.price.includes_tax == with_taxes` if possible.

When given `priceful` does not have tax amount and taxes cannot be calculated automatically (`should_calculate_taxes_automatically` returns false), return the given `priceful` as is.

Given `request` is used for constructing a `TaxingContext`.

Return type `shoop.core.pricing.Priceful`

```
class shoop.core.utils.prices.TaxedPriceInfo(price, base_price, quantity, tax_amount,
                                             **kwargs)
    Bases: shoop.core.pricing.PriceInfo
```

shoop.core.utils.product_bought_with_relations module

`shoop.core.utils.product_bought_with_relations.add_bought_with_relations_for_product(product, max_quantity)`

Add `ProductCrossSell` objects with type `ProductCrossSellType.BUGHT_WITH` based on other products ordered with `product_id`. Ordered amount is used as relation weight.

Parameters

- **product_id** (`int`) – product_id to add relations to
- **max_quantity** (`int`) – maximum amount of relations created

shoop.core.utils.product_caching_object module

class `shoop.core.utils.product_caching_object.ProductCachingObject`

Bases: `object`

product

product_id

shoop.core.utils.query module

`shoop.core.utils.query.group_by_period(queryset, column, period, **annotate)`

Group and annotate given queryset by a given date period.

Parameters

- **queryset** (`django.db.QuerySet`) – Original queryset
- **column** (`str`) – Column for grouping
- **period** (`str`) – Period for grouping ('year', 'month', 'day')
- **annotate** (`dict[str, str]`) – Dict for annotate()

Returns `OrderedDict` of period -> annotate columns

Return type `collections.OrderedDict`

shoop.core.utils.slugs module

`shoop.core.utils.slugs.generate_multilanguage_slugs(object, name_getter, slug_length=128)`

shoop.core.utils.tax_numbers module

`shoop.core.utils.tax_numbers.validate(tax_number)`

Validate a tax number.

Parameters **tax_number** (`str`) – Tax number to validate

Returns Type identifier of the tax number, if detected. Possible values for now are either “vat” or “unknown”.

Return type `str`

Raise `ValidationError` if tax number type was detected, but it is somehow malformed.

shoop.core.utils.users module

`shoop.core.utils.users.real_user_or_none(user)`

Convert anonymous user to None.

If user is anonymous, return None, otherwise return the user as is.

shoop.core.utils.vat module

`shoop.core.utils.vat.compile_pattern(prefix, pattern)`

exception `shoop.core.utils.vat.VatValidationError(*args, **kwargs)`

Bases: `django.core.exceptions.ValidationError`

code = None

exception `shoop.core.utils.vat.VatCannotIdentifyValidationError(*args, **kwargs)`

Bases: `shoop.core.utils.vat.VatValidationError`

```
code = 'vat_cannot_identify'
```

```
exception shoop.core.utils.vat.VatInvalidValidationError(*args, **kwargs)
Bases: shoop.core.utils.vat.VatValidationError
```

```
code = 'vat_invalid'
```

```
shoop.core.utils.vat.verify_vat(vat_id, default_prefix='')
Verify an EU VAT ID.
```

Returns a tuple (prefix, code_parts) – if both are truthy, the validation succeeded. If the prefix part is falsy, then the prefix was unknown and no validation was even attempted. If the prefix part is truthy, then it will contain the country prefix used for validation. The code_parts part can still be falsy, if the validation for the country's VAT number pattern failed.

Parameters

- **vat_id** (*str*) – The VAT ID string to validate.
- **default_prefix** (*str*) – The default prefix to assume if none can be parsed.

Returns Tuple of (prefix, code_parts)

```
shoop.core.utils.vat.get_vat_prefix_for_country(iso3166)
```

Module contents

Submodules

shoop.core.excs module

```
exception shoop.core.excs.ImmutabilityError
Bases: ValueError
```

```
exception shoop.core.excs.NoProductsToShipException
Bases: Exception
```

```
exception shoop.core.excs.NoPaymentToCreateException
Bases: Exception
```

```
exception shoop.core.excs.NoRefundToCreateException
Bases: Exception
```

```
exception shoop.core.excs.RefundExceedsAmountException
Bases: Exception
```

```
exception shoop.core.excs.ProductNotOrderableProblem(message, title=None)
Bases: shoop.utils.excs.Problem
```

```
exception shoop.core.excs.ProductNotVisibleProblem(message, title=None)
Bases: shoop.utils.excs.Problem
```

```
exception shoop.core.excs.ImpossibleProductModeException(message, code=None)
Bases: ValueError
```

shoop.core.middleware module

```
class shoop.core.middleware.ExceptionMiddleware
Bases: object
```

```
process_exception(request, exception)
```


shoop.core.settings module

`shoop.core.settings.SHOOP_HOME_CURRENCY = 'EUR'`

The home currency for the Shoop installation. All monetary values are implicitly in this currency unless somehow otherwise specified.

`shoop.core.settings.SHOOP_ADDRESS_HOME_COUNTRY = None`

The home country code (ISO 3166-1 alpha 2) for the Shoop installation. Among other things, addresses that would be printed with this country visible are printed with no country.

`shoop.core.settings.SHOOP_ALLOW_ANONYMOUS_ORDERS = True`

Whether or not anonymous orders (without a `creator` user) are allowed.

`shoop.core.settings.SHOOP_ORDER_IDENTIFIER_METHOD = 'id'`

Which method is used to calculate order identifiers (“order numbers”). May be either the string “id”, a callable or a spec string pointing to a callable that must return a string given an `order`.

`shoop.core.settings.SHOOP_REFERENCE_NUMBER_METHOD = 'unique'`

Which method is used to calculate order reference numbers.

May be a spec string pointing to a callable that must return a string given an `Order`, or one of the following built-in generators.

unique Unique reference number based on time and the order ID. The reference number has the Finnish bank reference check digit appended, making the reference number valid for Finnish bank transfers.

running Ascending reference number. The length of the reference number will be `SHOOP_REFERENCE_NUMBER_LENGTH + 1` (for the check digit described below). `SHOOP_REFERENCE_NUMBER_PREFIX` is prepended, if set. The reference number has the Finnish bank reference check digit appended, making the reference number valid for Finnish bank transfers.

shop_running As `running`, but with the shop ID prepended.

`shoop.core.settings.SHOOP_REFERENCE_NUMBER_LENGTH = 10`

The length of reference numbers generated by certain reference number generators.

`shoop.core.settings.SHOOP_REFERENCE_NUMBER_PREFIX = ''`

An arbitrary (numeric) prefix for certain reference number generators.

`shoop.core.settings.SHOOP_PRICING_MODULE = 'customer_group_pricing'`

The identifier of the pricing module to use for pricing products.

Determines how product prices are calculated. See [shoop.core.pricing](#) for details.

`shoop.core.settings.SHOOP_DISCOUNT_MODULES = ['catalog_campaigns']`

List of identifiers of discount modules to use.

Each discount module may change the price of a product. See [shoop.core.pricing.DiscountModule](#) for details.

`shoop.core.settings.SHOOP_ORDER_SOURCE_MODIFIER_MODULES = ['basket_campaigns']`

List of identifiers of order source modifier modules.

See [shoop.core.order_creator.OrderSourceModifierModule](#) for details.

`shoop.core.settings.SHOOP_TAX_MODULE = 'default_tax'`

The identifier of the tax module to use for determining taxes of products and order lines.

Determines taxing rules for products, shipping/payment methods and other order lines. See [shoop.core.taxing](#) for details.

`shoop.core.settings.SHOOP_ENABLE_ATTRIBUTES = True`

Whether product attributes are enabled. For installations not requiring attributes, disabling this may confer a small performance increase.

`shoop.core.settings.SHOOP_ENABLE_MULTIPLE_SHOPS = False`

Whether multiple shops are expected to be enabled in this installation. Enabling or disabling this flag does not make it (im)possible to set up multiple shops, but having it disabled may confer a small performance increase.

`shoop.core.settings.SHOOP_ENABLE_MULTIPLE_SUPPLIERS = False`

Whether multiple suppliers are enabled in this installation. Enabling this flag prevents supplier creation from admin.

`shoop.core.settings.SHOOP_ORDER_LABELS = [('default', 'Default')]`

A list of order labels (2-tuples of internal identifier / visible name).

Order labels serve as a simple taxonomy layer for easy “tagging” of orders even within a single Shop. For instance, an installation could define "default" and "internal" order labels, which are then usable in reports, admin filtering, etc.

`shoop.core.settings.SHOOP_DEFAULT_ORDER_LABEL = 'default'`

The order label (see `SHOOP_ORDER_LABELS`) to apply to orders by default. This should naturally be one of the keys in `SHOOP_ORDER_LABELS`.

`shoop.core.settings.SHOOP_ORDER_KNOWN_PAYMENT_DATA_KEYS = []`

A list of “known keys” within the `Order.payment_data` property bag.

The format of this setting is a list of 2-tuples of dict key / visible name, for example `[("ssn", "Social Security Number")]`.

For installations where customizations may save some human-readable, possibly important information in `payment_data`, this setting may be used to make this data easily visible in the administration backend.

`shoop.core.settings.SHOOP_ORDER_KNOWN_SHIPPING_DATA_KEYS = []`

A list of “known keys” within the `Order.shipping_data` property bag.

The format of this setting is a list of 2-tuples of dict key / visible name, for example `[("shipping_instruction", "Special Shipping Instructions")]`.

For installations where customizations may save some human-readable, possibly important information in `shipping_data`, this setting may be used to make this data easily visible in the administration backend.

`shoop.core.settings.SHOOP_ORDER_KNOWN_EXTRA_DATA_KEYS = []`

A list of “known keys” within the `Order.extra_data` property bag.

The format of this setting is a list of 2-tuples of dict key / visible name, for example `[("wrapping_color", "Wrapping Paper Color")]`.

For installations where customizations may save some human-readable, possibly important information in `extra_data`, this setting may be used to make this data easily visible in the administration backend.

`shoop.core.settings.SHOOP_TELEMETRY_ENABLED = True`

A flag to enable/disable the telemetry system

`shoop.core.settings.SHOOP_TELEMETRY_URL = 'https://telemetry.shoop.io/collect/'`

The submission URL for Shoop’s telemetry (statistics) system

`shoop.core.settings.SHOOP_DEFAULT_CACHE_DURATION = 1800`

Default cache duration for various caches in seconds

`shoop.core.settings.SHOOP_CACHE_DURATIONS = {}`

Overrides for default cache durations by key namespace. These override possible defaults in `shoop.core.cache.impl.DEFAULT_CACHE_DURATIONS`.

`shoop.core.settings.SHOOP_CALCULATE_TAXES_AUTOMATICALLY_IF_POSSIBLE = True`

Whether taxes should be calculated automatically in `TaxModule`

shoop.core.signals module**shoop.core.telemetry module**

`shoop.core.telemetry.safe_json(data_dict, indent=None)`

`shoop.core.telemetry.get_installation_key()`

Get the unique installation ID for this Shoop instance.

If one doesn't exist, it's generated and saved at this point.

Returns Installation key string

Return type `str`

`shoop.core.telemetry.is_opt_out()`

`shoop.core.telemetry.is_in_grace_period()`

Return True if the telemetry module is within the 24-hours-from-installation grace period where no stats are sent. This is to “safely” allow opting out of telemetry without leaving a trace.

Returns Graceness flag.

Return type `bool`

`shoop.core.telemetry.is_telemetry_enabled()`

`shoop.core.telemetry.set_opt_out(flag)`

Set whether this installation is opted-out from telemetry submissions.

Parameters `flag` (`bool`) – Opt-out flag. True for opt-out, false for opt-in (default)

Returns New flag state

Return type `bool`

`shoop.core.telemetry.get_last_submission_time()`

`shoop.core.telemetry.get_last_submission_data()`

`shoop.core.telemetry.save_telemetry_submission(data)`

Save a blob of data as the latest telemetry submission.

Naturally updates the latest submission time.

Parameters `data` (`dict`) – A blob of data.

`shoop.core.telemetry.get_telemetry_data(request, indent=None)`

Get the telemetry data that would be sent.

Parameters `request` (`django.http.HttpRequest` | `None`) – HTTP request. Optional.

Returns Data blob.

Return type `str`

exception `shoop.core.telemetry.TelemetryNotSent(message, code)`

Bases: `Exception`

`shoop.core.telemetry.try_send_telemetry(request=None, max_age_hours=72, raise_on_error=False)`

Send telemetry information (unless opted-out, in grace period or disabled).

Parameters

- `request` (`django.http.HttpRequest` | `None`) – HTTP request. Optional.

- **max_age_hours** (*int* / *None*) – How many hours must have passed since the last submission to be able to resend. If *None*, not checked.
- **raise_on_error** (*bool*) – Raise exceptions when telemetry is not sent instead of quietly returning *False*.

Returns Sent data (possibly with response information) or *False* if not sent.

Return type dict|bool

Module contents

class `shoop.core.ShoopCoreAppConfig` (**args*, ***kwargs*)

Bases: `shoop.apps.AppConfig`

name = 'shoop.core'

verbose_name = 'Shoop Core'

label = 'shoop'

required_installed_apps = ('django.contrib.auth', 'django.contrib.contenttypes', 'easy_thumbnails', 'filer')

provides = {'pricing_module': ['shoop.core.pricing.default_pricing:DefaultPricingModule'], 'api_populator': ['shoop.c

shoop.customer_group_pricing package

Submodules

shoop.customer_group_pricing.admin_form_part module

class `shoop.customer_group_pricing.admin_form_part.CustomerGroupPricingForm` (***kwargs*)

Bases: `django.forms.forms.Form`

save ()

get_shop_group_field (*shop*, *group*)

base_fields = `OrderedDict()`

declared_fields = `OrderedDict()`

media

class `shoop.customer_group_pricing.admin_form_part.CustomerGroupPricingFormPart` (*request*, *object=None*)

Bases: `shoop.admin.form_part.FormPart`

priority = 10

get_form_defs ()

form_valid (*form*)

shoop.customer_group_pricing.api module

`shoop.customer_group_pricing.api.populate_customer_group_pricing_api` (*router*)

Parameters **router** (`rest_framework.routers.DefaultRouter`) – Router

shoop.customer_group_pricing.models module

class shoop.customer_group_pricing.models.**CgpPrice** (*id, product, shop, group, price_value*)

Bases: *shoop.utils.properties.MoneyPropped*, *django.db.models.base.Model*

product

shop

group

price

Property for Price object.

Similar to MoneyProperty but also has `includes_tax` field.

Operators with TaxfulPrice and TaxlessPrice objects.

exception DoesNotExist

Bases: *django.core.exceptions.ObjectDoesNotExist*

exception CgpPrice.MultipleObjectsReturned

Bases: *django.core.exceptions.MultipleObjectsReturned*

CgpPrice.objects = <django.db.models.manager.Manager object>

shoop.customer_group_pricing.module module

class shoop.customer_group_pricing.module.**CustomerGroupPricingModule**

Bases: *shoop.core.pricing.PricingModule*

identifier = 'customer_group_pricing'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

get_price_info (*context, product, quantity=1*)

Module contents

class shoop.customer_group_pricing.**CustomerGroupPricingAppConfig** (**args,*
***kwargs*)

Bases: *shoop.apps.AppConfig*

name = 'shoop.customer_group_pricing'

verbose_name = <django.utils.functional.lazy.<locals>.__proxy__ object>

label = 'shoop_customer_group_pricing'

provides = {'pricing_module': ['shoop.customer_group_pricing.module:CustomerGroupPricingModule'], 'api_popula

shoop.default_tax package**Subpackages****shoop.default_tax.admin_module package****Submodules**

shoop.default_tax.admin_module.views module

```
class shoop.default_tax.admin_module.views.TaxRuleForm(data=None,          files=None,
                                                         auto_id='id_%s',          pre-
                                                         fix=None,          initial=None,
                                                         error_class=<class
                                                         'django.forms.utils.ErrorList'>,
                                                         label_suffix=None,
                                                         empty_permitted=False,    in-
                                                         stance=None)

Bases: django.forms.models.ModelForm

class Meta
    Bases: object

    model
        alias of TaxRule

    fields = ['tax_classes', 'customer_tax_groups', 'country_codes_pattern', 'region_codes_pattern', 'postal_codes_p
    help_texts = {'postal_codes_pattern': <django.utils.functional.lazy.<locals>.__proxy__ object at 0x7f713e72bd30>

TaxRuleForm.clean()

TaxRuleForm.base_fields = OrderedDict([('tax_classes', <django.forms.models.ModelMultipleChoiceField object a
TaxRuleForm.declared_fields = OrderedDict()

TaxRuleForm.media
class shoop.default_tax.admin_module.views.TaxRuleEditView(**kwargs)
    Bases: shoop.admin.utils.views.CreateOrUpdateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of TaxRule

    template_name = 'shoop/default_tax/admin/edit.jinja'

    form_class
        alias of TaxRuleForm

    context_object_name = 'tax_rule'

    add_form_errors_as_messages = True

class shoop.default_tax.admin_module.views.TaxRuleListView(**kwargs)
    Bases: shoop.admin.utils.views.PicotableView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    model
        alias of TaxRule

    columns = [<shoop.admin.utils.picotable.Column object at 0x7f71392e3898>, <shoop.admin.utils.picotable.Column obje
```

Module contents

```
class shoop.default_tax.admin_module.TaxRulesAdminModule
    Bases: shoop.admin.base.AdminModule

    name = <django.utils.functional.lazy.<locals>.__proxy__ object>

    category = <django.utils.functional.lazy.<locals>.__proxy__ object>

    breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>
```

```

get_urls()
get_menu_entries(request)
get_required_permissions()
get_model_url(object, kind)

```

Submodules

shoop.default_tax.models module

```

class shoop.default_tax.models.TaxRuleQuerySet(model=None, query=None, using=None,
                                              hints=None)

```

Bases: `django.db.models.query.QuerySet`

may_match_postal_code(postalcode)

```

class shoop.default_tax.models.TaxRule(id, enabled, country_codes_pattern, re-
                                     gion_codes_pattern, postal_codes_pattern,
                                     _postal_codes_min, _postal_codes_max, priority,
                                     override_group, tax)

```

Bases: `django.db.models.base.Model`

tax_classes

customer_tax_groups

tax

objects = <django.db.models.manager.ManagerFromTaxRuleQuerySet object>

matches(taxing_context)

Check if this tax rule matches given taxing context.

save(*args, **kwargs)

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception TaxRule.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

shoop.default_tax.module module

```

class shoop.default_tax.module.DefaultTaxModule

```

Bases: `shoop.core.taxing.TaxModule`

identifier = 'default_tax'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

get_taxed_price(context, price, tax_class)

```

shoop.default_tax.module.get_taxes_of_effective_rules(taxing_context, tax_rules)

```

Get taxes grouped by priority from effective tax rules.

Effective tax rules is determined by first limiting the scope to the rules that match the given taxing context (see `TaxRule.match`) and then further limiting the matching rules by selecting only the rules in the highest numbered override group.

The `Tax` objects in the effective rules will be grouped by the priority of the rules. The tax groups are returned as list of tax lists.

Parameters **tax_rules** (*Iterable[TaxRule]*) – Tax rules to filter from. These should be ordered descending by override group and then ascending by priority.

Return type list[list[shoop.core.models.Tax]]

Module contents

class `shoop.default_tax.AppConfig` (**args, **kwargs*)

Bases: `shoop.apps.AppConfig`

name = 'shoop.default_tax'

verbose_name = <django.utils.functional.lazy.<locals>.__proxy__ object>

label = 'default_tax'

provides = {'admin_module': ['shoop.default_tax.admin_module:TaxRulesAdminModule'], 'tax_module': ['shoop.defa

shoop.front package

Subpackages

shoop.front.admin_module package

Subpackages

shoop.front.admin_module.carts package

Subpackages

shoop.front.admin_module.carts.views package

Module contents

class `shoop.front.admin_module.carts.views.CartListView` (***kwargs*)

Bases: `shoop.admin.utils.views.PicotableView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

columns = [<shoop.admin.utils.picotable.Column object at 0x7f7138983c18>, <shoop.admin.utils.picotable.Column obje

format_abandoned_status (*instance, *args, **kwargs*)

format_created_date (*instance, *args, **kwargs*)

format_taxful_total_price (*instance, *args, **kwargs*)

format_updated_date (*instance, *args, **kwargs*)

get_context_data (***kwargs*)

get_object_abstract (*instance, item*)

get_queryset ()

Ignore potentially active carts, displaying only those not updated for at least 2 hours.

model

alias of `StoredBasket`

Module contents

Module contents

```
shoop.front.admin_module.get_unfinalized_cart_block(currency, days=14)
class shoop.front.admin_module.CartAdminModule(currency=None, *args, **kwargs)
    Bases: shoop.admin.currencybound.CurrencyBound, shoop.admin.base.AdminModule
    name = 'Cart'
    get_dashboard_blocks(request)
    get_urls()
    get_menu_category_icons()
    get_required_permissions()
    get_menu_entries(request)
```

shoop.front.apps package

Subpackages

shoop.front.apps.auth package

Submodules

shoop.front.apps.auth.forms module

```
class shoop.front.apps.auth.forms.EmailAuthenticationForm(*args, **kwargs)
    Bases: django.contrib.auth.forms.AuthenticationForm
    error_messages = {'inactive': 'This account is inactive.', 'invalid_login': 'Please enter a correct %(username)s and password'}
    clean_username()
    confirm_login_allowed(user)
        Do not let user with inactive person contact to login.
    base_fields = OrderedDict([('username', <django.forms.fields.CharField object at 0x7f713e99dba8>), ('password', <django.forms.fields.CharField object at 0x7f713e99dba8>)])
    declared_fields = OrderedDict([('username', <django.forms.fields.CharField object at 0x7f713e99dba8>), ('password', <django.forms.fields.CharField object at 0x7f713e99dba8>)])
    media
class shoop.front.apps.auth.forms.RecoverPasswordForm(data=None, files=None, auto_id='id_%s', pre-fix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=None, empty_permitted=False)
    Bases: django.forms.forms.Form
    token_generator = <django.contrib.auth.tokens.PasswordResetTokenGenerator object>
    subject_template_name = ('shoop/user/recover_password_mail_subject.jinja',)
    email_template_name = 'shoop/user/recover_password_mail_content.jinja'
```

```
from_email = None
clean()
save(request)
process_user(user_to_recover)
base_fields = OrderedDict([('username', <django.forms.fields.CharField object at 0x7f713e2aaeb8>), ('email', <django.forms.fields.CharField object at 0x7f713e2aaeb8>), ('password', <django.forms.fields.CharField object at 0x7f713e2aaeb8>)])
declared_fields = OrderedDict([('username', <django.forms.fields.CharField object at 0x7f713e2aaeb8>), ('email', <django.forms.fields.CharField object at 0x7f713e2aaeb8>), ('password', <django.forms.fields.CharField object at 0x7f713e2aaeb8>)])
media
```

shoop.front.apps.auth.urls module

shoop.front.apps.auth.views module

```
class shoop.front.apps.auth.views.LoginView(**kwargs)
    Bases: django.views.generic.edit.FormView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    template_name = 'shoop/user/login.jinja'
    form_class
        alias of EmailAuthenticationForm
    get_context_data(**kwargs)
    get_form(form_class=None)
    form_valid(form)
    get_success_url()
class shoop.front.apps.auth.views.LogoutView(**kwargs)
    Bases: django.views.generic.base.TemplateView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    template_name = 'shoop/user/logout.jinja'
    dispatch(request, *args, **kwargs)
class shoop.front.apps.auth.views.RecoverPasswordView(**kwargs)
    Bases: django.views.generic.edit.FormView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    template_name = 'shoop/user/recover_password.jinja'
    form_class
        alias of RecoverPasswordForm
    success_url = <django.utils.functional.lazy.<locals>.__proxy__ object>
    form_valid(form)
class shoop.front.apps.auth.views.RecoverPasswordConfirmView(**kwargs)
    Bases: django.views.generic.edit.FormView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    template_name = 'shoop/user/recover_password_confirm.jinja'
```

```

form_class
    alias of SetPasswordForm

token_generator = <django.contrib.auth.tokens.PasswordResetTokenGenerator object>

success_url = <django.utils.functional.lazy.<locals>.__proxy__ object>

get_form_kwargs()

get_target_user()

dispatch(request, *args, **kwargs)

form_valid(form)

```

```

class shoop.front.apps.auth.views.RecoverPasswordSentView(**kwargs)
    Bases: django.views.generic.base.TemplateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    template_name = 'shoop/user/recover_password_sent.jinja'

class shoop.front.apps.auth.views.RecoverPasswordCompleteView(**kwargs)
    Bases: django.views.generic.base.TemplateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    template_name = 'shoop/user/recover_password_complete.jinja'

```

Module contents

```

class shoop.front.apps.auth.AuthAppConfig(*args, **kwargs)
    Bases: shoop.apps.AppConfig

    name = 'shoop.front.apps.auth'

    verbose_name = 'Shoop Frontend - User Authentication'

    label = 'shoop_front.auth'

    provides = {'front_urls': ['shoop.front.apps.auth.urls:urlpatterns']}

```

shoop.front.apps.customer_information package

Submodules

shoop.front.apps.customer_information.forms module

```

class shoop.front.apps.customer_information.forms.PersonContactForm(*args,
                                                                    **kwargs)

    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        model
            alias of PersonContact

        fields = ('first_name', 'last_name', 'phone', 'email', 'gender', 'marketing_permission')

    PersonContactForm.base_fields = OrderedDict([('first_name', <django.forms.fields.CharField object at 0x7f713...

```

```
PersonContactForm.declared_fields = OrderedDict()

PersonContactForm.media
class shoop.front.apps.customer_information.forms.AddressForm(*args, **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        model
            alias of MutableAddress

        fields = ('name', 'phone', 'email', 'street', 'street2', 'postal_code', 'city', 'region', 'country')

AddressForm.base_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713e2bc6a0>), ('p
AddressForm.declared_fields = OrderedDict()

AddressForm.media
class shoop.front.apps.customer_information.forms.CompanyContactForm(*args,
                                                                    **kwargs)

    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        model
            alias of CompanyContact

        fields = ('name', 'tax_number', 'phone', 'email', 'marketing_permission')

CompanyContactForm.base_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713e2c1
CompanyContactForm.declared_fields = OrderedDict()

CompanyContactForm.media
```

shoop.front.apps.customer_information.urls module

shoop.front.apps.customer_information.views module

```
shoop.front.apps.customer_information.views.change_password(request)
class shoop.front.apps.customer_information.views.CustomerEditView(**kwargs)
    Bases: django.views.generic.edit.FormView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    template_name = 'shoop/customer_information/edit_customer.jinja'

    get_form(form_class=None)

    form_valid(form)

class shoop.front.apps.customer_information.views.CompanyEditView(**kwargs)
    Bases: django.views.generic.edit.FormView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    template_name = 'shoop/customer_information/edit_company.jinja'

    get_form(form_class=None)

    form_valid(form)
```

Module contents

```
class shoop.front.apps.customer_information.AppConfig(*args, **kwargs)
    Bases: shoop.apps.AppConfig

    name = 'shoop.front.apps.customer_information'

    verbose_name = <django.utils.functional.lazy.<locals>.__proxy__ object>

    label = 'shoop_front.customer_information'

    provides = {'front_urls': ['shoop.front.apps.customer_information.urls:urlpatterns']}
```

shoop.front.apps.personal_order_history package

Submodules

shoop.front.apps.personal_order_history.urls module

shoop.front.apps.personal_order_history.views module

```
class shoop.front.apps.personal_order_history.views.OrderViewMixin
    Bases: object

    model
        alias of Order

    get_queryset()

class shoop.front.apps.personal_order_history.views.OrderListView(**kwargs)
    Bases: shoop.front.apps.personal_order_history.views.OrderViewMixin,
django.views.generic.list.ListView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    template_name = 'shoop/personal_order_history/order_list.jinja'

    context_object_name = 'orders'

class shoop.front.apps.personal_order_history.views.OrderDetailView(**kwargs)
    Bases: shoop.front.apps.personal_order_history.views.OrderViewMixin,
django.views.generic.detail.DetailView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    template_name = 'shoop/personal_order_history/order_detail.jinja'

    context_object_name = 'order'
```

Module contents

```
class shoop.front.apps.personal_order_history.AppConfig(*args, **kwargs)
    Bases: shoop.apps.AppConfig

    name = 'shoop.front.apps.personal_order_history'

    verbose_name = <django.utils.functional.lazy.<locals>.__proxy__ object>

    label = 'shoop_front.personal_order_history'

    provides = {'front_urls': ['shoop.front.apps.personal_order_history.urls:urlpatterns']}
```

shoop.front.apps.registration package

Submodules

shoop.front.apps.registration.settings module

`shoop.front.apps.registration.settings.SHOOP_REGISTRATION_REQUIRES_ACTIVATION = True`
Require email-based activation for users?

This corresponds to using the default or simple django-registration backends.

shoop.front.apps.registration.urls module

shoop.front.apps.registration.views module

`shoop.front.apps.registration.views.activation_complete(request)`
`shoop.front.apps.registration.views.registration_complete(request)`

class `shoop.front.apps.registration.views.RegistrationViewMixin`

Bases: `object`

template_name = 'shoop/registration/register.jinja'

get_success_url (*args, **kwargs)

class `shoop.front.apps.registration.views.RegistrationNoActivationView(**kwargs)`

Bases: `shoop.front.apps.registration.views.RegistrationViewMixin`,
`registration.backends.simple.views.RegistrationView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

class `shoop.front.apps.registration.views.RegistrationWithActivationView(**kwargs)`

Bases: `shoop.front.apps.registration.views.RegistrationViewMixin`,
`registration.backends.default.views.RegistrationView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

class `shoop.front.apps.registration.views.RegistrationView(**kwargs)`

Bases: `django.views.generic.base.View`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

dispatch (request, *args, **kwargs)

class `shoop.front.apps.registration.views.ActivationView(**kwargs)`

Bases: `registration.backends.default.views.ActivationView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

template_name = 'shoop/registration/activation_failed.jinja'

get_success_url (*args, **kwargs)

Module contents

Shoop Registration Add-on The `shoop.front.apps.registration` add-on provides simple user registration and email token based activation.

It is based on the `django-registration-redux` package.

Installation Add registration and `shoop.front.apps.registration` into your `INSTALLED_APPS` (and run migrations, of course).

The application registers its URLs via the `front_urls` provides mechanism.

URL names

- `shoop:registration_register` – the entry point for registration.

```
class shoop.front.apps.registration.RegistrationAppConfig(*args, **kwargs)
    Bases: shoop.apps.AppConfig

    name = 'shoop.front.apps.registration'
    verbose_name = 'Shoop Frontend - User Registration'
    label = 'shoop_front.registration'
    required_installed_apps = {'registration': 'django-registration-redux is required for user registration and activation'}
    provides = {'front_urls': ['shoop.front.apps.registration.urls:urlpatterns']}
    ready()
```

`shoop.front.apps.simple_order_notification` package

Subpackages

`shoop.front.apps.simple_order_notification.admin_module` package

Module contents

```
class shoop.front.apps.simple_order_notification.admin_module.SimpleOrderNotificationModule
    Bases: shoop.admin.base.AdminModule

    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    category = <django.utils.functional.lazy.<locals>.__proxy__ object>
    get_notifications(request)
```

Submodules

`shoop.front.apps.simple_order_notification.templates` module

Module contents

```
shoop.front.apps.simple_order_notification.send_simple_order_notification(sender,
                                                                            or-
                                                                            der,
                                                                            re-
                                                                            quest,
                                                                            **kwargs)
```

Parameters `order` (`shoop.core.models.Order`) – Order

```
class shoop.front.apps.simple_order_notification.SimpleOrderNotificationAppConfig(*args,
                                                                                    **kwargs)
    Bases: shoop.apps.AppConfig
```

```
name = 'shoop.front.apps.simple_order_notification'
verbose_name = 'Shoop Frontend - Simple Order Notification'
label = 'shoop_front.simple_order_notification'
provides = {'admin_module': ['shoop.front.apps.simple_order_notification.admin_module:SimpleOrderNotificationM
ready()
```

shoop.front.apps.simple_search package

Submodules

shoop.front.apps.simple_search.template_helpers module

class shoop.front.apps.simple_search.template_helpers.**TemplateHelpers**

Bases: `object`

name = 'simple_search'

get_search_form(*context*, *template_name*=`'shoop/simple_search/search_form.jinja'`)

Get a product search form, usable e.g. for navigation bars. The `q` request parameter is used by default to pre-fill the search query field. The name of the template rendered can be overridden with the `template_name` parameter.

Parameters

- **context** (`jinja2.runtime.Context`) – Template context
- **template_name** (`str`) – Template file name

shoop.front.apps.simple_search.urls module

shoop.front.apps.simple_search.views module

shoop.front.apps.simple_search.views.**get_search_product_ids**(*request*, *query*)

class shoop.front.apps.simple_search.views.**SearchForm**(*data*=`None`, *files*=`None`,
auto_id=`'id_%s'`, *pre*-
fix=`None`, *initial*=`None`,
error_class=<class
'django.forms.utils.ErrorList'>,
label_suffix=`None`,
empty_permitted=`False`)

Bases: `django.forms.forms.Form`

clean()

base_fields = `OrderedDict([('q', <django.forms.fields.CharField object at 0x7f713e2d0eb8>), ('sort', <django.forms.f`

declared_fields = `OrderedDict([('q', <django.forms.fields.CharField object at 0x7f713e2d0eb8>), ('sort', <django.fo`

media

class shoop.front.apps.simple_search.views.**SearchView**(***kwargs*)

Bases: `django.views.generic.list.ListView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

form_class

alias of `SearchForm`


```

template_name = 'shoop/simple_search/search.jinja'

model
    alias of Product

context_object_name = 'products'

dispatch(request, *args, **kwargs)

get_queryset()

get_context_data(**kwargs)

```

Module contents

```

class shoop.front.apps.simple_search.SimpleSearchAppConfig(*args, **kwargs)
    Bases: shoop.apps.AppConfig

    name = 'shoop.front.apps.simple_search'
    verbose_name = 'Shoop Frontend - Simple Search'
    label = 'shoop_front.simple_search'
    provides = {'front_template_helper_namespace': ['shoop.front.apps.simple_search.template_helpers:TemplateHelpers

```

Module contents

shoop.front.basket package

Submodules

shoop.front.basket.command_dispatcher module

```

class shoop.front.basket.command_dispatcher.BasketCommandDispatcher(request,
                                                                    bas-
                                                                    ket=None)

    Bases: object

```

BasketCommandDispatcher handles (usually AJAX) requests that somehow update the basket. You should never instantiate BasketCommandDispatcher yourself – instead use `get_basket_command_dispatcher()`.

All `handle_*` methods are expected to accept `**kwargs`.

```

commands_module = <module 'shoop.front.basket.commands' from '/home/docs/checkouts/readthedocs.org/user_builds/
get_command_handler(command)

handle(command, kwargs=None)
    Dispatch and handle processing of the given command.

```

Parameters

- **command** (*unicode*) – Name of command to run
- **kwargs** (*dict*) – Arguments to pass to the command handler. If empty, `request.POST` is used.

Returns response

Return type `HttpResponse`

preprocess_kwargs (*command, kwargs*)

Preprocess kwargs before they are passed to the given *command* handler. Useful for subclassing. Must return the new *kwargs*, even if it wasn't mutated.

Parameters

- **command** – The name of the command about to be run
- **kwargs** – dict of arguments

Returns dict of arguments

postprocess_response (*command, kwargs, response*)

Postprocess the response dictionary (not a HTTP response!) before it is either turned into JSON or otherwise processed (in the case of non-AJAX requests).

Parameters

- **command** – The command that was run.
- **kwargs** – The actual kwargs the command was run with.
- **response** – The response the command returned.

Returns The response to be processed and sent to the client.

shoop.front.basket.commands module

`shoop.front.basket.commands.handle_add` (*request, basket, product_id, quantity=1, supplier_id=None, **kwargs*)

Handle adding a product to the basket.

Parameters

- **product_id** – product ID to add (or if *child_product_id* is truey, the parent ID)
- **quantity** – quantity of products to add
- **child_product_id** – child product ID to add (if truey)
- **supplier_id** – The supplier ID for the new line. If None, the first supplier is used.

`shoop.front.basket.commands.handle_add_var` (*request, basket, product_id, quantity=1, **kwargs*)

Handle adding a complex variable product into the basket by resolving the combination variables. This actually uses *kwargs*, expecting *var_XXX=YYY* to exist there, where *XXX* is the PK of a *ProductVariationVariable* and *YYY* is the PK of a *ProductVariationVariableValue*. Confused yet?

Parameters

- **quantity** – Quantity of the resolved variation to add.
- **kwargs** – Expected to contain *var_** values, see above.

`shoop.front.basket.commands.handle_del` (*request, basket, line_id, **kwargs*)

Handle deleting a distinct order line from the basket given its unique line ID.

Parameters *line_id* – The line ID to delete.

Returns

`shoop.front.basket.commands.handle_clear` (*request, basket, **kwargs*)

Handle fully clearing the basket.

`shoop.front.basket.commands.handle_add_campaign_code` (*request, basket, code*)

`shoop.front.basket.commands.handle_update(request, basket, **kwargs)`

Handle updating a basket, i.e. deleting some lines or updating quantities.

This dispatches further to whatever is declared by the `SHOOP_BASKET_UPDATE_METHODS_SPEC` configuration entry.

shoop.front.basket.objects module

class `shoop.front.basket.objects.BasketLine` (*source=None, **kwargs*)

Bases: `shoop.core.order_creator.SourceLine`

shop_product

ShopProduct object of this line.

Return type `shoop.core.models.ShopProduct`

cache_info (*request*)

type

set_quantity (*quantity*)

can_delete

can_change_quantity

class `shoop.front.basket.objects.BaseBasket` (*request, basket_name='basket'*)

Bases: `shoop.core.order_creator.OrderSource`

save ()

Persist any changes made into the basket to storage.

One does not usually need to directly call this; `ShoopFrontMiddleware` will usually take care of it.

delete ()

Clear and delete the basket data.

finalize ()

Mark the basket as “completed” (i.e. an order is created/a conversion made).

This will also clear the basket’s data.

clear_all ()

Clear all data for this basket.

add_line (***kwargs*)

create_line (***kwargs*)

add_code (*code*)

clear_codes ()

remove_code (*code*)

get_lines ()

clean_empty_lines ()

add_product (*supplier, shop, product, quantity, force_new_line=False, extra=None, parent_line=None*)

update_line (*data_line, **kwargs*)

add_product_with_child_product (*supplier, shop, product, child_product, quantity*)

delete_line (*line_id*)

`find_line_by_line_id (line_id)`
`find_lines_by_parent_line_id (parent_line_id)`
`orderable`
`get_validation_errors ()`
`get_product_ids_and_quantities ()`
`get_available_shipping_methods ()`
Get available shipping methods.
 Return type list[ShippingMethod]
`get_available_payment_methods ()`
Get available payment methods.
 Return type list[PaymentMethod]

shoop.front.basket.order_creator module

class `shoop.front.basket.order_creator.BasketOrderCreator` (*request=None*)
Bases: `shoop.core.order_creator.OrderCreator`
Initialize order creator.
 Parameters **request** (*django.http.HttpRequest/None*) – Optional request object for backward compatibility. Passing non-None value is DEPRECATED.

shoop.front.basket.storage module

exception `shoop.front.basket.storage.BasketCompatibilityError`
Bases: `Exception`
exception `shoop.front.basket.storage.ShopMismatchBasketCompatibilityError`
Bases: `shoop.front.basket.storage.BasketCompatibilityError`
exception `shoop.front.basket.storage.PriceUnitMismatchBasketCompatibilityError`
Bases: `shoop.front.basket.storage.BasketCompatibilityError`
class `shoop.front.basket.storage.BasketStorage`
Bases: `object`
 load (*basket*)
 Load the given basket's data dictionary from the storage.
 Return type dict
 Raises `BasketCompatibilityError` if basket loaded from the storage is not compatible with the requested basket.
 save (*basket, data*)
 Save the given data dictionary into the storage for the given basket.

 delete (*basket*)
 Delete the basket from storage.

 finalize (*basket*)
 Mark the basket as “finalized” (i.e. completed) in the storage.
 The actual semantics of what finalization does are up to each backend.

```
class shoop.front.basket.storage.DirectSessionBasketStorage
```

Bases: *shoop.front.basket.storage.BasketStorage*

save (*basket*, *data*)

delete (*basket*)

```
class shoop.front.basket.storage.DictStoredBasket (id, shop_id, currency,
                                                    prices_include_tax, data)
```

Bases: *object*

classmethod from_basket_and_data (*basket*, *data*)

classmethod from_dict (*mapping*)

as_dict ()

```
class shoop.front.basket.storage.DatabaseBasketStorage
```

Bases: *shoop.front.basket.storage.BasketStorage*

save (*basket*, *data*)

delete (*basket*)

finalize (*basket*)

```
shoop.front.basket.storage.get_storage ()
```

Retrieve a basket storage object.

Returns A basket storage object

Return type *BasketStorage*

shoop.front.basket.update_methods module

```
class shoop.front.basket.update_methods.BasketUpdateMethods (request, basket)
```

Bases: *object*

get_prefix_to_method_map ()

Override this method to link prefixes with their associated methods to call.

Format of the dictionary is: { FIELD_NAME_PREFIX: METHOD }.

METHOD is a function which accepts the keyword arguments given in *update_basket_contents*. It should perform the necessary changes to the *basket_line* and then return whether the value had changed or not. (See *update_quantity* or *delete_line* for examples.)

delete_line (*line*, ***kwargs*)

update_quantity (*line*, *value*, ***kwargs*)

Module contents

```
shoop.front.basket.get_basket_order_creator (request=None)
```

```
shoop.front.basket.get_basket_view ()
```

```
shoop.front.basket.get_basket_command_dispatcher (request)
```

Return type *shoop.front.basket.command_dispatcher.BasketCommandDispatcher*

```
shoop.front.basket.get_basket (request)
```

Return type *shoop.front.basket.objects.BaseBasket*

shoop.front.checkout package

Submodules

shoop.front.checkout.addresses module

```
class shoop.front.checkout.addresses.AddressForm(**kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        model
            alias of MutableAddress

        fields = ('name', 'phone', 'email', 'street', 'street2', 'postal_code', 'city', 'region', 'country')

    AddressForm.base_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713b441d68>), ('p
    AddressForm.declared_fields = OrderedDict()

    AddressForm.media

class shoop.front.checkout.addresses.CompanyForm(data=None, files=None,
    auto_id='id_%s', prefix=None,
    initial=None, error_class=<class
    'django.forms.utils.ErrorList'>,
    label_suffix=None,
    empty_permitted=False, in-
    stance=None)

    Bases: shoop.front.checkout._mixins.TaxNumberCleanMixin,
    django.forms.models.ModelForm

    company_name_field = 'name'

    class Meta
        Bases: object

        model
            alias of CompanyContact

        fields = ('name', 'tax_number')

    CompanyForm.base_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f713c17d860>), ('t
    CompanyForm.declared_fields = OrderedDict()

    CompanyForm.media

class shoop.front.checkout.addresses.AddressesPhase(**kwargs)
    Bases: shoop.front.checkout.CheckoutPhaseViewMixin, django.views.generic.edit.FormView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    identifier = 'addresses'

    title = <django.utils.functional.lazy.<locals>.__proxy__ object>

    template_name = 'shoop/front/checkout/addresses.jinja'

    address_kinds = ('shipping', 'billing')

    address_form_class
        alias of AddressForm
```

```

address_form_classes = {}

company_form_class
    alias of CompanyForm

get_form(form_class=None)

get_initial()

is_valid()

form_valid(form)

process()

```

shoop.front.checkout.confirm module

```

class shoop.front.checkout.confirm.ConfirmForm(data=None, files=None,
        auto_id='id_%s', prefix=None, initial=None, error_class=<class
        'django.forms.utils.ErrorList'>, label_suffix=None, empty_permitted=False)

```

Bases: `django.forms.forms.Form`

base_fields = `OrderedDict([('accept_terms', <django.forms.fields.BooleanField object at 0x7f713a3b1080>), ('market`

declared_fields = `OrderedDict([('accept_terms', <django.forms.fields.BooleanField object at 0x7f713a3b1080>), ('m`

media

```

class shoop.front.checkout.confirm.ConfirmPhase(**kwargs)

```

Bases: `shoop.front.checkout.CheckoutPhaseViewMixin`, `django.views.generic.edit.FormView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

identifier = 'confirm'

title = <django.utils.functional.lazy.<locals>.__proxy__ object>

template_name = 'shoop/front/checkout/confirm.jinja'

form_class

alias of *ConfirmForm*

process()

is_valid()

get_context_data(**kwargs)

form_valid(form)

create_order()

shoop.front.checkout.empty module

```

class shoop.front.checkout.empty.EmptyPhase(**kwargs)

```

Bases: `shoop.front.checkout.CheckoutPhaseViewMixin`, `django.views.generic.base.TemplateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

identifier = 'empty'

title = <django.utils.functional.lazy.<locals>.__proxy__ object>

template_name = 'shoop/front/checkout/empty.jinja'

process()

shoop.front.checkout.methods module**class** shoop.front.checkout.methods.**MethodWidget** (*attrs=None, choices=()*)

Bases: django.forms.widgets.Widget

render (*name, value, attrs=None*)**media****class** shoop.front.checkout.methods.**MethodChoiceIterator** (*field*)

Bases: django.forms.models.ModelChoiceIterator

choice (*obj*)**class** shoop.front.checkout.methods.**MethodsForm** (**args, **kwargs*)

Bases: django.forms.forms.Form

limit_method_fields ()**base_fields** = OrderedDict([('shipping_method', <django.forms.models.ModelChoiceField object at 0x7f7139dd23c8>)**declared_fields** = OrderedDict([('shipping_method', <django.forms.models.ModelChoiceField object at 0x7f7139dd**media****class** shoop.front.checkout.methods.**MethodsPhase** (***kwargs*)Bases: *shoop.front.checkout.CheckoutPhaseViewMixin*, django.views.generic.edit.FormView

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

identifier = 'methods'**title** = <django.utils.functional.lazy.<locals>.__proxy__ object>**template_name** = 'shoop/front/checkout/methods.jinja'**form_class**alias of *MethodsForm***is_valid** ()**process** ()**get_form_kwargs** ()**form_valid** (*form*)**get_initial** ()**class** shoop.front.checkout.methods.**ShippingMethodPhase** (***kwargs*)

Bases: shoop.front.checkout.methods._MethodDependentCheckoutPhase, django.views.generic.base.View

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

identifier = 'shipping'**get_method** ()**class** shoop.front.checkout.methods.**PaymentMethodPhase** (***kwargs*)

Bases: shoop.front.checkout.methods._MethodDependentCheckoutPhase, django.views.generic.base.View

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

identifier = 'payment'**get_method** ()

shoop.front.checkout.single_page module

```

class shoop.front.checkout.single_page.OrderForm(*args, **kwargs)
    Bases: shoop.front.checkout._mixins.TaxNumberCleanMixin,
            django.forms.forms.Form
    limit_method_fields()
    base_fields = OrderedDict([('company_name', <django.forms.fields.CharField object at 0x7f713a472470>), ('tax_nun
    declared_fields = OrderedDict([('company_name', <django.forms.fields.CharField object at 0x7f713a472470>), ('ta
    media
class shoop.front.checkout.single_page.SingleCheckoutPhase(*kwargs)
    Bases: shoop.front.checkout.CheckoutPhaseViewMixin, django.views.generic.edit.FormView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    identifier = 'checkout'
    final = True
    template_name = 'shoop/front/checkout/single_phase.jinja'
    billing_address_form_class
        alias of AddressForm
    shipping_address_form_class
        alias of AddressForm
    order_form_class
        alias of OrderForm
    get_form(form_class=None)
    get_context_data(**kwargs)
    form_valid(form)
    process()

```

Module contents

```

class shoop.front.checkout.BasicServiceCheckoutPhaseProvider
    Bases: shoop.front.checkout.ServiceCheckoutPhaseProvider
    Helper for implementing basic ServiceCheckoutPhaseProvider.
    This helper should be useful for most cases, where one only has to provide a checkout phase for certain service
    provider type just by initializing some predefined class.
    get_checkout_phase(checkout_process, service)
        Get checkout phase for given service.
        Return type shoop.front.checkout.CheckoutPhaseViewMixin|None
    phase_class = None
    service_provider_class = None
class shoop.front.checkout.CheckoutPhaseViewMixin
    Bases: object
    checkout_process = None
    final = False
    get_success_url()

```

```
identifier = None
is_valid()
next_phase = None
phases = ()
previous_phase = None
process()
request = None
reset()
should_skip()
storage
title = None
```

```
class shoop.front.checkout.CheckoutProcess (phase_specs, phase_kwargs)
```

Bases: `object`

```
add_phase_attributes (target_phase, current_phase=None)
```

Add phase instance attributes (previous, next, etc) to the given target phase, using the optional `current_phase` as the current phase for previous and next.

This is exposed as a public API for the benefit of phases that need to do sub-phase initialization and dispatching, such as method phases.

```
complete()
```

To be called from a phase (`self.checkout_process.complete()`) when the checkout process is complete.

```
get_current_phase (requested_phase_identifier)
```

```
get_next_phase (current_phase, target_phase)
```

```
get_previous_phase (current_phase, target_phase)
```

```
instantiate_phase_class (phase_class, **extra_kwargs)
```

```
phases
```

Return type `Iterable[CheckoutPhaseViewMixin]`

```
prepare_current_phase (phase_identifier)
```

```
reset()
```

```
class shoop.front.checkout.ServiceCheckoutPhaseProvider
```

Bases: `object`

Interface for providing checkout phase for a service.

Items specified in `front_service_checkout_phase_provider` provide category should implement this interface.

```
get_checkout_phase (checkout_process, service)
```

Get checkout phase for given service.

If this provider is for another service, then the return value will be `None`.

Return type `shoop.front.checkout.CheckoutPhaseViewMixin|None`

shoop.front.models package**Submodules****shoop.front.models.stored_basket module**

`shoop.front.models.stored_basket.generate_key()`

class `shoop.front.models.stored_basket.StoredBasket` (*id, key, shop, customer, orderer, creator, created_on, updated_on, persistent, deleted, finished, title, data, taxless_total_price_value, taxful_total_price_value, currency, prices_include_tax, product_count*)

Bases: `shoop.utils.properties.MoneyPropped`, `django.db.models.base.Model`

shop

customer

orderer

creator

data

A placeholder class that provides a way to set the attribute on the model.

taxful_total_price

taxless_total_price

products

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `StoredBasket.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`StoredBasket.get_next_by_created_on(*moreargs, **morekwargs)`

`StoredBasket.get_next_by_updated_on(*moreargs, **morekwargs)`

`StoredBasket.get_previous_by_created_on(*moreargs, **morekwargs)`

`StoredBasket.get_previous_by_updated_on(*moreargs, **morekwargs)`

`StoredBasket.objects = <django.db.models.manager.Manager object>`

Module contents

class `shoop.front.models.StoredBasket` (*id, key, shop, customer, orderer, creator, created_on, updated_on, persistent, deleted, finished, title, data, taxless_total_price_value, taxful_total_price_value, currency, prices_include_tax, product_count*)

Bases: `shoop.utils.properties.MoneyPropped`, `django.db.models.base.Model`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `StoredBasket.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`StoredBasket.creator`

```
StoredBasket.customer
StoredBasket.data
    A placeholder class that provides a way to set the attribute on the model.
StoredBasket.get_next_by_created_on(*moreargs, **morekwargs)
StoredBasket.get_next_by_updated_on(*moreargs, **morekwargs)
StoredBasket.get_previous_by_created_on(*moreargs, **morekwargs)
StoredBasket.get_previous_by_updated_on(*moreargs, **morekwargs)
StoredBasket.objects = <django.db.models.manager.Manager object>
StoredBasket.orderer
StoredBasket.products
StoredBasket.shop
StoredBasket.taxful_total_price
StoredBasket.taxless_total_price
```

shoop.front.template_helpers package

Submodules

shoop.front.template_helpers.category module

```
shoop.front.template_helpers.category.get_manufacturers(context)
```

shoop.front.template_helpers.general module

```
shoop.front.template_helpers.general.get_visible_products(context, n_products,
                                                             ordering=None, filter_dict=None, orderable_only=True)
```

Parameters

- **context** (*django.template.context.Context*) – Rendering context
- **n_products** (*int*) – Number of products to return
- **ordering** (*str*) – String specifying ordering
- **filter_dict** (*dict[str, object]*) – Dictionary of filter parameters
- **orderable_only** (*bool*) – Boolean limiting results to orderable products

Return type list[shoop.core.models.Product]

```
shoop.front.template_helpers.general.get_best_selling_products(context,
                                                                n_products=12,
                                                                cutoff_days=30,
                                                                orderable_only=True)
```

```
shoop.front.template_helpers.general.get_newest_products(context, n_products=6,
                                                            orderable_only=True)
```

```
shoop.front.template_helpers.general.get_random_products(context, n_products=6,
                                                            orderable_only=True)
```

```
shoop.front.template_helpers.general.get_all_manufacturers(context)
```

```
shoop.front.template_helpers.general.get_root_categories(context)
```

```
shoop.front.template_helpers.general.get_pagination_variables(context, objects,
                                                             limit)
```

Get pagination variables for template

Parameters

- **context** – template context
- **objects** – objects paginated
- **limit** – per page limit

Returns variables to render object-list with pagination

shoop.front.template_helpers.product module

```
shoop.front.template_helpers.product.get_visible_attributes(product)
```

```
shoop.front.template_helpers.product.get_products_bought_with(context, product,
                                                             count=5)
```

```
shoop.front.template_helpers.product.is_visible(context, product)
```

```
shoop.front.template_helpers.product.get_product_cross_sells(context, product,
                                                             relation_type=<ProductCrossSellType.RELATION_2>,
                                                             count=4,
                                                             orderable_only=True)
```

```
shoop.front.template_helpers.product.map_relation_type(relation_type)
```

Map relation type to enum value.

Return type *ProductCrossSellType*

Raises `LookupError` if unknown string is given

shoop.front.template_helpers.urls module

```
shoop.front.template_helpers.urls.model_url(context, model, absolute=False)
```

```
shoop.front.template_helpers.urls.get_url(url, *args, **kwargs)
```

Try to get the reversed URL for the given route name, args and kwargs.

If reverse resolution fails, returns None (instead of throwing an exception).

Parameters

- **url** (*str*) – URL name.
- **args** (*Iterable[object]*) – URL args
- **kwargs** (*dict[str, object]*) – URL kwargs

Returns Reversed URL or None

Return type `str|None`

```
shoop.front.template_helpers.urls.has_url(url, *args, **kwargs)
```

Try to get the reversed URL for the given route name, args and kwargs and return a success flag.

Parameters

- **url** (*str*) – URL name.

- **args** (*Iterable[object]*) – URL args
- **kwargs** (*dict[str, object]*) – URL kwargs

Returns Success flag

Return type `bool`

Module contents

shoop.front.templatetags package

Submodules

shoop.front.templatetags.shoop_front module

class `shoop.front.templatetags.shoop_front.HelpersNamespace`

Bases: `object`

`shoop.front.templatetags.shoop_front.markdown` (*value*)

shoop.front.templatetags.thumbnails module

`shoop.front.templatetags.thumbnails.process_thumbnailer_options` (*kwargs*)

`shoop.front.templatetags.thumbnails.thumbnail` (*source*, *alias=None*, *generate=True*,
***kwargs*)

`shoop.front.templatetags.thumbnails.thumbnailer` (*source*)

Module contents

shoop.front.utils package

Submodules

shoop.front.utils.product_sorting module

`shoop.front.utils.product_sorting.sort_products` (*request*, *products*, *sort*)

shoop.front.utils.product_statistics module

`shoop.front.utils.product_statistics.get_best_selling_product_info` (*shop_ids*,
cut-
off_days=30)

`shoop.front.utils.product_statistics.get_products_ordered_with` (*prod*, *count=20*,
request=None,
lan-
guage=None)

`shoop.front.utils.product_statistics.get_products_by_brand` (*prod*, *count=6*,
request=None, *lan-*
guage=None)

```
shoop.front.utils.product_statistics.get_products_by_same_categories(prod,
                                                                    count=6,
                                                                    re-
                                                                    quest=None,
                                                                    lan-
                                                                    guage=None)
```

shoop.front.utils.views module

```
shoop.front.utils.views.cache_product_things(request, products, language=None, at-
                                             tribute_identifiers=('author', ))
```

Module contents

shoop.front.views package

Submodules

shoop.front.views.basket module

```
class shoop.front.views.basket.DefaultBasketView(**kwargs)
```

Bases: `django.views.generic.base.TemplateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
template_name = 'shoop/front/basket/default_basket.jinja'
```

```
get_context_data(**kwargs)
```

```
class shoop.front.views.basket.BasketView(**kwargs)
```

Bases: `django.views.generic.base.View`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
dispatch(request, *args, **kwargs)
```

shoop.front.views.category module

```
class shoop.front.views.category.ProductListForm(data=None,          files=None,
                                                  auto_id='id_%s',    prefix=None,
                                                  initial=None,       error_class=<class
                                                  'django.forms.utils.ErrorList'>,
                                                  label_suffix=None,
                                                  empty_permitted=False)
```

Bases: `django.forms.forms.Form`

```
base_fields = OrderedDict([('sort', <django.forms.fields.CharField object at 0x7f713e301748>), ('manufacturers', <d
```

```
declared_fields = OrderedDict([('sort', <django.forms.fields.CharField object at 0x7f713e301748>), ('manufacturers', <d
```

```
media
```

```
class shoop.front.views.category.CategoryView(**kwargs)
```

Bases: `django.views.generic.detail.DetailView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
template_name = 'shoop/front/product/category.jinja'
```

```
model
```

alias of `Category`

```
template_object_name = 'category'

get_queryset ()

get_context_data (**kwargs)
```

shoop.front.views.checkout module

```
class shoop.front.views.checkout.BaseCheckoutView (**kwargs)
    Bases: django.views.generic.base.View

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    phase_specs = []

    empty_phase_spec = None

    dispatch (request, *args, **kwargs)
```

shoop.front.views.index module

```
class shoop.front.views.index.IndexView (**kwargs)
    Bases: django.views.generic.base.TemplateView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    template_name = 'shoop/front/index.jinja'

    get_context_data (**kwargs)
```

shoop.front.views.order module

```
class shoop.front.views.order.OrderCompleteView (**kwargs)
    Bases: django.views.generic.detail.DetailView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    template_name = 'shoop/front/order/complete.jinja'

    model
        alias of Order

    context_object_name = 'order'

    render_to_response (context, **response_kwargs)

    get_object (queryset=None)

class shoop.front.views.order.OrderRequiresVerificationView (**kwargs)
    Bases: django.views.generic.detail.DetailView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

    template_name = 'shoop/front/order/requires_verification.jinja'

    model
        alias of Order

    get_object (queryset=None)

    get_context_data (**kwargs)

    get (request, **kwargs)
```


shoop.front.views.payment module

`shoop.front.views.payment.get_payment_urls(request, order)`

class `shoop.front.views.payment.ProcessPaymentView(**kwargs)`

Bases: `django.views.generic.detail.DetailView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of `Order`

context_object_name = `'order'`

get_object (`queryset=None`)

get_context_data (`**kwargs`)

dispatch (`request, *args, **kwargs`)

shoop.front.views.product module

class `shoop.front.views.product.ProductDetailView(**kwargs)`

Bases: `django.views.generic.detail.DetailView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

template_name = `'shoop/front/product/detail.jinja'`

model

alias of `Product`

context_object_name = `'product'`

get_queryset ()

get_context_data (`**kwargs`)

get (`request, *args, **kwargs`)

Module contents**Submodules****shoop.front.error_handling module**

`shoop.front.error_handling.make_error_view(status, template_name=None)`

`shoop.front.error_handling.install_error_handlers()`

Install custom error handlers.

Error handlers to be added are for errors 400, 403, 404, and 500.

Error handlers will be injected only if: * `settings.SHOOP_FRONT_INSTALL_ERROR_HANDLERS` is `True` * `settings.ROOT_URLCONF` doesn't already contain the handler

shoop.front.middleware module

`shoop.front.middleware.ProblemMiddleware`

alias of `ExceptionMiddleware`

```
class shoop.front.middleware.ShoopFrontMiddleware
```

```
    Bases: object
```

Handle Shoop specific tasks for each request and response.

- Set request attributes that rest of the Shoop front-end rely on.
- Set Django's timezone according to personal preferences (i.e. `request.person.timezone`).

Todo

Fallback to shop timezone?

- Make sure that basket is saved before response is returned to the browser.

Attributes set for requests:

```
request.shop [shoop.core.models.Shop] Currently active Shop.
```

Todo

Define better

```
request.person [shoop.core.models.Contact] PersonContact of currently  
logged in user or AnonymousContact if there is no logged in user.
```

```
request.customer [shoop.core.models.Contact] Customer contact used when creat-  
ing Orders. This can be same as request.person, but for example in B2B shops this is  
usually a CompanyContact whereas request.person is a PersonContact.
```

```
request.basket [shoop.front.basket.objects.BaseBasket] Shopping basket in  
use.
```

```
process_request (request)
```

```
process_response (request, response)
```

```
classmethod refresh_on_user_change (request, user=None, **kwargs)
```

```
classmethod refresh_on_logout (request, **kwargs)
```

```
process_view (request, view_func, *view_args, **view_kwargs)
```

shoop.front.notify_events module

```
class shoop.front.notify_events.OrderReceived (**variable_values)
```

```
    Bases: shoop.notify.base.Event
```

```
    identifier = 'order_received'
```

```
    bindings = {}
```

```
    name = 'Order Received'
```

```
    variables = {'customer_email': <shoop.notify.base.Variable object at 0x7f7138572978>, 'order': <shoop.notify.base.Va
```

```
shoop.front.notify_events.send_order_received_notification (order, **kwargs)
```

shoop.front.settings module

`shoop.front.settings.SHOOP_BASKET_ORDER_CREATOR_SPEC = 'shoop.front.basket.order_creator:BasketOrderCreator'`
 Spec string for the class used for creating Order from a Basket.

This is the easiest way to customize the order creation process without having to override a single URL or touch the `shoop.front` code.

`shoop.front.settings.SHOOP_BASKET_VIEW_SPEC = 'shoop.front.views.basket:DefaultBasketView'`
 Spec string for the Django CBV (or an API-compliant class) for the basket view.

This view deals with `/basket/`.

`shoop.front.settings.SHOOP_BASKET_COMMAND_DISPATCHER_SPEC = 'shoop.front.basket.command_dispatcher:BasketCommandDispatcher'`
 Spec string for the command dispatcher used when products are added/deleted/etc. from the basket.

This view deals with commands POST`ed to ```/basket/`.

`shoop.front.settings.SHOOP_BASKET_UPDATE_METHODS_SPEC = 'shoop.front.basket.update_methods:BasketUpdateMethods'`
 Spec string for the update method dispatcher used when the basket is updated (usually on the basket page).

`shoop.front.settings.SHOOP_BASKET_CLASS_SPEC = 'shoop.front.basket.objects:BaseBasket'`
 Spec string for the basket class used in the frontend.

This is used to customize the behavior of the basket for a given installation, for instance to modify prices of products based on certain conditions, etc.

`shoop.front.settings.SHOOP_BASKET_STORAGE_CLASS_SPEC = 'shoop.front.basket.storage:DatabaseBasketStorage'`
 The spec string defining which basket storage class to use for the frontend.

Basket storages are responsible for persisting visitor basket state, whether in the database (`DatabaseBasketStorage`) or directly in the session (`DirectSessionBasketStorage`). Custom storage backends could use caches, flat files, etc. if required.

`shoop.front.settings.SHOOP_CHECKOUT_VIEW_SPEC = 'shoop.front.views.checkout:DefaultCheckoutView'`
 Spec string for the Django CBV (or an API-compliant class) for the checkout view.

This is used to customize the behavior of the checkout process; most likely to switch in a view with a different `phase_specs`.

`shoop.front.settings.SHOOP_FRONT_INSTALL_ERROR_HANDLERS = True`
 Whether Shoop uses its own error handlers.

If this value is set to `False` django defaults are used or the ones specified in `settings.ROOT_URLCONF` file.

Setting this to `True` won't override handlers specified in `settings.ROOT_URLCONF`.

Handled error cases are: 400, 403, 404, and 500

`shoop.front.settings.SHOOP_FRONT_ADDRESS_FIELD_PROPERTIES = {}`

A dictionary defining properties to override the default field properties of the checkout address form. Should map the field name (as a string) to a dictionary containing the overriding Django form field properties, as in the following example which makes the postal code a required field:

```
SHOOP_FRONT_ADDRESS_FIELD_PROPERTIES = { "postal_code": {"required": True}
```

Definition list ends without a blank line; unexpected unindent.

```
}
```

It should be noted, however, that overriding some settings (such as making a required field non-required) could create other validation issues.

shoop.front.signals module

shoop.front.urls module

Module contents

```
class shoop.front.ShoopFrontAppConfig(*args, **kwargs)
    Bases: shoop.apps.AppConfig

    name = 'shoop.front'

    verbose_name = 'Shoop Frontend'

    label = 'shoop_front'

    provides = {'admin_module': ['shoop.front.admin_module.CartAdminModule'], 'notify_event': ['shoop.front.notify_e

    ready()
```

shoop.guide package

Submodules

shoop.guide.admin_module module

```
class shoop.guide.admin_module.GuideAdminModule
    Bases: shoop.admin.base.AdminModule

    get_search_results(request, query)
```

shoop.guide.apps module

```
class shoop.guide.apps.AppConfig(*args, **kwargs)
    Bases: shoop.apps.AppConfig

    name = 'shoop.guide'

    verbose_name = 'Shoop Guide'

    provides = {'admin_module': ['shoop.guide.admin_module:GuideAdminModule']}
```

shoop.guide.settings module

```
shoop.guide.settings.SHOOP_GUIDE_API_URL = 'https://readthedocs.org/api/v2/search/?project=shoop-guide&version=4.0.0'
    ReadtheDocs API URL

    URL for fetching search results via ReadtheDocs API.

shoop.guide.settings.SHOOP_GUIDE_LINK_URL = 'http://shoop-guide.readthedocs.io/en/latest/search.html?check_key=shoop-guide'
    ReadtheDocs link URL

    URL for manually linking search query link. Query parameters are added to end of URL when constructing
    link.

shoop.guide.settings.SHOOP_GUIDE_FETCH_RESULTS = True
    Whether or not to fetch search results from ReadtheDocs

    If true, fetch results via the ReadtheDocs API, otherwise only display a link to RTD search page.

shoop.guide.settings.SHOOP_GUIDE_TIMEOUT_LIMIT = 2
    Timeout limit for fetching search results

    Time limit in seconds before a search result request should timeout, so as not to block search results in case of
    slow response.
```

Module contents

shoop.notify package

Subpackages

shoop.notify.actions package

Submodules

shoop.notify.actions.debug module

```
class shoop.notify.actions.debug.SetDebugFlag (data, validate=True)
    Bases: shoop.notify.base.Action

    identifier = 'set_debug_flag'

    execute (context)

    bindings = {'flag_name': <shoop.notify.base.Binding object at 0x7f713b0c6908>}

    name = 'Set Debug Flag'

    variables = {}
```

shoop.notify.actions.email module

```
class shoop.notify.actions.email.SendEmail (data, validate=True)
    Bases: shoop.notify.base.Action

    identifier = 'send_email'

    template_use = <TemplateUse.MULTILINGUAL: 2>

    template_fields = {'body': <django.forms.fields.CharField object at 0x7f7139c35be0>, 'subject': <django.forms.field

    execute (context)

        Parameters context (shoop.notify.script.Context) – Script Context

    bindings = {'recipient': <shoop.notify.base.Binding object at 0x7f7139c351d0>, 'language': <shoop.notify.base.Binding

    name = 'Send Email'

    variables = {}
```

shoop.notify.actions.notification module

```
class shoop.notify.actions.notification.AddNotification (data, validate=True)
    Bases: shoop.notify.base.Action

    identifier = 'add_notification'

    execute (context)

    bindings = {'message_identifier': <shoop.notify.base.Binding object at 0x7f713ba1f940>, 'priority': <shoop.notify.base.

    name = 'Add Notification'

    variables = {}
```

shoop.notify.actions.order module

```
class shoop.notify.actions.order.AddOrderLogEntry (data, validate=True)
    Bases: shoop.notify.base.Action

    identifier = 'add_order_log_entry'

    execute (context)

    bindings = {'message_identifier': <shoop.notify.base.Binding object at 0x7f713ba06438>, 'order': <shoop.notify.base.B...
    name = 'Add Order Log Entry'
    variables = {}
```

Module contents

```
class shoop.notify.actions.AddNotification (data, validate=True)
    Bases: shoop.notify.base.Action

    bindings = {'message_identifier': <shoop.notify.base.Binding object at 0x7f713ba1f940>, 'priority': <shoop.notify.base.B...
    execute (context)

    identifier = 'add_notification'
    name = 'Add Notification'
    variables = {}

class shoop.notify.actions.AddOrderLogEntry (data, validate=True)
    Bases: shoop.notify.base.Action

    bindings = {'message_identifier': <shoop.notify.base.Binding object at 0x7f713ba06438>, 'order': <shoop.notify.base.B...
    execute (context)

    identifier = 'add_order_log_entry'
    name = 'Add Order Log Entry'
    variables = {}

class shoop.notify.actions.SendEmail (data, validate=True)
    Bases: shoop.notify.base.Action

    bindings = {'recipient': <shoop.notify.base.Binding object at 0x7f7139c351d0>, 'language': <shoop.notify.base.Binding...
    execute (context)

        Parameters context (shoop.notify.script.Context) – Script Context

    identifier = 'send_email'
    name = 'Send Email'
    template_fields = {'body': <django.forms.fields.CharField object at 0x7f7139c35be0>, 'subject': <django.forms.field...
    template_use = <TemplateUse.MULTILINGUAL: 2>
    variables = {}

class shoop.notify.actions.SetDebugFlag (data, validate=True)
    Bases: shoop.notify.base.Action

    bindings = {'flag_name': <shoop.notify.base.Binding object at 0x7f713b0c6908>}

    execute (context)
```

```

identifier = 'set_debug_flag'
name = 'Set Debug Flag'
variables = {}

```

shoop.notify.admin_module package

Subpackages

shoop.notify.admin_module.views package

Submodules

shoop.notify.admin_module.views.edit module

class shoop.notify.admin_module.views.edit.**ScriptEditView** (**kwargs)

Bases: *shoop.admin.utils.views.CreateOrUpdateView*

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of *Script*

form_class

alias of *ScriptForm*

template_name = 'notify/admin/edit_script.jinja'

context_object_name = 'script'

get_context_data (**kwargs)

form_valid (form)

shoop.notify.admin_module.views.editor module

shoop.notify.admin_module.views.editor.**script_item_editor** (request)

class shoop.notify.admin_module.views.editor.**ScriptAPI** (request, script)

Bases: *object*

Parameters

- **request** (*django.http.HttpRequest*) – Request
- **script** (*shoop.notify.models.Script*) – Script

dispatch ()

handle_get_data (data)

handle_save_data (data)

class shoop.notify.admin_module.views.editor.**EditScriptContentView** (**kwargs)

Bases: *django.views.generic.detail.DetailView*

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

template_name = 'notify/admin/script_content_editor.jinja'

model

alias of *Script*

```
context_object_name = 'script'

post (request, *args, **kwargs)

get_context_data (**kwargs)
```

shoop.notify.admin_module.views.list module

```
class shoop.notify.admin_module.views.list.ScriptListView (**kwargs)
```

Bases: *shoop.admin.utils.views.PicortableListView*

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of *Script*

columns = [*<shoop.admin.utils.picortable.Column object at 0x7f713b9f6f60>*, *<shoop.admin.utils.picortable.Column object at 0x7f713b9f6f60>*]

get_object_url (*instance*)

get_event_identifier_text (*instance*)

get_toolbar ()

get_object_abstract (*instance, item*)

Module contents

```
shoop.notify.admin_module.views.script_item_editor (request)
```

```
class shoop.notify.admin_module.views.ScriptEditView (**kwargs)
```

Bases: *shoop.admin.utils.views.CreateOrUpdateView*

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

context_object_name = 'script'

form_class

alias of *ScriptForm*

form_valid (*form*)

get_context_data (**kwargs)

model

alias of *Script*

template_name = 'notify/admin/edit_script.jinja'

```
class shoop.notify.admin_module.views.EditScriptContentView (**kwargs)
```

Bases: *django.views.generic.detail.DetailView*

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

context_object_name = 'script'

get_context_data (**kwargs)

model

alias of *Script*

post (*request, *args, **kwargs*)

template_name = 'notify/admin/script_content_editor.jinja'


```
class shoop.notify.admin_module.views.ScriptListView(**kwargs)
```

Bases: *shoop.admin.utils.views.PicortableListView*

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

columns = [*<shoop.admin.utils.picotable.Column object at 0x7f713b9f6f60>*, *<shoop.admin.utils.picotable.Column object at 0x7f713b9f6f60>*]

get_event_identifier_text (*instance*)

get_object_abstract (*instance, item*)

get_object_url (*instance*)

get_toolbar ()

model

alias of *Script*

Submodules

shoop.notify.admin_module.forms module

```
class shoop.notify.admin_module.forms.ScriptForm(**kwargs)
```

Bases: *django.forms.models.ModelForm*

```
class Meta
```

Bases: *object*

model

alias of *Script*

fields = ('event_identifier', 'name', 'enabled')

ScriptForm.**base_fields** = *OrderedDict*([('event_identifier', *<django.forms.fields.ChoiceField object at 0x7f713b82f60>*), ('name', *<django.forms.fields.CharField object at 0x7f713b82f60>*), ('enabled', *<django.forms.fields.BooleanField object at 0x7f713b82f60>*)])

ScriptForm.**declared_fields** = *OrderedDict*([('event_identifier', *<django.forms.fields.ChoiceField object at 0x7f713b82f60>*), ('name', *<django.forms.fields.CharField object at 0x7f713b82f60>*), ('enabled', *<django.forms.fields.BooleanField object at 0x7f713b82f60>*)])

ScriptForm.**media**

```
class shoop.notify.admin_module.forms.ScriptItemEditForm(*args, **kwargs)
```

Bases: *django.forms.forms.Form*

populate_form ()

get_initial ()

save ()

base_fields = *OrderedDict*()

declared_fields = *OrderedDict*()

media

shoop.notify.admin_module.utils module

```
shoop.notify.admin_module.utils.get_name_map(category_key)
```

```
shoop.notify.admin_module.utils.get_enum_choices_dict(enum_class)
```

Module contents

```
class shoop.notify.admin_module.NotifyAdminModule
```

Bases: *shoop.admin.base.AdminModule*

name = *<django.utils.functional.lazy.<locals>.__proxy__ object>*

```
breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>
get_urls ()
get_menu_category_icons ()
get_menu_entries (request)
get_required_permissions ()
mark_notification_read_view (request, pk)
get_notifications (request)
get_model_url (object, kind)
```

shoop.notify.conditions package

Submodules

shoop.notify.conditions.simple module

```
class shoop.notify.conditions.simple.NonEmpty (data, validate=True)
    Bases: shoop.notify.base.Condition
    identifier = 'non_empty'
    description = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    test (context)
    bindings = {'v': <shoop.notify.base.Binding object at 0x7f7138fcb668>}
    variables = {}
class shoop.notify.conditions.simple.Empty (data, validate=True)
    Bases: shoop.notify.base.Condition
    identifier = 'empty'
    description = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    test (context)
    bindings = {'v': <shoop.notify.base.Binding object at 0x7f7138770a20>}
    variables = {}
class shoop.notify.conditions.simple.BaseEqual (data, validate=True)
    Bases: shoop.notify.base.Condition
    identifier_suffix = 'equal'
    test (context)
    bindings = {}
    identifier = 'base_equal'
    name = 'Base Equal'
    variables = {}
```

```

class shoop.notify.conditions.simple.CaseInsensitiveStringEqual (data, validate=True)
    Bases: shoop.notify.base.Condition
    identifier_suffix = 'equal'
    test (context)
    bindings = {}
    identifier = 'case_insensitive_string_equal'
    name = 'Case Insensitive String Equal'
    variables = {}

shoop.notify.conditions.simple.construct_simple (base, var_type)

```

Module contents

```

class shoop.notify.conditions.BooleanEqual (data, validate=True)
    Bases: shoop.notify.conditions.simple.BaseEqual
    bindings = {'v1': <shoop.notify.base.Binding object at 0x7f713af4a780>, 'v2': <shoop.notify.base.Binding object at 0x7f713af4a780>}
    identifier = 'boolean_equal'
    name = 'Boolean Equal'
    variables = {}

class shoop.notify.conditions.Empty (data, validate=True)
    Bases: shoop.notify.base.Condition
    bindings = {'v': <shoop.notify.base.Binding object at 0x7f7138770a20>}
    description = <django.utils.functional.lazy.<locals>.__proxy__ object>
    identifier = 'empty'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    test (context)
    variables = {}

class shoop.notify.conditions.IntegerEqual (data, validate=True)
    Bases: shoop.notify.conditions.simple.BaseEqual
    bindings = {'v1': <shoop.notify.base.Binding object at 0x7f713897fbc0>, 'v2': <shoop.notify.base.Binding object at 0x7f713897fbc0>}
    identifier = 'integer_equal'
    name = 'Integer Equal'
    variables = {}

class shoop.notify.conditions.LanguageEqual (data, validate=True)
    Bases: shoop.notify.conditions.simple.CaseInsensitiveStringEqual
    bindings = {'v1': <shoop.notify.base.Binding object at 0x7f7138d1fdd8>, 'v2': <shoop.notify.base.Binding object at 0x7f7138d1fdd8>}
    identifier = 'language_equal'
    name = 'Language Equal'
    variables = {}

class shoop.notify.conditions.NonEmpty (data, validate=True)
    Bases: shoop.notify.base.Condition

```

```
bindings = {'v': <shoop.notify.base.Binding object at 0x7f7138fcb668>}
description = <django.utils.functional.lazy.<locals>.__proxy__ object>
identifier = 'non_empty'
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
test (context)
variables = {}

class shoop.notify.conditions.TextEqual (data, validate=True)
    Bases: shoop.notify.conditions.simple.CaseInsensitiveStringEqual

    bindings = {'v1': <shoop.notify.base.Binding object at 0x7f713891a278>, 'v2': <shoop.notify.base.Binding object at 0x7f713891a278>}
    identifier = 'text_equal'
    name = 'Text Equal'
    variables = {}
```

shoop.notify.models package

Submodules

shoop.notify.models.notification module

```
class shoop.notify.models.notification.NotificationManager
    Bases: django.db.models.manager.Manager

    for_user (user)

    unread_for_user (user)

class shoop.notify.models.notification.Notification (*args, **kwargs)
    Bases: django.db.models.base.Model

    A model for persistent notifications to be shown in the admin, etc.

    recipient_type
        A placeholder class that provides a way to set the attribute on the model.

    recipient

    priority
        A placeholder class that provides a way to set the attribute on the model.

    marked_read_by

    objects = <shoop.notify.models.notification.NotificationManager object>

    save (*args, **kwargs)

    mark_read (user)

    is_read

    data

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist
```

```

exception Notification.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned
Notification.get_next_by_created_on (*moreargs, **morekwargs)
Notification.get_previous_by_created_on (*moreargs, **morekwargs)
Notification.get_priority_display (*moreargs, **morekwargs)
Notification.get_recipient_type_display (*moreargs, **morekwargs)
Notification.url
Notification.set_reverse_url (**reverse_kwargs)

```

shoop.notify.models.script module

```

class shoop.notify.models.script.Script (id, event_identifier, identifier, created_on, name, enabled, _step_data)
    Bases: django.db.models.base.Model
    get_steps ()
        :rtype Iterable[Step]
    set_steps (steps)
    get_serialized_steps ()
    set_serialized_steps (serialized_data)
    event_class
    execute (context)
        Execute the script in the given context.

        Parameters context (shoop.notify.script.Context) – Script context
exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist
exception Script.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned
Script.get_next_by_created_on (*moreargs, **morekwargs)
Script.get_previous_by_created_on (*moreargs, **morekwargs)
Script.objects = <django.db.models.manager.Manager object>

```

Module contents

```

class shoop.notify.models.Notification (*args, **kwargs)
    Bases: django.db.models.base.Model
    A model for persistent notifications to be shown in the admin, etc.
exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist
exception Notification.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned
Notification.data
Notification.get_next_by_created_on (*moreargs, **morekwargs)
Notification.get_previous_by_created_on (*moreargs, **morekwargs)

```

```
Notification.get_priority_display(*moreargs, **morekwargs)
Notification.get_recipient_type_display(*moreargs, **morekwargs)
Notification.is_read
Notification.mark_read(user)
Notification.marked_read_by
Notification.objects = <shoop.notify.models.notification.NotificationManager object>
Notification.priority
    A placeholder class that provides a way to set the attribute on the model.
Notification.recipient
Notification.recipient_type
    A placeholder class that provides a way to set the attribute on the model.
Notification.save(*args, **kwargs)
Notification.set_reverse_url(**reverse_kwargs)
Notification.url
class shoop.notify.models.Script(id, event_identifier, identifier, created_on, name, enabled,
                                _step_data)
    Bases: django.db.models.base.Model
    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist
    exception Script.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned
    Script.event_class
    Script.execute(context)
        Execute the script in the given context.
        Parameters context (shoop.notify.script.Context) – Script context
    Script.get_next_by_created_on(*moreargs, **morekwargs)
    Script.get_previous_by_created_on(*moreargs, **morekwargs)
    Script.get_serialized_steps()
    Script.get_steps()
        :rtype Iterable[Step]
    Script.objects = <django.db.models.manager.Manager object>
    Script.set_serialized_steps(serialized_data)
    Script.set_steps(steps)
```

Submodules

shoop.notify.base module

```
class shoop.notify.base.BaseMetaclass
    Bases: type
class shoop.notify.base.Variable(name, type=<class 'shoop.notify.typology.Type'>, re-
                                quired=True, help_text='')
    Bases: object
```

```

    get_matching_types (variable_dict)

class shoop.notify.base.Binding (name, type=<class 'shoop.notify.typology.Type'>,
                                required=False, help_text='', constant_use=<ConstantUse.VARIABLE_ONLY: 1>, default=None)
    Bases: shoop.notify.base.Variable

    accepts_any_type

    allow_constant

    allow_variable

    get_value (context, bind_data)

class shoop.notify.base.TemplatedBinding (*args, **kwargs)
    Bases: shoop.notify.base.Binding

    get_value (context, bind_data)

class shoop.notify.base.Base
    Bases: object

    identifier = None

    name = None

    description = None

    variables = {}

    bindings = {}

    provide_category = None

    classmethod class_for_identifier (identifier)

class shoop.notify.base.Event (**variable_values)
    Bases: shoop.notify.base.Base

    provide_category = 'notify_event'

    identifier = None

    log_target_variable = None
        The name of the variable to be used as the log target for this event.

        The target variable must have an add_log_entry method.

    log_target

    load_variables (variable_values)

    run ()

    bindings = {}

    variables = {}

class shoop.notify.base.ScriptItem (data, validate=True)
    Bases: shoop.notify.base.Base

    provide_category = None

    verify_bindings ()

    get_value (context, binding_name)
        Get the actual value of a binding from the given script context.

```

Parameters

- **context** (`shoop.notify.script.Context`) – Script Context
- **binding_name** (`str`) – Binding name.

Returns The variable value

get_values (`context`)

Get all binding values in a dict.

Parameters **context** (`shoop.notify.script.Context`) – Script Context

Returns Dict of binding name -> value

Return type dict[name, value]

classmethod **unserialize** (`data, validate=True`)

serialize ()

classmethod **get_ui_info_map** ()

bindings = {}

identifier = 'script_item'

name = 'Script Item'

variables = {}

class `shoop.notify.base.Condition` (`data, validate=True`)

Bases: `shoop.notify.base.ScriptItem`

provide_category = 'notify_condition'

test (`context`)

bindings = {}

identifier = 'condition'

name = 'Condition'

variables = {}

class `shoop.notify.base.Action` (`data, validate=True`)

Bases: `shoop.notify.base.ScriptItem`

provide_category = 'notify_action'

template_use = <TemplateUse.NONE: 0>

template_fields = {}

execute (`context`)

Parameters **context** (`shoop.notify.script.Context`) – Script Context

get_template (`context`)

Get this action's template instance, bound in the context.

Return type `shoop.notify.template.Template`

bindings = {}

get_template_values (`context, language_preferences=()`)

Render this Action's template with data from the given context.

Parameters

- **context** (`shoop.notify.script.Context`) – Script Context
- **language_preferences** (`list[str]`) – Language preference list. The first language in the template to have values for all fields will be used. Has no effect for UNILINGUAL template_use.

Returns Dict of field name -> rendered template text.

Return type dict[str, str]|None

```

identifier = 'action'
name = 'Action'
variables = {}

```

shoop.notify.enums module

class `shoop.notify.enums.TemplateUse`

Bases: `enumfields.enums.Enum`

An enumeration.

NONE = `<TemplateUse.NONE: 0>`

UNILINGUAL = `<TemplateUse.UNILINGUAL: 1>`

MULTILINGUAL = `<TemplateUse.MULTILINGUAL: 2>`

class `shoop.notify.enums.ConstantUse`

Bases: `enumfields.enums.Enum`

An enumeration.

VARIABLE_ONLY = `<ConstantUse.VARIABLE_ONLY: 1>`

CONSTANT_ONLY = `<ConstantUse.CONSTANT_ONLY: 2>`

VARIABLE_OR_CONSTANT = `<ConstantUse.VARIABLE_OR_CONSTANT: 3>`

class `shoop.notify.enums.StepNext`

Bases: `enumfields.enums.Enum`

An enumeration.

CONTINUE = `<StepNext.CONTINUE: 'continue'>`

STOP = `<StepNext.STOP: 'stop'>`

class `shoop.notify.enums.StepConditionOperator`

Bases: `enumfields.enums.Enum`

An enumeration.

ALL = `<StepConditionOperator.ALL: 'all'>`

ANY = `<StepConditionOperator.ANY: 'any'>`

NONE = `<StepConditionOperator.NONE: 'none'>`

class `shoop.notify.enums.RecipientType`

Bases: `enumfields.enums.Enum`

An enumeration.

ADMINS = `<RecipientType.ADMINS: 1>`

SPECIFIC_USER = `<RecipientType.SPECIFIC_USER: 2>`

```
class shoop.notify.enums.Priority
    Bases: enumfields.enums.Enum

    An enumeration.

    LOW = <Priority.LOW: 1>
    NORMAL = <Priority.NORMAL: 2>
    HIGH = <Priority.HIGH: 3>
    CRITICAL = <Priority.CRITICAL: 4>
```

shoop.notify.runner module

```
shoop.notify.runner.run_event(event)
```

shoop.notify.script module

```
shoop.notify.script.none(conditions)
```

```
class shoop.notify.script.Step(conditions=(), actions=(), next=<StepNext.CONTINUE: 'continue'>, cond_op=<StepConditionOperator.ALL: 'all'>, enabled=True)
```

Bases: `object`

execute(context)

serialize()

classmethod unserialize(step_data)

enabled

```
class shoop.notify.script.Context(variables=None)
```

Bases: `object`

classmethod from_variables(**variables)

Create Context from variables.

Return type *shoop.notify.script.Context*

classmethod from_event(event)

Create Context from event.

Return type *shoop.notify.script.Context*

get(name, default=None)

set(name, value)

get_variables()

log(level, msg, *args, **kwargs)

Log a message with the context's logger (not the log target). This may be an useful debugging tool.

The parameters are the same as for `logging.Logger.log()`.

add_log_entry_on_log_target(message, identifier, **kwargs)

Add a log entry on the context's log target.

The kwargs are passed to the target's `add_log_entry` method.

If no log target exists or if it has no `add_log_entry` method, this method does nothing.

Parameters

- **message** (*str*) – The message text.

- **identifier** (*str*) – The message identifier. Unlike with `add_log_entry`, this is required.
- **kwargs** (*dict*) – Other kwargs to pass to `add_log_entry`

`log_entry_queryset`

shoop.notify.settings module

shoop.notify.template module

exception `shoop.notify.template.NoLanguageMatches`

Bases: `Exception`

`shoop.notify.template.render_in_context` (*context*, *template_text*, *html_intent=False*)

Render the given Jinja2 template text in the script context.

Parameters

- **context** (`shoop.notify.script.Context`) – Script context.
- **template_text** (*str*) – Jinja2 template text.
- **html_intent** (*bool*) – Is the template text intended for HTML output? This currently turns on autoescaping.

Returns Rendered template text

Return type `str`

Raises Whatever Jinja2 might happen to raise

class `shoop.notify.template.Template` (*context*, *data*)

Bases: `object`

Parameters

- **context** (`shoop.notify.script.Context`) – Script context
- **data** (*dict*) – Template data dictionary

has_language (*language*, *fields*)

render (*language*, *fields*)

Render this template in the given language, returning the given fields.

Parameters

- **language** (*str*) – Language code (ISO 639-1 or ISO 639-2)
- **fields** (*list[str]*) – Desired fields to render.

Returns Dict of field -> rendered content.

Return type `dict[str, str]`

render_first_match (*language_preferences*, *fields*)

shoop.notify.typology module

class `shoop.notify.typology.Type`

Bases: `object`

name = `None`

identifier = `None`

get_field (**kwargs)

Get a Django form field for this type.

The kwargs are passed directly to the field constructor.

Parameters **kwargs** (*dict*) – Kwargs for field constructor

Returns Form field

Return type `django.forms.Field`

unserialize (*value*)

validate (*value*)

is_coercible_from (*other_type*)

class `shoop.notify.typology.Boolean`

Bases: `shoop.notify.typology.Type`

name = `<django.utils.functional.lazy.<locals>.__proxy__ object>`

identifier = `'boolean'`

class `shoop.notify.typology.Integer`

Bases: `shoop.notify.typology._Number`

name = `<django.utils.functional.lazy.<locals>.__proxy__ object>`

identifier = `'integer'`

get_field (**kwargs)

class `shoop.notify.typology.Decimal`

Bases: `shoop.notify.typology._Number`

name = `<django.utils.functional.lazy.<locals>.__proxy__ object>`

identifier = `'decimal'`

get_field (**kwargs)

class `shoop.notify.typology.Text`

Bases: `shoop.notify.typology._String`

name = `<django.utils.functional.lazy.<locals>.__proxy__ object>`

identifier = `'text'`

is_coercible_from (*other_type*)

class `shoop.notify.typology.Language`

Bases: `shoop.notify.typology._String`

name = `<django.utils.functional.lazy.<locals>.__proxy__ object>`

identifier = `'language'`

class `shoop.notify.typology.Email`

Bases: `shoop.notify.typology._String`

name = `<django.utils.functional.lazy.<locals>.__proxy__ object>`

identifier = `'email'`

get_field (**kwargs)

class `shoop.notify.typology.URL`

Bases: `shoop.notify.typology._String`

```

    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    identifier = 'url'
    get_field (**kwargs)

class shoop.notify.typology.Phone
    Bases: shoop.notify.typology._String

    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    identifier = 'phone'

class shoop.notify.typology.Model(model_label)
    Bases: shoop.notify.typology.Type

    Parameters model_label (str) – Model label in Django app.Model format (e.g.
        shoop.Order)

    identifier = 'model'

    name

    model_label = None

    unserialize (value)

    is_coercible_from (other_type)

    get_model ()

    Return type django.db.models.Model

    get_field (**kwargs)

class shoop.notify.typology.Enum(enum_class)
    Bases: shoop.notify.typology.Type

    identifier = 'enum'

    name

    enum_class = None

    unserialize (value)

    get_field (**kwargs)

```

Module contents

```

class shoop.notify.Action(data, validate=True)
    Bases: shoop.notify.base.ScriptItem

    bindings = {}

    execute (context)

    Parameters context (shoop.notify.script.Context) – Script Context

    get_template (context)
        Get this action's template instance, bound in the context.

    Return type shoop.notify.template.Template

    get_template_values (context, language_preferences=())
        Render this Action's template with data from the given context.

    Parameters

```

- **context** (`shoop.notify.script.Context`) – Script Context
- **language_preferences** (`list[str]`) – Language preference list. The first language in the template to have values for all fields will be used. Has no effect for UNILINGUAL template_use.

Returns Dict of field name -> rendered template text.

Return type `dict[str, str]|None`

```
identifier = 'action'
name = 'Action'
provide_category = 'notify_action'
template_fields = {}
template_use = <TemplateUse.NONE: 0>
variables = {}
class shoop.notify.Binding(name, type=<class 'shoop.notify.typology.Type'>, required=False,
                           help_text='', constant_use=<ConstantUse.VARIABLE_ONLY: 1>,
                           default=None)
    Bases: shoop.notify.base.Variable
    accepts_any_type
    allow_constant
    allow_variable
    get_value(context, bind_data)
class shoop.notify.Context(variables=None)
    Bases: object
    add_log_entry_on_log_target(message, identifier, **kwargs)
        Add a log entry on the context's log target.

        The kwargs are passed to the target's add_log_entry method.

        If no log target exists or if it has no add_log_entry method, this method does nothing.

    Parameters
    • message (str) – The message text.
    • identifier (str) – The message identifier. Unlike with add_log_entry, this is
      required.
    • kwargs (dict) – Other kwargs to pass to add_log_entry
    classmethod from_event(event)
        Create Context from event.

    Return type shoop.notify.script.Context
    classmethod from_variables(**variables)
        Create Context from variables.

    Return type shoop.notify.script.Context
    get(name, default=None)
    get_variables()
```

```
log (level, msg, *args, **kwargs)
```

Log a message with the context's logger (not the log target). This may be a useful debugging tool.

The parameters are the same as for `logging.Logger.log()`.

```
log_entry_queryset
```

```
set (name, value)
```

```
class shoop.notify.Condition (data, validate=True)
```

Bases: `shoop.notify.base.ScriptItem`

```
bindings = {}
```

```
identifier = 'condition'
```

```
name = 'Condition'
```

```
provide_category = 'notify_condition'
```

```
test (context)
```

```
variables = {}
```

```
class shoop.notify.ConstantUse
```

Bases: `enumfields.enums.Enum`

An enumeration.

```
CONSTANT_ONLY = <ConstantUse.CONSTANT_ONLY: 2>
```

```
VARIABLE_ONLY = <ConstantUse.VARIABLE_ONLY: 1>
```

```
VARIABLE_OR_CONSTANT = <ConstantUse.VARIABLE_OR_CONSTANT: 3>
```

```
class shoop.notify.Event (**variable_values)
```

Bases: `shoop.notify.base.Base`

```
bindings = {}
```

```
identifier = None
```

```
load_variables (variable_values)
```

```
log_target
```

```
log_target_variable = None
```

```
provide_category = 'notify_event'
```

```
run ()
```

```
variables = {}
```

```
class shoop.notify.TemplateUse
```

Bases: `enumfields.enums.Enum`

An enumeration.

```
MULTILINGUAL = <TemplateUse.MULTILINGUAL: 2>
```

```
NONE = <TemplateUse.NONE: 0>
```

```
UNILINGUAL = <TemplateUse.UNILINGUAL: 1>
```

```
class shoop.notify.Variable (name, type=<class 'shoop.notify.typology.Type'>, required=True,  
                             help_text='')
```

Bases: `object`

```
get_matching_types (variable_dict)
```

shoop.order_printouts package

Subpackages

shoop.order_printouts.admin_module package

Submodules

shoop.order_printouts.admin_module.toolbar module

```
class shoop.order_printouts.admin_module.toolbar.SimplePrintoutsToolBarButton(order,  
                                                                              **kwargs)  
    Bases: shoop.admin.toolbar.DropDownActionButton  
    get_menu_items()
```

shoop.order_printouts.admin_module.views module

```
shoop.order_printouts.admin_module.views.get_footer_information(shop)  
shoop.order_printouts.admin_module.views.get_delivery_pdf(request, shipment_pk)  
shoop.order_printouts.admin_module.views.get_confirmation_pdf(request, order_pk)
```

Module contents

```
class shoop.order_printouts.admin_module.PrintoutsAdminModule  
    Bases: shoop.admin.base.AdminModule  
    get_urls()
```

Submodules

shoop.order_printouts.pdf_export module

```
shoop.order_printouts.pdf_export.render_html_to_pdf(html, stylesheet_paths=[])  
shoop.order_printouts.pdf_export.wrap_pdf_in_response(pdf_data)
```

Module contents

```
class shoop.order_printouts.AppConfig(*args, **kwargs)  
    Bases: shoop.apps.AppConfig  
    name = 'shoop.order_printouts'  
    verbose_name = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    label = 'shoop_order_printouts'  
    provides = {'admin_module': ['shoop.order_printouts.admin_module:PrintoutsAdminModule'], 'admin_order_toolba
```

shoop.simple cms package

Subpackages

shoop.simple_cms.admin_module package**Submodules****shoop.simple_cms.admin_module.views module**

class shoop.simple_cms.admin_module.views.**PageForm**(**kwargs)

Bases: *shoop.utils.multilanguage_model_form.MultiLanguageModelForm*

class **Meta**

Bases: *object*

model

alias of *Page*

fields = ['title', 'url', 'content', 'available_from', 'available_to', 'identifier', 'visible_in_menu', 'parent', 'list_child']

PageForm.**clean**()

If title or content has been given on any language we must enforce that the other fields are also required in that language.

This is done the way it is because url is not required by default in model level.

PageForm.**is_url_valid**(language_code, field_name, url)

Ensure URL given is unique.

Check through the pages translation model objects to make sure that the url given doesn't already exist.

Possible failure cases: for new page: * URL already exists

for existing page: * URL (other than owned by existing page) exists * URL exists in other languages of existing page

PageForm.**base_fields** = *OrderedDict*([('title', <django.forms.fields.CharField object at 0x7f71386ce4e0>), ('url', <d

PageForm.**declared_fields** = *OrderedDict*([('available_from', <django.forms.fields.DateTimeField object at 0x7f71

PageForm.**media**

class shoop.simple_cms.admin_module.views.**PageEditView**(**kwargs)

Bases: *shoop.admin.utils.views.CreateOrUpdateView*

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of *Page*

template_name = 'shoop/simple_cms/admin/edit.jinja'

form_class

alias of *PageForm*

context_object_name = 'page'

add_form_errors_as_messages = *True*

save_form(form)

class shoop.simple_cms.admin_module.views.**PageListView**(**kwargs)

Bases: *shoop.admin.utils.views.PicotableView*

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

model

alias of *Page*

```
columns = [<shoop.admin.utils.picotable.Column object at 0x7f71386ce1d0>, <shoop.admin.utils.picotable.Column object at 0x7f71386ce1d0>]
get_object_abstract (instance, item)
```

Module contents

```
class shoop.simple_cms.admin_module.SimpleCMSAdminModule
    Bases: shoop.admin.base.AdminModule

    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>
    get_urls ()
    get_menu_category_icons ()
    get_menu_entries (request)
    get_required_permissions ()
    get_model_url (object, kind)
```

Submodules

shoop.simple_cms.models module

```
class shoop.simple_cms.models.PageQuerySet (*args, **kwargs)
    Bases: parler.managers.TranslatableQuerySet

    visible (dt=None)
        Get pages that should be publicly visible.

        This does not do permission checking.

        Parameters dt (datetime.datetime) – Datetime for visibility check

        Returns QuerySet of pages.

        Return type QuerySet[Page]
class shoop.simple_cms.models.Page (id, available_from, available_to, created_by, modified_by,
                                     created_on, modified_on, identifier, visible_in_menu, parent,
                                     list_children_on_page)
    Bases: mptt.models.MPTTModel, parler.models.TranslatableModel

    created_by
    modified_by
    parent
    translations
    objects = <django.db.models.manager.TreeManagerFromPageQuerySet object>
    is_visible (dt=None)
    get_html ()
    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist
    exception Page.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned
    Page.children
```

`Page.content`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Page.get_next_by_created_on(*moreargs, **morekwargs)`

`Page.get_next_by_modified_on(*moreargs, **morekwargs)`

`Page.get_previous_by_created_on(*moreargs, **morekwargs)`

`Page.get_previous_by_modified_on(*moreargs, **morekwargs)`

`Page.title`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

`Page.url`

Descriptor for translated attributes.

This attribute proxies all get/set calls to the translated model.

class `shoop.simple_cms.models.PageTranslation(id, language_code, title, url, content, master)`

Bases: `parler.models.TranslatedFieldsModel`

exception `DoesNotExist`

Bases: `parler.models.TranslationDoesNotExist`, `shoop.simple_cms.models.DoesNotExist`, `shoop.simple_cms.models.DoesNotExist`

exception `PageTranslation.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`PageTranslation.get_language_code_display(*moreargs, **morekwargs)`

`PageTranslation.master`

`PageTranslation.objects = <django.db.models.manager.Manager object>`

shoop.simple_cms.plugins module

class `shoop.simple_cms.plugins.PageLinksConfigForm(**kwargs)`

Bases: `shoop.xtheme.plugins.forms.GenericPluginForm`

A configuration for the PageLinksPlugin

populate()

A custom populate method to display page choices

clean()

A custom clean method to save page configuration information in a serializable form

base_fields = OrderedDict()

declared_fields = OrderedDict()

media

class `shoop.simple_cms.plugins.PageLinksPlugin(config)`

Bases: `shoop.xtheme.TemplatedPlugin`

A plugin for displaying links to visible CMS pages in the shop front

Instantiate a Plugin with the given `config` dictionary.

Parameters `config(dict)` – Dictionary of freeform configuration data

identifier = 'simple_cms.page_links'

```
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
template_name = 'shoop/simple_cms/plugins/page_links.jinja'
editor_form_class
    alias of PageLinksConfigForm
fields = [('title', <shoop.xtheme.plugins.forms.TranslatableField object at 0x7f713a3f57f0>), ('show_all_pages', <django
get_context_data (context)
    A custom get_context_data method to return pages, possibly filtering expired pages based on the plugin's
    hide_expired setting
```

shoop.simple_cms.template_helpers module

```
class shoop.simple_cms.template_helpers.SimpleCMSTemplateHelpers
    Bases: object
    name = 'simple_cms'
    get_page_by_identifier (identifier)
    get_visible_pages ()
```

shoop.simple_cms.urls module

shoop.simple_cms.views module

```
class shoop.simple_cms.views.PageView (**kwargs)
    Bases: django.views.generic.detail.DetailView
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    model
        alias of Page
    slug_field = 'translations__url'
    slug_url_kwarg = 'url'
    template_name = 'shoop/simple_cms/page.jinja'
    context_object_name = 'page'
    get (request, *args, **kwargs)
        Override normal get method to return correct page based on the active language and slug
    Cases:
        1. Page is not found: raise Http404() like django would
        2. No translation in active language for the page: raise Http404()
        3. Translation was found for active language, but the url doesn't match given url: return
           HttpResponseRedirect to the active languages url
        4. If none of the upper matches: render page normally
    get_queryset ()
```

Module contents

```
class shoop.simple_cms.AppConfig(*args, **kwargs)
    Bases: shoop.apps.AppConfig

    name = 'shoop.simple_cms'

    verbose_name = <django.utils.functional.lazy.<locals>.__proxy__ object>

    label = 'shoop_simple_cms'

    provides = {'admin_module': ['shoop.simple_cms.admin_module:SimpleCMSAdminModule'], 'front_urls_post': ['shoop.simple_cms.front_urls_post:SimpleCMSFrontUrlsPost']}
```

shoop.simple_supplier package

Subpackages

shoop.simple_supplier.admin_module package

Submodules

shoop.simple_supplier.admin_module.forms module

```
class shoop.simple_supplier.admin_module.forms.SimpleSupplierForm(**kwargs)
    Bases: django.forms.forms.Form

    save()

    get_suppliers(product)

    can_manage_stock()

    get_stock_information(supplier, product)

    get_stock_adjustment_form(supplier, product)

    base_fields = OrderedDict()

    declared_fields = OrderedDict()

    media

class shoop.simple_supplier.admin_module.forms.SimpleSupplierFormPart(request, ob-
                                                                    ject=None)
    Bases: shoop.admin.form_part.FormPart

    priority = 15

    name = 'simple_supplier'

    form
        alias of SimpleSupplierForm

    get_form_defs()

    form_valid(form)
```

shoop.simple_supplier.admin_module.views module**class** shoop.simple_supplier.admin_module.views.**StocksListView** (**kwargs)Bases: *shoop.admin.utils.views.PicortableListView*

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

template_name = 'shoop/simple_supplier/admin/base_picortable.jinja'**model**

alias of Product

columns = [<shoop.admin.utils.picortable.Column object at 0x7f713b0dc908>, <shoop.admin.utils.picortable.Column object at 0x7f713b0dc908>]**get_object_abstract** (instance, item)**get_queryset** ()**get_context_data** (**kwargs)**get_stock_information** (instance)**get_stock_adjustment_form** (instance)shoop.simple_supplier.admin_module.views.**get_adjustment_success_message** (stock_adjustment)shoop.simple_supplier.admin_module.views.**process_stock_adjustment** (request,
supplier_id,
product_id)**Module contents****class** shoop.simple_supplier.admin_module.**StocksAdminModule**Bases: *shoop.admin.base.AdminModule***name** = <django.utils.functional.lazy.<locals>.__proxy__ object>**category** = <django.utils.functional.lazy.<locals>.__proxy__ object>**get_urls** ()**get_menu_category_icons** ()**get_required_permissions** ()**get_menu_entries** (request)**Submodules****shoop.simple_supplier.forms module****class** shoop.simple_supplier.forms.**StockAdjustmentForm** (data=None, files=None,
auto_id='id_%s', pre-
fix=None, initial=None,
error_class=<class
'django.forms.utils.ErrorList'>,
label_suffix=None,
empty_permitted=False)Bases: *django.forms.forms.Form***clean_delta** ()**base_fields** = OrderedDict([('purchase_price', <django.forms.fields.DecimalField object at 0x7f713b023f60>), ('delta', <django.forms.fields.DecimalField object at 0x7f713b023f60>)])**declared_fields** = OrderedDict([('purchase_price', <django.forms.fields.DecimalField object at 0x7f713b023f60>), ('delta', <django.forms.fields.DecimalField object at 0x7f713b023f60>)])

media

shoop.simple_supplier.models module

class shoop.simple_supplier.models.**StockAdjustment** (*id, product, supplier, created_on, created_by, delta, purchase_price_value*)

Bases: django.db.models.base.Model

product

supplier

created_by

purchase_price

Property for Price object.

Similar to MoneyProperty but also has includes_tax field.

Operators with TaxfulPrice and TaxlessPrice objects.

currency

includes_tax

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception StockAdjustment.**MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

StockAdjustment.**get_next_by_created_on** (*moreargs, **morekwargs)

StockAdjustment.**get_previous_by_created_on** (*moreargs, **morekwargs)

StockAdjustment.**objects** = <django.db.models.manager.Manager object>

class shoop.simple_supplier.models.**StockCount** (*id, product, supplier, logical_count, physical_count, stock_value_value*)

Bases: django.db.models.base.Model

product

supplier

stock_value

Property for Price object.

Similar to MoneyProperty but also has includes_tax field.

Operators with TaxfulPrice and TaxlessPrice objects.

stock_unit_price

Property for Price object.

Similar to MoneyProperty but also has includes_tax field.

Operators with TaxfulPrice and TaxlessPrice objects.

currency

includes_tax

stock_unit_price_value

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

```
exception StockCount.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

StockCount.objects = <django.db.models.manager.Manager object>
```

shoop.simple_supplier.module module

```
class shoop.simple_supplier.module.SimpleSupplierModule (supplier, options)
    Bases: shoop.core.suppliers.base.BaseSupplierModule
```

```
    identifier = 'simple_supplier'
    name = 'Simple Supplier'
    get_stock_statuses (product_ids)
    adjust_stock (product_id, delta, purchase_price=0, created_by=None)
    update_stock (product_id)
```

shoop.simple_supplier.utils module

```
shoop.simple_supplier.utils.get_current_stock_value (supplier_id, product_id)
    Count stock values for supplier and product combination
```

Logical count is events minus orders bought (not cancelled) describing how many products is currently orderable
Physical count is events minus orders actually sent describing how many products is currently in stock

Parameters

- **supplier_id** – supplier_id to count stock values for
- **product_id** – product_id to count stock values for

Returns logical and physical count for product

Return type dict

```
shoop.simple_supplier.utils.get_stock_information_div_id (supplier, product)
```

```
shoop.simple_supplier.utils.get_stock_information_html (supplier, product)
    Get html string to show current stock information for product
```

Parameters

- **supplier** (shoop.core.models.Supplier) – shoop Supplier
- **product** (shoop.core.models.Product) – shoop Product

Returns html div as a string

Return type str

```
shoop.simple_supplier.utils.get_stock_adjustment_div (request, supplier, product)
    Get html string to adjust stock values
```

Contains inputs for purchase_price_value and delta

Parameters

- **request** (django.http.HttpRequest) – HTTP request
- **supplier** (shoop.core.models.Supplier) – shoop Supplier
- **product** (shoop.core.models.Product) – shoop Product

Returns html div as a string

Return type `str`

Module contents

class `shoop.simple_supplier.ShoopSimpleSupplierAppConfig(*args, **kwargs)`

Bases: `shoop.apps.AppConfig`

name = `'shoop.simple_supplier'`

verbose_name = `'Shoop Simple Supplier'`

label = `'simple_supplier'`

provides = `{'supplier_module': ['shoop.simple_supplier.module:SimpleSupplierModule'], 'admin_product_form_part'}`

`shoop.testing` package

Subpackages

`shoop.testing.admin_module` package

Submodules

shoop.testing.admin_module.mockerview module

class `shoop.testing.admin_module.mockerview.Mockers`

Bases: `object`

Namespace object for mocker methods.

The docstrings for the callables are user-visible.

mock_order ()

Create a random order

mock_person ()

Create a random person

mock_company ()

Create a random company

class `shoop.testing.admin_module.mockerview.MockerForm(data=None, files=None, auto_id='id_%s', pre-fix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=None, empty_permitted=False)`

Bases: `django.forms.forms.Form`

base_fields = `OrderedDict([('type', <django.forms.fields.ChoiceField object at 0x7f7137c72320>), ('count', <django.f`

declared_fields = `OrderedDict([('type', <django.forms.fields.ChoiceField object at 0x7f7137c72320>), ('count', <dja`

media

class `shoop.testing.admin_module.mockerview.MockerView(**kwargs)`

Bases: `django.views.generic.edit.FormView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
form_class
    alias of MockerForm

template_name = 'shoop_testing/mocker.jinja'

mockers = <shoop.testing.admin_module.mockers_view.Mockers object>

get_mockers ()

get_form (form_class=None)

form_valid (form)
```

Module contents

```
class shoop.testing.admin_module.TestingAdminModule
    Bases: shoop.admin.base.AdminModule

    get_urls ()

    get_menu_entries (request)

    get_required_permissions ()
```

shoop.testing.management package

Subpackages

shoop.testing.management.commands package

Submodules

shoop.testing.management.commands.shoop_populate_mock module

```
class shoop.testing.management.commands.shoop_populate_mock.Command (stdout=None,
                                                                    stderr=None,
                                                                    no_color=False)

    Bases: django.core.management.base.BaseCommand

    option_list = (<Option at 0x7f7137c5d668: --with-superuser>,)

    handle (*args, **options)
```

Module contents

Module contents

shoop.testing.models package

Module contents

class `shoop.testing.models.CarrierWithCheckoutPhase` (*id, polymorphic_ctype, identifier, enabled, logo, serviceprovider_ptr, carrier_ptr, customcarrier_ptr*)

Bases: `shoop.core.models.CustomCarrier`

Replace Django's inheritance accessor member functions for our model (`self.__class__`) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate `base_objects` manager.

exception DoesNotExist

Bases: `shoop.core.models._service_shipping.DoesNotExist`

exception CarrierWithCheckoutPhase.MultipleObjectsReturned

Bases: `shoop.core.models._service_shipping.MultipleObjectsReturned`

`CarrierWithCheckoutPhase.base_objects = <parler.managers.TranslatableManager object>`

`CarrierWithCheckoutPhase.customcarrier_ptr`

`CarrierWithCheckoutPhase.objects = <shoop.core.models._base._PolyTransManager object>`

`CarrierWithCheckoutPhase.polymorphic_primary_key_name = 'id'`

`CarrierWithCheckoutPhase.polymorphic_super_sub_accessors_replaced = False`

class `shoop.testing.models.ExpensiveSwedenBehaviorComponent` (*id, polymorphic_ctype, servicebehaviorcomponent_ptr*)

Bases: `shoop.core.models.ServiceBehaviorComponent`

exception DoesNotExist

Bases: `shoop.core.models._service_base.DoesNotExist`

exception ExpensiveSwedenBehaviorComponent.MultipleObjectsReturned

Bases: `shoop.core.models._service_base.MultipleObjectsReturned`

`ExpensiveSwedenBehaviorComponent.base_objects = <django.db.models.manager.Manager object>`

`ExpensiveSwedenBehaviorComponent.get_costs (service, source)`

`ExpensiveSwedenBehaviorComponent.get_unavailability_reasons (service, source)`

`ExpensiveSwedenBehaviorComponent.name = 'Expenseefe-a Svedee Sheepping'`

`ExpensiveSwedenBehaviorComponent.objects = <polymorphic.managers.PolymorphicManager object>`

`ExpensiveSwedenBehaviorComponent.polymorphic_primary_key_name = 'id'`

`ExpensiveSwedenBehaviorComponent.polymorphic_super_sub_accessors_replaced = False`

`ExpensiveSwedenBehaviorComponent.servicebehaviorcomponent_ptr`

```
class shoop.testing.models.PaymentWithCheckoutPhase(id, polymorphic_ctype, identifier,
                                                    enabled, logo, serviceprovider_ptr,
                                                    paymentprocessor_ptr, custompaymentprocessor_ptr)
```

Bases: *shoop.core.models.CustomPaymentProcessor*

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

exception DoesNotExist

Bases: *shoop.core.models._service_payment.DoesNotExist*

exception PaymentWithCheckoutPhase.MultipleObjectsReturned

Bases: *shoop.core.models._service_payment.MultipleObjectsReturned*

PaymentWithCheckoutPhase.base_objects = <parler.managers.TranslatableManager object>

PaymentWithCheckoutPhase.custompaymentprocessor_ptr

PaymentWithCheckoutPhase.objects = <shoop.core.models._base._PolyTransManager object>

PaymentWithCheckoutPhase.polymorphic_primary_key_name = 'id'

PaymentWithCheckoutPhase.polymorphic_super_sub_accessors_replaced = False

PaymentWithCheckoutPhase.process_payment_return_request (service, order, request)

```
class shoop.testing.models.PseudoPaymentProcessor(id, polymorphic_ctype, identifier,
                                                  enabled, logo, serviceprovider_ptr,
                                                  paymentprocessor_ptr, bg_color,
                                                  fg_color)
```

Bases: *shoop.core.models.PaymentProcessor*

Replace Django's inheritance accessor member functions for our model (self.__class__) with our own versions. We monkey patch them until a patch can be added to Django (which would probably be very small and make all of this obsolete).

If we have inheritance of the form ModelA -> ModelB -> ModelC then Django creates accessors like this: - ModelA: modelb - ModelB: modela_ptr, modelb, modelc - ModelC: modela_ptr, modelb, modelb_ptr, modelc

These accessors allow Django (and everyone else) to travel up and down the inheritance tree for the db object at hand.

The original Django accessors use our polymorphic manager. But they should not. So we replace them with our own accessors that use our appropriate base_objects manager.

exception DoesNotExist

Bases: *shoop.core.models._service_payment.DoesNotExist*

exception PseudoPaymentProcessor.MultipleObjectsReturned

Bases: *shoop.core.models._service_payment.MultipleObjectsReturned*

PseudoPaymentProcessor.base_objects = <parler.managers.TranslatableManager object>

```

PseudoPaymentProcessor.compute_pseudo_mac(order)
PseudoPaymentProcessor.get_payment_process_response(service, order, urls)
PseudoPaymentProcessor.get_service_choices()
PseudoPaymentProcessor.objects = <shoop.core.models._base._PolyTransManager object>
PseudoPaymentProcessor.paymentprocessor_ptr
PseudoPaymentProcessor.polymorphic_primary_key_name = 'id'
PseudoPaymentProcessor.polymorphic_super_sub_accessors_replaced = False
PseudoPaymentProcessor.process_payment_return_request(service, order, request)

```

Submodules

shoop.testing.factories module

```

class shoop.testing.factories.FuzzyBoolean(probability, **kwargs)
    Bases: factory.fuzzy.BaseFuzzyAttribute

```

```

    fuzz()

```

```

class shoop.testing.factories.UserFactory
    Bases: factory.django.DjangoModelFactory

```

Would be called if trying to instantiate the class.

```

    username = <factory.declarations.Sequence object>
    email = <factory.declarations.Sequence object>
    password = <factory.declarations.PostGenerationMethodCall object>
    first_name = <factory.fuzzy.FuzzyText object>
    last_name = <factory.fuzzy.FuzzyText object>

```

```

class shoop.testing.factories.CompanyFactory
    Bases: factory.django.DjangoModelFactory

```

Would be called if trying to instantiate the class.

```

    name = <factory.fuzzy.FuzzyText object>
    vat_id = <factory.fuzzy.FuzzyText object>
    email = <factory.declarations.Sequence object>

```

```

class shoop.testing.factories.ShopFactory
    Bases: factory.django.DjangoModelFactory

```

Would be called if trying to instantiate the class.

```

    slug = <factory.fuzzy.FuzzyText object>
    name = <factory.fuzzy.FuzzyText object>
    owner = <factory.declarations.SubFactory object>

```

```

class shoop.testing.factories.ProductTypeFactory
    Bases: factory.django.DjangoModelFactory

```

Would be called if trying to instantiate the class.

```

    identifier = <factory.declarations.Sequence object>

```

```
    name = <factory.fuzzy.FuzzyText object>
class shoop.testing.factories.SalesUnitFactory
    Bases: factory.django.DjangoModelFactory
    Would be called if trying to instantiate the class.
    name = <factory.fuzzy.FuzzyText object>
    short_name = <factory.fuzzy.FuzzyText object>
class shoop.testing.factories.CategoryFactory
    Bases: factory.django.DjangoModelFactory
    Would be called if trying to instantiate the class.
    identifier = <factory.declarations.Sequence object>
    name = <factory.fuzzy.FuzzyText object>
    status = <factory.fuzzy.FuzzyChoice object>
    post = <factory.declarations.PostGeneration object>
class shoop.testing.factories.ShopProductFactory
    Bases: factory.django.DjangoModelFactory
    Would be called if trying to instantiate the class.
    visible = <shoop.testing.factories.FuzzyBoolean object>
    listed = <shoop.testing.factories.FuzzyBoolean object>
    purchasable = <shoop.testing.factories.FuzzyBoolean object>
    searchable = <shoop.testing.factories.FuzzyBoolean object>
    default_price_value = <factory.fuzzy.FuzzyDecimal object>
class shoop.testing.factories.FuzzyName (prefix='', length=12, suffix='',
                                         chars='abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ'
                                         **kwargs)
    Bases: factory.fuzzy.FuzzyText
    fuzz ()
class shoop.testing.factories.ProductFactory
    Bases: factory.django.DjangoModelFactory
    Would be called if trying to instantiate the class.
    type = <factory.declarations.LazyAttribute object>
    sku = <factory.fuzzy.FuzzyText object>
    sales_unit = <factory.declarations.LazyAttribute object>
    tax_class = <factory.declarations.LazyAttribute object>
    profit_center = <factory.fuzzy.FuzzyInteger object>
    cost_center = <factory.fuzzy.FuzzyInteger object>
    name = <shoop.testing.factories.FuzzyName object>
    post = <factory.declarations.PostGeneration object>
shoop.testing.factories.get_address (**overrides)
shoop.testing.factories.default_by_identifier (model)
```

```

shoop.testing.factories.get_default_attribute_set()
shoop.testing.factories.get_default_product_type()
shoop.testing.factories.get_tax(code, name, rate=None, amount=None)
shoop.testing.factories.create_default_tax_rule(tax)
shoop.testing.factories.get_default_tax()
shoop.testing.factories.get_test_tax(rate)
shoop.testing.factories.get_default_tax_class()
shoop.testing.factories.get_custom_payment_processor()
shoop.testing.factories.get_custom_carrier()
shoop.testing.factories.get_default_payment_method()
shoop.testing.factories.get_payment_method(shop=None, price=None, waive_at=None,
                                         name=None)
shoop.testing.factories.get_default_shipping_method()
shoop.testing.factories.get_shipping_method(shop=None, price=None, waive_at=None,
                                         name=None)
shoop.testing.factories.get_default_customer_group()
shoop.testing.factories.get_default_supplier()
shoop.testing.factories.get_default_shop()
shoop.testing.factories.get_shop(prices_include_tax, currency='EUR')
shoop.testing.factories.get_default_product()
shoop.testing.factories.get_default_shop_product()
shoop.testing.factories.get_default_sales_unit()
shoop.testing.factories.get_default_category()
shoop.testing.factories.get_initial_order_status()
shoop.testing.factories.get_completed_order_status()
shoop.testing.factories.create_product(sku, shop=None, supplier=None, de-
                                     fault_price=None, **attrs)
shoop.testing.factories.create_empty_order(prices_include_tax=False, shop=None)
shoop.testing.factories.add_product_to_order(order, supplier, product, quantity, tax-
                                     less_base_unit_price, tax_rate=0, pric-
                                     ing_context=None)
shoop.testing.factories.create_order_with_product(product, supplier, quantity, tax-
                                     less_base_unit_price, tax_rate=0,
                                     n_lines=1, shop=None)
shoop.testing.factories.get_random_filer_image()
shoop.testing.factories.get_faker(providers, locale=None)
shoop.testing.factories.create_random_address(fake=None, **values)

```

`shoop.testing.factories.create_random_person(locale=None, mini-
mum_name_comp_len=0)`

Create a random `PersonContact` from the given locale (or a random one).

The minimum length for name components can be given, to work around possible issues with components expecting a long-enough string.

Parameters

- **locale** (*str/None*) – Locale name
- **minimum_name_comp_len** (*int*) – Minimum name component length

Returns Person contact

Return type *PersonContact*

`shoop.testing.factories.create_random_company()`

`shoop.testing.factories.create_random_order(customer=None, products=(), comple-
tion_probability=0, shop=None)`

shoop.testing.image_generator module

class `shoop.testing.image_generator.BaseImageGenerator(image, palette, seed)`
Bases: *object*

Parameters

- **image** (*PIL.Image.Image*) – The image to draw on
- **palette** – A list of RGB tuples
- **seed** (*int*) – Random generator seed

generate ()

draw_circle (*x, y, w, h, color*)

draw_rectangle (*x, y, w, h, color*)

class `shoop.testing.image_generator.RandomImageGenerator(image, palette, seed)`
Bases: *shoop.testing.image_generator.BaseImageGenerator*

Parameters

- **image** (*PIL.Image.Image*) – The image to draw on
- **palette** – A list of RGB tuples
- **seed** (*int*) – Random generator seed

generate ()

step ()

class `shoop.testing.image_generator.ModernArtImageGenerator(image, palette, seed)`
Bases: *shoop.testing.image_generator.BaseImageGenerator*

Parameters

- **image** (*PIL.Image.Image*) – The image to draw on
- **palette** – A list of RGB tuples
- **seed** (*int*) – Random generator seed

generate ()


```
class shoop.testing.image_generator.RingImageGenerator (image, palette, seed)
    Bases: shoop.testing.image_generator.BaseImageGenerator
```

Parameters

- **image** (*PIL.Image.Image*) – The image to draw on
- **palette** – A list of RGB tuples
- **seed** (*int*) – Random generator seed

```
generate ()
```

```
shoop.testing.image_generator.generate_image (width, height, palette=None, seed=None,
                                                supersample=2)
```

shoop.testing.mock_population module

```
class shoop.testing.mock_population.Populator
```

```
    Bases: object
```

```
    populate ()
```

```
    generate_pricing (product)
```

```
    populate_if_required ()
```

```
shoop.testing.mock_population.populate_if_required ()
```

shoop.testing.service_forms module

```
class shoop.testing.service_forms.PseudoPaymentProcessorForm (**kwargs)
```

```
    Bases: shoop.admin.forms._base.ShoopAdminForm
```

```
class Meta
```

```
    Bases: object
```

```
    model
```

```
        alias of PseudoPaymentProcessor
```

```
    exclude = ['identifier']
```

```
PseudoPaymentProcessorForm.base_fields = OrderedDict([(‘enabled’, <django.forms.fields.BooleanField object>)])
```

```
PseudoPaymentProcessorForm.declared_fields = OrderedDict([(‘name’, <django.forms.fields.CharField object>)])
```

```
PseudoPaymentProcessorForm.media
```

```
class shoop.testing.service_forms.PaymentWithCheckoutPhaseForm (**kwargs)
```

```
    Bases: shoop.admin.forms._base.ShoopAdminForm
```

```
class Meta
```

```
    Bases: object
```

```
    model
```

```
        alias of PaymentWithCheckoutPhase
```

```
    exclude = ['identifier']
```

```
PaymentWithCheckoutPhaseForm.base_fields = OrderedDict([(‘enabled’, <django.forms.fields.BooleanField object>)])
```

```
PaymentWithCheckoutPhaseForm.declared_fields = OrderedDict([(‘name’, <django.forms.fields.CharField object>)])
```

```
PaymentWithCheckoutPhaseForm.media
```

```
class shoop.testing.service_forms.CarrierWithCheckoutPhaseForm (**kwargs)
```

```
    Bases: shoop.admin.forms._base.ShoopAdminForm
```

```
class Meta
    Bases: object

    model
        alias of CarrierWithCheckoutPhase

    exclude = ['identifier']

CarrierWithCheckoutPhaseForm.base_fields = OrderedDict([('enabled', <django.forms.fields.BooleanField object at 0x7f7138ff6160>)])
CarrierWithCheckoutPhaseForm.declared_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f7138ff6160>)])
CarrierWithCheckoutPhaseForm.media
```

shoop.testing.simple_checkout_phase module

```
class shoop.testing.simple_checkout_phase.TestPaymentCheckoutPhaseForm(data=None,
                                                                       files=None,
                                                                       auto_id='id_%s',
                                                                       pre-
                                                                       fix=None,
                                                                       ini-
                                                                       tial=None,
                                                                       er-
                                                                       ror_class=<class
                                                                       'django.forms.utils.ErrorList'>,
                                                                       la-
                                                                       bel_suffix=None,
                                                                       empty_permitted=False)

    Bases: django.forms.forms.Form

    base_fields = OrderedDict([('input_field', <django.forms.fields.BooleanField object at 0x7f7138ff6160>)])
    declared_fields = OrderedDict([('input_field', <django.forms.fields.BooleanField object at 0x7f7138ff6160>)])
    media

class shoop.testing.simple_checkout_phase.TestShipmentCheckoutPhaseForm(data=None,
                                                                           files=None,
                                                                           auto_id='id_%s',
                                                                           pre-
                                                                           fix=None,
                                                                           ini-
                                                                           tial=None,
                                                                           er-
                                                                           ror_class=<class
                                                                           'django.forms.utils.ErrorList'>,
                                                                           la-
                                                                           bel_suffix=None,
                                                                           empty_permitted=False)

    Bases: django.forms.forms.Form

    base_fields = OrderedDict([('input_field', <django.forms.fields.CharField object at 0x7f7138fe1e10>)])
    declared_fields = OrderedDict([('input_field', <django.forms.fields.CharField object at 0x7f7138fe1e10>)])
    media

class shoop.testing.simple_checkout_phase.TestMethodCheckoutPhase(**kwargs)
    Bases: shoop.front.checkout.CheckoutPhaseViewMixin, django.views.generic.edit.FormView

    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
```

```

    template_name = 'shoop_testing/simple_checkout_phase.jinja'
    data_attribute = None
    storage_identifier = None
    get_initial()
    form_valid(form)
    is_valid()
    process()
class shoop.testing.simple_checkout_phase.TestPaymentCheckoutPhase(**kwargs)
    Bases: shoop.testing.simple_checkout_phase.TestMethodCheckoutPhase
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    identifier = 'test_payment_phase'
    title = 'Test Payment Phase'
    form_class
        alias of TestPaymentCheckoutPhaseForm
    storage_identifier = 'payment_with_checkout_phase'
    data_attribute = 'payment_data'
class shoop.testing.simple_checkout_phase.TestShipmentCheckoutPhase(**kwargs)
    Bases: shoop.testing.simple_checkout_phase.TestMethodCheckoutPhase
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    identifier = 'test_shipment_phase'
    title = 'Test Shipment Phase'
    form_class
        alias of TestShipmentCheckoutPhaseForm
    storage_identifier = 'shipment_with_checkout_phase'
    data_attribute = 'shipping_data'
class shoop.testing.simple_checkout_phase.PaymentPhaseProvider
    Bases: shoop.front.checkout.BasicServiceCheckoutPhaseProvider
    phase_class
        alias of TestPaymentCheckoutPhase
    service_provider_class
        alias of PaymentWithCheckoutPhase
class shoop.testing.simple_checkout_phase.ShipmentPhaseProvider
    Bases: shoop.front.checkout.BasicServiceCheckoutPhaseProvider
    phase_class
        alias of TestShipmentCheckoutPhase
    service_provider_class
        alias of CarrierWithCheckoutPhase

```

shoop.testing.soup_utils module

`shoop.testing.soup_utils.extract_form_fields(soup)`
Turn a BeautifulSoup form in to a dict of fields and default values

shoop.testing.text_data module

`shoop.testing.text_data.random_title(second_adj_chance=0.3, prefix='', suffix='')`

shoop.testing.utils module

`shoop.testing.utils.apply_request_middleware(request, **attrs)`
Apply all the `process_request` capable middleware configured into the given request.

Parameters

- **request** (`django.http.HttpRequest`) – The request to massage.
- **attrs** (`dict`) – Additional attributes to set after massage.

Returns The same request, massaged in-place.

Return type `django.http.HttpRequest`

Module contents

class `shoop.testing.ShoopTestingAppConfig(*args, **kwargs)`

Bases: `shoop.apps.AppConfig`

name = 'shoop.testing'

verbose_name = 'Shoop Testing & Demo Utilities'

label = 'shoop_testing'

provides = {'admin_module': ['shoop.testing.admin_module:TestingAdminModule'], 'service_provider_admin_form':

shoop.themes package

Subpackages

shoop.themes.classic_gray package

Subpackages

shoop.themes.classic_gray.views package

Module contents

`shoop.themes.classic_gray.views.basket_partial(request)`

`shoop.themes.classic_gray.views.product_preview(request)`

`shoop.themes.classic_gray.views.products(request)`

`shoop.themes.classic_gray.views.product_price(request)`

Submodules

shoop.themes.classic_gray.plugins module

class shoop.themes.classic_gray.plugins.**ProductHighlightPlugin** (*config*)

Bases: *shoop.xtheme.TemplatedPlugin*

Instantiate a Plugin with the given config dictionary.

Parameters **config** (*dict*) – Dictionary of freeform configuration data

identifier = 'classic_gray.product_highlight'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

template_name = 'classic_gray/highlight_plugin.jinja'

fields = [(('title', <shoop.xtheme.plugins.forms.TranslatableField object at 0x7f7137d0a400>), ('type', <django.forms.fie

get_context_data (*context*)

class shoop.themes.classic_gray.plugins.**ProductCrossSellsPlugin** (*config*)

Bases: *shoop.xtheme.TemplatedPlugin*

identifier = 'classic_gray.product_cross_sells'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

template_name = 'classic_gray/cross_sells_plugin.jinja'

required_context_variables = ['product']

fields = [(('title', <shoop.xtheme.plugins.forms.TranslatableField object at 0x7f7137b9c128>), ('type', <enumfields.form

get_context_data (*context*)

Module contents

class shoop.themes.classic_gray.**ClassicGrayTheme** (*settings_obj=None*)

Bases: *shoop.xtheme.Theme*

Initialize this theme, with an optional ThemeSettings object

Parameters **settings_obj** (*ThemeSettings/None*) – A theme settings object for this theme, if one exists. Passing this in will avoid extraneous database queries.

identifier = 'shoop.themes.classic_gray'

name = 'Shoop Classic Gray Theme'

author = 'Juha Kujala'

template_dir = 'classic_gray/'

fields = [(('show_welcome_text', <django.forms.fields.BooleanField object at 0x7f713f7bbb70>)]

get_configuration_form (*form_kwargs*)

get_view (*view_name*)

get_cms_navigation_links ()

class shoop.themes.classic_gray.**ClassicGrayThemeAppConfig** (**args, **kwargs*)

Bases: *shoop.apps.AppConfig*

name = 'shoop.themes.classic_gray'

verbose_name = 'Shoop Classic Gray Theme'

label = 'shoop.themes.classic_gray'

provides = {'xtheme_plugin': ['shoop.themes.classic_gray.plugins:ProductHighlightPlugin', 'shoop.themes.classic_gray

Module contents

shoop.utils package

Submodules

shoop.utils.analog module

class `shoop.utils.analog.LogEntryKind`

Bases: `enumfields.enums.Enum`

An enumeration.

OTHER = `<LogEntryKind.OTHER: 0>`

AUDIT = `<LogEntryKind.AUDIT: 1>`

EDIT = `<LogEntryKind.EDIT: 2>`

DELETION = `<LogEntryKind.DELETION: 3>`

NOTE = `<LogEntryKind.NOTE: 4>`

EMAIL = `<LogEntryKind.EMAIL: 5>`

WARNING = `<LogEntryKind.WARNING: 6>`

ERROR = `<LogEntryKind.ERROR: 7>`

class `shoop.utils.analog.BaseLogEntry(*args, **kwargs)`

Bases: `django.db.models.base.Model`

target = `None`

user

kind

A placeholder class that provides a way to set the attribute on the model.

extra

A placeholder class that provides a way to set the attribute on the model.

class `Meta`

Bases: `object`

abstract = `False`

`BaseLogEntry.get_kind_display(*moreargs, **morekwargs)`

`BaseLogEntry.get_next_by_created_on(*moreargs, **morekwargs)`

`BaseLogEntry.get_previous_by_created_on(*moreargs, **morekwargs)`

`shoop.utils.analog.define_log_model(model_class)`

shoop.utils.dates module

`shoop.utils.dates.parse_date(value)`

Tries to make a date out of the value. If impossible, it raises an exception.

Parameters `value` – A value of some ilk.

Returns `Date`

Return type `datetime.date`

Raises `ValueError` –

`shoop.utils.dates.parse_time(value)`

Tries to make a time out of the value. If impossible, it raises an exception.

Parameters `value` – A value of some ilk.

Returns `Time`

Return type `datetime.time`

Raises `ValueError` –

`shoop.utils.dates.try_parse_date(value)`

Tries to make a time out of the value. If impossible, returns None.

Parameters `value` – A value of some ilk.

Returns `Date`

Return type `datetime.date`

`shoop.utils.dates.try_parse_time(value)`

Tries to make a time out of the value. If impossible, returns None.

Parameters `value` – A value of some ilk.

Returns `Time`

Return type `datetime.time`

`shoop.utils.decorators` module

`shoop.utils.decorators.non_reentrant(func)`

`shoop.utils.excs` module

exception `shoop.utils.excs.Problem(message, title=None)`

Bases: `Exception`

User-visible exception

message

with_link(url, title)

Append a link to this Problem and return itself.

This API is designed after `Exception.with_traceback()`, so you can fluently chain this in a `raise` statement:

```
raise Problem("Oops").with_link("...", "...")
```

Parameters

- **url** (`str`) – URL string
- **title** (`str`) – Title text

Returns This same Problem

Return type `shoop.utils.excs.Problem`

exception `shoop.utils.excs.ExceptionalResponse(response)`

Bases: `Exception`

`shoop.utils.excs.extract_messages(obj_list)`

Extract “messages” from a list of exceptions or other objects.

For `ValidationErrors`, messages are flattened into the output. For `Exceptions`, `args[0]` is added into the output. For other objects, `force_text` is called.

Parameters `obj_list` (*Iterable[object]*) – List of exceptions etc.

Return type `Iterable[str]`

shoop.utils.fields module

```
class shoop.utils.fields.RelaxedModelChoiceField(queryset, empty_label='—
—', cache_choices=None, re-
quired=True, widget=None,
label=None, initial=None,
help_text='', to_field_name=None,
limit_choices_to=None, *args,
**kwargs)
```

Bases: `django.forms.models.ModelChoiceField`

to_python (*value*)

shoop.utils.filer module

`shoop.utils.filer.filer_folder_from_path(path)`

Split `path` by slashes and create a hierarchy of `Filer Folder` objects accordingly. Blank path components are ignored, so `“/////foo/////bar//”` is the same as `“foo/bar”`.

The empty string (and `None`) are handled as “no folder”, i.e. root folder.

Parameters `path` (*str|None*) – Pathname or None

Returns `Folder`

Return type `filer.models.Folder`

`shoop.utils.filer.filer_file_from_upload(request, path, upload_data, sha1=None)`

Create a `filer.models.filemodels.File` from an upload (`UploadedFile` or such). If the `sha1` parameter is passed and a file with said SHA1 is found, it will be returned instead.

Parameters

- **request** (*django.http.request.HttpRequest|None*) – Request, to figure out the owner for this file
- **path** (*basestring|filer.models.Folder*) – Pathname string (see [filer_folder_from_path](#)) or a `Filer Folder`.
- **upload_data** (*django.core.files.base.File*) – Upload data
- **sha1** (*basestring*) – SHA1 checksum. If given and a matching model with the SHA1 is found, it is returned instead.

Return type `filer.models.filemodels.File`

`shoop.utils.filer.filer_image_from_upload(request, path, upload_data, sha1=None)`

Create a `Filer Image` from an upload (`UploadedFile` or such). If the `sha1` parameter is passed and an `Image` with said SHA1 is found, it will be returned instead.

Parameters

- **request** (*django.http.request.HttpRequest|None*) – Request, to figure out the owner for this file

- **path** (*basestring/filer.models.Folder*) – Pathname string (see *filer_folder_from_path*) or a Filer Folder.
- **upload_data** (*django.core.files.base.File*) – Upload data
- **sha1** (*basestring*) – SHA-1 checksum of the data, if available, to do deduplication

Return type *filer.models.imagemodels.Image*

`shoop.utils.filer.filer_image_from_data(request, path, file_name, file_data, sha1=None)`

Create a Filer Image from the given data string. If the `sha1` parameter is passed and `True` (the value `True`, not a `Truey` value), the SHA-1 of the data string is calculated and passed to the underlying creation function. If the `sha1` parameter is `truthy` (generally the SHA-1 hex string), it's passed directly to the creation function.

Parameters

- **request** (*django.http.request.HttpRequest/None*) – Request, to figure out the owner for this file
- **path** (*basestring/filer.models.Folder*) – Pathname string (see *filer_folder_from_path*) or a Filer Folder.
- **file_name** – File name
- **file_data** (*bytes*) – Upload data
- **sha1** (*basestring/bool*) – SHA-1 checksum of the data, if available, to do deduplication. May also be `True` to calculate the SHA-1 first.

Return type *filer.models.imagemodels.Image*

shoop.utils.form_group module

exception `shoop.utils.form_group.FormInstantiationAttributeError`

Bases: `Exception`

AttributeErrors occurring within the `forms` property are transmogrified into these.

class `shoop.utils.form_group.FormDef(name, form_class, required=True, kwargs=None)`

Bases: `object`

instantiate (*prefix, group_initial=None, **extra_kwargs*)

class `shoop.utils.form_group.FormGroup(data=None, files=None, auto_id='id_%s', prefix=None, initial=None)`

Bases: `object`

add_form_def (*name, form_class, required=True, kwargs=None*)

instantiate_forms ()

forms

full_clean ()

errors

Returns an `ErrorDict` for the data provided for the form

is_valid ()

shoop.utils.forms module

`shoop.utils.forms.merged_initial_and_data(form)`

`shoop.utils.forms.get_effective_form_data(form)`

Return 'effective' data for the form, in essence running its validation methods, but trying to return its state to what it was before the `full_clean`. Not fool- proof, but what in this universe is.

Returns data dict

`shoop.utils.forms.fill_model_instance(instance, data)`

Fill whatever fields possible in `instance` using the data dict. :param instance: :param data:

shoop.utils.http module

`shoop.utils.http.retry_request(n_retries=5, **kwargs)`

Retry a `requests` request with exponential backoff up to `n_retries` times.

Parameters

- **n_retries** (*int*) – Maximum retries
- **kwargs** – Kwargs to pass to `requests.request()`.

Returns Requests response.

Return type `requests.Response`

shoop.utils.i18n module

`shoop.utils.i18n.get_babel_locale`

Parse a Django-format (dash-separated) locale string and return a Babel locale.

This function is decorated with `lru_cache`, so executions should be cheap even if `babel.Locale.parse()` most definitely is not.

Parameters **locale_string** (*str*) – A locale string (“en-US”, “fi-FI”, “fi”)

Returns Babel Locale

Return type `babel.Locale`

`shoop.utils.i18n.get_current_babel_locale(fallback='en-US-POSIX')`

Get a Babel locale based on the thread’s locale context.

Parameters **fallback** – Locale to fallback to; set to `None` to raise an exception instead.

Returns Babel Locale

Return type `babel.Locale`

`shoop.utils.i18n.format_percent(value, digits=0)`

`shoop.utils.i18n.get_locally_formatted_datetime(datetime)`

Return a formatted, localized version of `datetime` based on the current context.

`shoop.utils.i18n.format_money(amount, digits=None, widen=0, locale=None)`

Format a Money object in the given locale.

If neither `digits` or `widen` is passed, the preferred number of digits for the amount’s currency is used.

Parameters

- **amount** (*Money*) – The Money object to format
- **digits** (*int | None*) – How many digits to format the currency with.
- **widen** (*int | None*) – How many digits to widen any existing decimal width with.
- **locale** (*Locale | str*) – Locale object or locale identifier

Returns Formatted string

Return type *str*

```
shoop.utils.i18n.get_language_name(language_code)
```

Get a language's name in the currently active locale.

Parameters `language_code` (*str*) – Language code (possibly with an added script suffix (zh_Hans, zh-Hans))

Returns The language name, or the code if the language couldn't be derived.

Return type *str*

```
shoop.utils.i18n.javascript_catalog_all(request, domain='djangojs')
```

Get JavaScript message catalog for all apps in INSTALLED_APPS.

```
shoop.utils.i18n.get_currency_name(currency)
```

shoop.utils.importing module

```
shoop.utils.importing.load(specification, context_explanation='Load')
```

```
shoop.utils.importing.clear_load_cache()
```

```
shoop.utils.importing.cached_load(setting_name, default_value=None)
```

shoop.utils.iterables module

```
shoop.utils.iterables.first(iterable, default=None)
```

Get the first item from the iterable, if possible, or return default.

The iterable is, naturally, iterated for one value.

Parameters

- **iterable** (*Iterable*) – An iterable.
- **default** (*object*) – Default value

Returns The first item from the iterable, or default

Return type *object*

```
shoop.utils.iterables.batch(iterable, count)
```

Yield batches of count items from the given iterable.

```
>>> tuple(x for x in batch([1, 2, 3, 4, 5, 6, 7], 3))
([1, 2, 3], [4, 5, 6], [7])
```

Parameters

- **iterable** (*Iterable*) – An iterable
- **count** (*int*) – Number of items per batch. If <= 0, nothing is yielded.

Returns Iterable of lists of items

Return type *Iterable[list[object]]*

shoop.utils.models module

```
shoop.utils.models.get_data_dict(obj)
```

```
shoop.utils.models.copy_model_instance(obj)
```

shoop.utils.money module**class** `shoop.utils.money.Money`Bases: `shoop.utils.numbers.UnittedDecimal`

Money value with currency.

The pure decimal value is available from the base classes `value` property (preferred way) or by casting to `Decimal`.

Money objects with different currencies cannot be compared or calculated with and will raise `UnitMixupError`.

See `__new__`.

Create new Money instance with given value and currency.

If no currency is given explicitly and `value` has a property named `currency`, then that will be used. Otherwise currency is a required argument and not passing one will raise a `TypeError`.

Parameters

- **value** (`str/numbers.Number`) – Value as string or number
- **currency** (`str/None`) – Currency as ISO-4217 code (3-letter string) or None.

classmethod `from_data (value, currency)`**unit_matches_with** (`other`)**new** (`value`)**shoop.utils.multilanguage_model_form module**`shoop.utils.multilanguage_model_form.to_language_codes (languages)`**class** `shoop.utils.multilanguage_model_form.MultiLanguageModelForm (**kwargs)`Bases: `parler.forms.TranslatableModelForm`**base_fields** = `OrderedDict()`**save** (`commit=True`)**pre_master_save** (`instance`)**declared_fields** = `OrderedDict()`**media****shoop.utils.numbers module**`shoop.utils.numbers.bankers_round (value, ndigits=0)``shoop.utils.numbers.strip_non_float_chars (s)`

Strips characters that aren't part of normal floats.

`shoop.utils.numbers.parse_decimal_string (s)`

Parse decimals with “best effort”.

Parses a string (or unicode) that may be embellished with spaces and other weirdness into the most probable `Decimal`.

Parameters **s** (`str`) – Input value**Returns** `Decimal`**Return type** `Decimal``shoop.utils.numbers.try_parse_decimal_string (s)`

`shoop.utils.numbers.get_string_sort_order(s)`

Return a sorting order value for a string that contains a garment size.

Parameters `s` (*str*) – Input value (string or number)

Returns Sorting tuple

Return type `tuple`

class `shoop.utils.numbers.UnittedDecimal`

Bases: `decimal.Decimal`

Decimal with unit.

Allows creating decimal classes that cannot be mixed, e.g. to prevent operations like:

`TaxfulPrice(1) + TaxlessPrice(2)`

where `TaxfulPrice` and `TaxlessPrice` are subclasses of `UnittedDecimal`.

copy_negate (**args, **kwargs*)

new (*value*)

Create new instance with given value using same unit as self.

Post-condition: If `x = y.new(v)`, then `x.unit_matches_with(y)` and `x.value == v`.

Returns Object with same type as self and matching unit, but with given decimal value

Return type `UnittedDecimal`

quantize (*exp, *args, **kwargs*)

unit_matches_with (*other*)

Test if self and other have matching units.

Return type `bool`

value

Value of this decimal without the unit.

Return type `decimal.Decimal`

exception `shoop.utils.numbers.UnitMixupError` (*obj1, obj2, msg='Unit mixup'*)

Bases: `TypeError`

Invoked operation for `UnittedDecimal` and object with non-matching unit.

The objects involved are stored in instance variables `obj1` and `obj2`. Former is instance of `UnittedDecimal` or its subclass and the other could be any object.

Variables

- **obj1** (`UnittedDecimal`) – Involved object 1
- **obj2** (*Any*) – Involved object 2

shoop.utils.objects module

`shoop.utils.objects.extract_inner_value` (*source, attr_chain*)

Search for a stored value by recursing through dict keys and attributes. Erroneous/missing keys/attribute names will return `None`.

Parameters

- **source** – The original object, that either has attributes or is itself a dict.

- **attr_chain** (*tuple*) – A tuple of properly ordered strings, containing the attributes and/or keys to successively obtain.

```
>>> mydict = {"foo": {"bar": {"thing": 25}}}
>>> extract_inner_value(mydict, ("foo", "bar", "thing"))
25
>>> extract_inner_value(mydict, ("foo", "bar", "unthing"))
>>> bool(extract_inner_value(mydict, ("__getitem__",)))
True
>>> bool(extract_inner_value(mydict, ("__getotem__",)))
False
```

shoop.utils.objects.**compare_partial_dicts** (*source*, *comparee*)

Compare dicts in a “partial” manner. All key/value pairs in *source* must exist and be equal to those in *comparee*.

This differs from a raw `==` in that keys that do not exist in *source* may exist in *comparee*.

Parameters

- **source** (*dict*) – source dict
- **comparee** (*dict*) – comparee dict

Return type `bool`

Returns True or False

shoop.utils.objects.**compact** (*in_obj*, *none_only=True*, *deep=True*)

Compact iterable by removing falsy values.

Iterable may be a mapping or a list.

By default uses `not value` to test for falseness, but if *none_only* is set, will use `value is None`.

By default, iterables within the iterable are also compacted. This can be controlled by the *deep* argument.

Parameters

- **in_obj** (*Iterable*) – The object to compact
- **none_only** (*bool*) – Remove only Nones
- **deep** (*bool*) – Recurse through iterables within *in_obj*

Returns Flattened iterable

Return type `listdict`

shoop.utils.patterns module

class shoop.utils.patterns.**Pattern** (*pattern_text*)

Bases: `object`

Compile a pattern from the given *pattern_text*.

Patterns are comma-separated atoms of the forms:

- `*` – matches anything
- `text` – matched directly
- `min-max` – inclusive range matched lexicographically OR as integers if possible
- `wild*` – wildcards (asterisks and question marks allowed)

In addition, atoms may be prefixed with `!` to negate them.

For instance, “10-20,!15” would match all strings (or numbers) between 10 and 20, but not 15.

matches (*target*)

Evaluate this Pattern against the target.

Return type `bool`

get_alphabetical_limits ()

as_normalized ()

Return the pattern’s source text in a “normalized” form.

Return type `str`

`shoop.utils.patterns.pattern_matches` (*pattern*, *target*)

Verify that a target string matches the given pattern.

For pattern strings, compiled patterns are cached.

Parameters

- **pattern** (*str/Pattern*) – The pattern. Either a pattern string or a Pattern instance
- **target** (*str*) – Target string to test against.

Returns Whether the test succeeded.

Return type `bool`

shoop.utils.properties module

class `shoop.utils.properties.MoneyProperty` (*value*, *currency*)

Bases: `object`

Property for a Money amount.

Will return Money objects when the property is being get and accepts Money objects on set. Value and currency are read/written from/to other fields.

Fields are given as locators, that is a string in dotted format, e.g. locator “foo.bar” points to `instance.foo.bar` where `instance` is an instance of the class owning the `MoneyProperty`.

Setting value of this property to a Money object with different currency that is currently set (in the field pointed by the currency locator), will raise an `UnitMixupError`.

Initialize MoneyProperty with given field locators.

Parameters

- **value** (*str*) – Locator for value of the Money
- **currency** (*str*) – Locator for currency of the Money

value_class

alias of Money

class `shoop.utils.properties.PriceProperty` (*value*, *currency*, *includes_tax*, ***kwargs*)

Bases: `shoop.utils.properties.MoneyProperty`

Property for Price object.

Similar to `MoneyProperty` but also has `includes_tax` field.

Operators with `TaxfulPrice` and `TaxlessPrice` objects.

Initialize PriceProperty with given field locators.

Parameters

- **value** (*str*) – Locator for value of the Price
- **currency** (*str*) – Locator for currency of the Price
- **includes_tax** (*str*) – Locator for includes_tax of the Price

value_class

alias of Price

class `shoop.utils.properties.TaxfulPriceProperty` (*value*, *currency*)
Bases: `shoop.utils.properties.MoneyProperty`

Initialize MoneyProperty with given field locators.

Parameters

- **value** (*str*) – Locator for value of the Money
- **currency** (*str*) – Locator for currency of the Money

value_class

alias of TaxfulPrice

class `shoop.utils.properties.TaxlessPriceProperty` (*value*, *currency*)
Bases: `shoop.utils.properties.MoneyProperty`

Initialize MoneyProperty with given field locators.

Parameters

- **value** (*str*) – Locator for value of the Money
- **currency** (*str*) – Locator for currency of the Money

value_class

alias of TaxlessPrice

class `shoop.utils.properties.MoneyPropped` (**args*, ***kwargs*)
Bases: `object`

Mixin for transforming MoneyProperty init parameters.

Add this mixin as (first) base for the class that has `MoneyProperty` properties and this will make its `__init__` transform passed kwargs to the fields specified in the `MoneyProperty`.

`shoop.utils.properties.resolve` (*obj*, *path*)
Resolve a locator path starting from object *obj*.

shoop.utils.serialization module

class `shoop.utils.serialization.ExtendedJSONEncoder` (*skipkeys=False*, *ensure_ascii=True*,
check_circular=True, *al-*
low_nan=True, *sort_keys=False*,
indent=None, *separators=None*,
default=None)

Bases: `django.core.serializers.json.DjangoJSONEncoder`

Constructor for JSONEncoder, with sensible defaults.

If *skipkeys* is false, then it is a `TypeError` to attempt encoding of keys that are not str, int, float or None. If *skipkeys* is True, such items are simply skipped.

If `ensure_ascii` is true, the output is guaranteed to be str objects with all incoming non-ASCII characters escaped. If `ensure_ascii` is false, the output can contain non-ASCII characters.

If `check_circular` is true, then lists, dicts, and custom encoded objects will be checked for circular references during encoding to prevent an infinite recursion (which would cause an `OverflowError`). Otherwise, no such check takes place.

If `allow_nan` is true, then NaN, Infinity, and -Infinity will be encoded as such. This behavior is not JSON specification compliant, but is consistent with most JavaScript based encoders and decoders. Otherwise, it will be a `ValueError` to encode such floats.

If `sort_keys` is true, then the output of dictionaries will be sorted by key; this is useful for regression tests to ensure that JSON serializations can be compared on a day-to-day basis.

If `indent` is a non-negative integer, then JSON array elements and object members will be pretty-printed with that indent level. An indent level of 0 will only insert newlines. None is the most compact representation.

If specified, separators should be an (item_separator, key_separator) tuple. The default is (', ', ': ') if `indent` is None and (',', ': ') otherwise. To get the most compact JSON representation, you should specify (',', ':') to eliminate whitespace.

If specified, default is a function that gets called for objects that can't otherwise be serialized. It should return a JSON encodable version of the object or raise a `TypeError`.

default (*o*)

shoop.utils.settings_doc module

```
shoop.utils.settings_doc.get_known_settings_documentation (order_by='app',
                                                            only_changed=False)
shoop.utils.settings_doc.get_known_settings_with_comments ()
```

shoop.utils.setup module

```
class shoop.utils.setup.Setup (load_from=None)
    Bases: object

    is_valid_key (key)

    commit (source)

    values ()

    get (key, default=None)

    getlist (key, default=())

    classmethod configure (configure)
```

shoop.utils.text module

```
shoop.utils.text.flatten (str, whitespace='-')
```

Flatten the given text into lowercase ASCII, removing diacriticals etc. Replace runs of whitespace with the given whitespace replacement.

```
>>> print(flatten("hellö, wörld"))
hello,-world
```

Parameters

- **str** (*str*) – The string to massage
- **whitespace** (*str*) – The string to replace whitespace with

Returns A flattened string

Return type `str`

`shoop.utils.text.identifierify(value, sep='_')`

Identifierify the given text (keep only alphanumerics and the given separator(s).

Parameters

- **value** (`str`) – The text to identifierify
- **sep** (`str`) – The separator(s) to keep

Returns An identifierified string

Return type `str`

`shoop.utils.text.snake_case(value)`

Snake_case the given value (join words with underscores). No other treatment is done; use `identifierify` for that.

`shoop.utils.text.kebab_case(value)`

Kebab-case the given value (join words with dashes). No other treatment is done; use `identifierify` for that.

`shoop.utils.text.camel_case(value)`

CamelCase the given value (join capitalized words). No other treatment is done; use `identifierify` for that.

`shoop.utils.text.space_case(value)`

Space case the given value (join words that may have been otherwise separated with spaces). No other treatment is done; use `identifierify` for that.

shoop.utils.translation module

`shoop.utils.translation.cache_translations(objects, languages=None, meta=None)`

Cache translation objects in given languages to the objects in one fell swoop. This will iterate a queryset, if one is passed!

Parameters

- **objects** – List or queryset of Translatable models
- **languages** – Iterable of languages to fetch. In addition, all “_current_language”s will be fetched

Returns objects

`shoop.utils.translation.cache_translations_for_tree(root_objects, languages=None)`

Cache translation objects in given languages, iterating MPTT trees.

Parameters

- **root_objects** (`Iterable[model]`) – List of MPTT models
- **languages** (`Iterable[str]`) – List of languages

Module contents

`shoop.utils.update_module_attributes(object_names, module_name)`

Update `__module__` attribute of objects in module.

Set the `__module__` attribute of the objects (resolved by the given object names from the given module name) to `module_name`.

Use case for this function in Shoop is to hide the actual location of objects imported from private submodules, so that they will show up nicely in the Sphinx generated API documentation. This is done by appending following line to the end of the `__init__.py` of the main package:

```
update_module_attributes(__all__, __name__)
```

Parameters

- **object_names** (*Iterable[str]*) – Names of the objects to update.
- **module_name** (*str*) – Name of the module where the objects reside and also the new value which will be assigned to `__module__` attribute of each object.

```
class shoop.utils.ShoopUtilsAppConfig(*args, **kwargs)
    Bases: shoop.apps.AppConfig
    name = 'shoop.utils'
    verbose_name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    label = 'shoop_utils'
```

shoop.xtheme package

Subpackages

shoop.xtheme.admin_module package

Submodules

shoop.xtheme.admin_module.views module

```
class shoop.xtheme.admin_module.views.ActivationForm(data=None, files=None,
    auto_id='id_%s', prefix=None,
    initial=None, error_class=<class
    'django.forms.utils.ErrorList'>,
    label_suffix=None,
    empty_permitted=False)

    Bases: django.forms.forms.Form
    A very simple form for activating a theme.
    base_fields = OrderedDict([('activate', <django.forms.fields.CharField object at 0x7f7137b12a90>)])
    declared_fields = OrderedDict([('activate', <django.forms.fields.CharField object at 0x7f7137b12a90>)])
    media
class shoop.xtheme.admin_module.views.ThemeConfigView(**kwargs)
    Bases: django.views.generic.edit.FormView
    A view for listing and activating themes.
    Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.
    template_name = 'shoop/xtheme/admin/config.jinja'
    form_class
        alias of ActivationForm
```

```
get_context_data (**kwargs)
```

```
form_valid (form)
```

```
class shoop.xtheme.admin_module.views.ThemeConfigDetailView (**kwargs)
```

Bases: *shoop.admin.utils.views.CreateOrUpdateView*

A view for configuring a single theme.

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

```
model
```

alias of ThemeSettings

```
template_name = 'shoop/xtheme/admin/config_detail.jinja'
```

```
form_class
```

alias of Form

```
context_object_name = 'theme_settings'
```

```
add_form_errors_as_messages = True
```

```
get_object (queryset=None)
```

```
get_theme ()
```

Get the theme object to configure.

Returns Theme object

Return type *shoop.xtheme.Theme*

```
get_context_data (**kwargs)
```

```
get_form (form_class=None)
```

```
get_success_url ()
```

Module contents

```
class shoop.xtheme.admin_module.XthemeAdminModule
```

Bases: *shoop.admin.base.AdminModule*

Admin module for Xtheme.

Allows theme activation/deactivation and further configuration.

```
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
```

```
breadcrumbs_menu_entry = <shoop.admin.base.MenuEntry object>
```

```
get_urls ()
```

```
get_menu_category_icons ()
```

```
get_menu_entries (request)
```

```
get_required_permissions ()
```

```
get_notifications (request)
```

shoop.xtheme.plugins package

Submodules

shoop.xtheme.plugins.category_links module

class `shoop.xtheme.plugins.category_links.CategoryLinksConfigForm` (***kwargs*)

Bases: `shoop.xtheme.plugins.forms.GenericPluginForm`

A configuration form for the CategoryLinksPlugin

populate ()

A custom populate method to display category choices

clean ()

A custom clean method to save category configuration information in a serializable form

base_fields = `OrderedDict()`

declared_fields = `OrderedDict()`

media

class `shoop.xtheme.plugins.category_links.CategoryLinksPlugin` (*config*)

Bases: `shoop.xtheme.TemplatedPlugin`

A plugin for displaying links to visible categories on the shop front

Instantiate a Plugin with the given `config` dictionary.

Parameters `config` (*dict*) – Dictionary of freeform configuration data

identifier = `'category_links'`

name = `<django.utils.functional.lazy.<locals>.__proxy__ object>`

template_name = `'shoop/xtheme/plugins/category_links.jinja'`

editor_form_class

alias of `CategoryLinksConfigForm`

fields = [`('title', <shoop.xtheme.plugins.forms.TranslatableField object at 0x7f71378f2c88>)`, `('show_all_categories', <d`

get_context_data (*context*)

A custom `get_context_data` method to return only visible categories for request customer.

shoop.xtheme.plugins.consts module

`shoop.xtheme.plugins.consts.FALLBACK_LANGUAGE_CODE` = `'*'`

The pseudo-language code used by `TranslatedFields` and the relevant Plugin API to mark the untranslated/fallback content

shoop.xtheme.plugins.forms module

class `shoop.xtheme.plugins.forms.PluginForm` (***kwargs*)

Bases: `django.forms.forms.Form`

Base class for plugin configuration forms.

populate ()

get_config ()

Get the new `config` dict for a plugin.

Called when the form is valid, akin to `django.forms.models.ModelForm.save`.

The default implementation just augments the old `config` with the cleaned data for the form.

Returns A new JSONable (!) `config` dict

Return type `dict`

```
base_fields = OrderedDict()
declared_fields = OrderedDict()
media
class shoop.xtheme.plugins.forms.GenericPluginForm(**kwargs)
    Bases: shoop.xtheme.plugins.forms.PluginForm
    A generic form for Xtheme plugins; populates itself based on fields in the plugin class.
    populate()
    base_fields = OrderedDict()
    declared_fields = OrderedDict()
    media
class shoop.xtheme.plugins.forms.TranslatableField(*args, **kwargs)
    Bases: django.forms.fields.Field
    widget
        alias of TranslatableFieldWidget
    clean(value)
```

shoop.xtheme.plugins.image module

```
class shoop.xtheme.plugins.image.ImagePluginChoiceWidget(attrs=None, clear-
                                                         able=False,
                                                         empty_text='—')
    Bases: shoop.admin.forms.widgets.ImageChoiceWidget
    Subclass of ImageChoiceWidget that will not raise an exception if given an invalid initial image ID (in case the
    image has been deleted)
```

```
get_object(value)
```

```
media
```

```
class shoop.xtheme.plugins.image.ImageIDField(max_value=None, min_value=None, *args,
**kwargs)
    Bases: django.forms.fields.IntegerField
    A custom field that stores the ID value of a Filer image and presents Shoop admin's image popup widget
    widget = <shoop.xtheme.plugins.image.ImagePluginChoiceWidget object>
    clean(value)
```

```
class shoop.xtheme.plugins.image.ImagePlugin(config)
    Bases: shoop.xtheme.TemplatedPlugin
```

A linkable image plugin

Instantiate a Plugin with the given config dictionary.

Parameters config (*dict*) – Dictionary of freeform configuration data

```
identifier = 'images'
```

```
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
```

```
template_name = 'shoop/xtheme/plugins/image.jinja'
```

```
fields = [('title', <shoop.xtheme.plugins.forms.TranslatableField object at 0x7f7137981dd8>), ('image_id', <shoop.xtheme.plugins.image.ImageIDField object at 0x7f7137981dd8>)]
```

get_context_data (*context*)

A custom get_context_data that returns the matching filer File

shoop.xtheme.plugins.snippets module

class shoop.xtheme.plugins.snippets.**SnippetsPlugin** (*config*)

Bases: *shoop.xtheme.Plugin*

Simple plugin class for including snippets and resources on the page, mostly for simple integrations.

Instantiate a Plugin with the given *config* dictionary.

Parameters *config* (*dict*) – Dictionary of freeform configuration data

identifier = 'snippets'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

fields = [(('in_place', <django.forms.fields.CharField object at 0x7f7137a40518>), ('head_end', <django.forms.fields.Ch

render (*context*)

shoop.xtheme.plugins.social_media_links module

class shoop.xtheme.plugins.social_media_links.**SocialMediaLinksPluginForm** (***kwargs*)

Bases: *shoop.xtheme.plugins.forms.GenericPluginForm*

Form for the social media links xtheme plugin. One field is provided for each entry in the plugin's *icon_classes* attribute, which maps social media site names to font-awesome icon classes by default.

populate ()

Populates form with default plugin fields as well as any social media link type included in the plugin's *icon_classes* attribute.

Also adds an ordering field for each link type to change display order.

clean ()

Returns cleaned data from default plugin fields and any link fields.

Processed link configuration information is stored and returned as a dictionary (*links*).

base_fields = *OrderedDict*()

declared_fields = *OrderedDict*()

media

class shoop.xtheme.plugins.social_media_links.**SocialMediaLinksPlugin** (*config*)

Bases: *shoop.xtheme.TemplatedPlugin*

An xtheme plugin for displaying site links to common social media sites.

Instantiate a Plugin with the given *config* dictionary.

Parameters *config* (*dict*) – Dictionary of freeform configuration data

identifier = 'social_media_links'

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

template_name = 'shoop/xtheme/plugins/social_media_links.jinja'

editor_form_class

alias of *SocialMediaLinksPluginForm*

fields = [(('topic', <shoop.xtheme.plugins.forms.TranslatableField object at 0x7f713fbb7160>), ('text', <shoop.xtheme.p

icon_classes = {'Youtube': 'youtube', 'Twitter': 'twitter', 'Yelp': 'yelp', 'Flickr': 'flickr', 'Tumbler': 'tumblr', 'Pinte

get_context_data (*context*)

Returns plugin settings and a sorted list of social media links

Returns Plugin context data

Return type `dict`

get_links ()

Returns the list of social media links sorted according to ordering

Returns List of link tuples (ordering, icon class, url)

Return type [(int, str, str)]

shoop.xtheme.plugins.text module

class shoop.xtheme.plugins.text.**TextPlugin** (*config*)

Bases: `shoop.xtheme.Plugin`

Very basic Markdown rendering plugin.

Instantiate a Plugin with the given `config` dictionary.

Parameters **config** (*dict*) – Dictionary of freeform configuration data

identifier = 'text'

name = 'Text'

fields = [('text', <shoop.xtheme.plugins.forms.TranslatableField object at 0x7f7137ce7c50>)]

render (*context*)

shoop.xtheme.plugins.widgets module

class shoop.xtheme.plugins.widgets.**TranslatableFieldWidget** (*languages*, *input_widget*=<class 'django.forms.widgets.TextInput'>, *attrs*=None)

Bases: `django.forms.widgets.Widget`

render (*name*, *value*, *attrs*=None)

id_for_label (*id_*)

value_from_datadict (*data*, *files*, *name*)

format_output (*widget_pairs*)

decompress (*value*)

media

class shoop.xtheme.plugins.widgets.**XThemeModelChoiceWidget** (*attrs*=None, *choices*=())

Bases: `django.forms.widgets.Select`

render (*name*, *value*, *attrs*=None, *choices*=())

media


```

class shoop.xtheme.plugins.widgets.XThemeModelChoiceField(queryset,
                                                         empty_label='———',
                                                         cache_choices=None,
                                                         required=True,      wid-
                                                         get=None,   label=None,
                                                         initial=None, help_text='',
                                                         to_field_name=None,
                                                         limit_choices_to=None,
                                                         *args, **kwargs)

Bases: django.forms.models.ModelChoiceField

widget
    alias of XThemeModelChoiceWidget

label_from_instance (obj)

```

Module contents

shoop.xtheme.templatetags package

Submodules

shoop.xtheme.templatetags.xtheme_tags module

Module contents

shoop.xtheme.views package

Submodules

shoop.xtheme.views.command module

shoop.xtheme.views.command.**handle_command**(*request*, *command*)
Internal dispatch function.

Parameters

- **request** (*django.http.HttpRequest*) – A request
- **command** (*str*) – Command string

Returns A response

Return type *django.http.HttpResponse*

shoop.xtheme.views.command.**command_dispatch**(*request*)
Xtheme command dispatch view.

Parameters **request** (*django.http.HttpRequest*) – A request

Returns A response

Return type *django.http.HttpResponse*

shoop.xtheme.views.editor module**class** shoop.xtheme.views.editor.**EditorView** (**kwargs)Bases: `django.views.generic.base.TemplateView`

Constructor. Called in the URLconf; can contain helpful extra keyword arguments, and other things.

template_name = 'shoop/xtheme/editor.jinja'**xtheme_injection** = False**changed** = False**get_context_data** (**kwargs)**dispatch** (request, *args, **kwargs)**post** (request, *args, **kwargs)**build_form** ()**save_layout** (layout=None)**dispatch_add_cell** (y, **kwargs)**dispatch_add_row** (y=None, **kwargs)**dispatch_del_row** (y, **kwargs)**dispatch_del_cell** (x, y, **kwargs)**dispatch_change_plugin** (plugin=', **kwargs)**dispatch_publish** (**kwargs)**dispatch_revert** (**kwargs)**shoop.xtheme.views.extra module**shoop.xtheme.views.extra.**clear_view_cache** (**kwargs)shoop.xtheme.views.extra.**get_view_by_name** (theme, view_name)shoop.xtheme.views.extra.**extra_view_dispatch** (request, view)

Dispatch to an Xtheme extra view.

Parameters

- **request** (`django.http.HttpRequest`) – A request
- **view** (`str`) – View name

Returns A response of some ilk**Return type** `django.http.HttpResponse`**shoop.xtheme.views.forms module****class** shoop.xtheme.views.forms.**LayoutCellGeneralInfoForm** (**kwargs)Bases: `django.forms.forms.Form`**CELL_FULL_WIDTH** = 12**CELL_WIDTH_CHOICES** = [(12, <django.utils.functional.lazy.<locals>.__proxy__ object at 0x7f713e2aa048>), (9, <django**populate** ()

Populate the form with fields for size and plugin selection.

save()

Save size configuration. Plugin configuration is done via JavaScript POST.

Both breakpoints (sm and md) are set to same value defined in `cell_width_field`. The reason for this is that the difference between these breakpoints is so minor that manually assigning both of these by shop admin introduces too much complexity to row-cell management UI.

base_fields = `OrderedDict([('plugin', <django.forms.fields.ChoiceField object at 0x7f713e31c978>)])`

declared_fields = `OrderedDict([('plugin', <django.forms.fields.ChoiceField object at 0x7f713e31c978>)])`

media

class `shoop.xtheme.views.forms.LayoutCellFormGroup` (**kwargs)

Bases: `shoop.utils.form_group.FormGroup`

Form group containing the LayoutCellGeneralInfoForm and a possible plugin-dependent configuration form.

save()

Module contents

Submodules

shoop.xtheme.editing module

`shoop.xtheme.editing.could_edit(request)`

Return true if the context of the given request would allow Xtheme editing.

Parameters **request** (`django.http.HttpRequest`) – HTTP request

Returns Would allow editing?

Return type `bool`

`shoop.xtheme.editing.is_edit_mode(request)`

Return true if the given request has xtheme editing enabled.

Parameters **request** (`django.http.HttpRequest`) – HTTP request

Returns In edit mode?

Return type `bool`

`shoop.xtheme.editing.set_edit_mode(request, flag)`

Enable or disable edit mode for the request.

Parameters

- **request** (`django.http.HttpRequest`) – HTTP request
- **flag** (`bool`) – Enable flag

`shoop.xtheme.editing.may_inject(context)`

Figure out if we may inject Xtheme editing into this view.

The requirements are that there is a CBV view object in the context, and that view object does not explicitly opt-out of editing with `xtheme_injection = False`

Parameters **context** (`jinja2.runtime.Context`) – Jinja rendering context

Returns Permission bool

Return type `bool`

`shoop.xtheme.editing.add_edit_resources` (*context*)

Possibly inject Xtheme editor injection resources into the given context's resources.

Parameters `context` (*jinja2.runtime.Context*) – Jinja rendering context

shoop.xtheme.engine module

class `shoop.xtheme.engine.XthemeTemplate`

Bases: `jinja2.environment.Template`

A subclass of Jinja templates with additional post-processing magic.

render (**args, **kwargs*)

Render the template and postprocess it.

Returns Rendered markup

Return type `str`

```
class shoop.xtheme.engine.XthemeEnvironment (block_start_string='{%',
                                             block_end_string='%}',
                                             variable_start_string '{{',
                                             variable_end_string='}}',
                                             comment_start_string='{#',
                                             comment_end_string='#}',
                                             line_statement_prefix=None,
                                             line_comment_prefix=None, trim_blocks=False,
                                             lstrip_blocks=False, newline_sequence='n',
                                             keep_trailing_newline=False, extensions=(),
                                             optimized=True, undefined=<class
                                             'jinja2.runtime.Undefined'>, finalize=None,
                                             autoescape=False, loader=None, cache_size=400,
                                             auto_reload=True, bytecode_cache=None)
```

Bases: `jinja2.environment.Environment`

Overrides the usual template class and allows dynamic switching of Xthemes.

Enable by adding "environment": "shoop.xtheme.engine.XthemeEnvironment" in your TEMPLATES settings.

template_class

alias of *XthemeTemplate*

get_template (*name, parent=None, globals=None*)

Load a template from the loader. If a loader is configured this method asks the loader for the template and returns a *Template*.

Parameters

- **name** (*str*) – Template name.
- **parent** (*str/None*) – If the parent parameter is not *None*, `join_path` is called to get the real template name before loading.
- **globals** (*dict/None*) – The `globals` parameter can be used to provide template wide globals. These variables are available in the context at render time.

Returns Template object

Return type *shoop.xtheme.engine.XthemeTemplate*

get_or_select_template (*template_name_or_list, parent=None, globals=None*)

Does a typecheck and dispatches to `select_template` or *get_template*.

Parameters

- **template_name_or_list** (*str*/*Iterable[str]*) – Template name or list
- **parent** (*str*/*None*) – If the `parent` parameter is not `None`, `join_path` is called to get the real template name before loading.
- **globals** – The `globals` parameter can be used to provide template wide globals. These variables are available in the context at render time.

Returns Template object**Return type** *shoop.xtheme.engine.XthemeTemplate*

```
shoop.xtheme.engine.concat()
S.join(iterable) -> str
```

Return a string which is the concatenation of the strings in the iterable. The separator between elements is S.

shoop.xtheme.forms module

```
class shoop.xtheme.forms.GenericThemeForm(**kwargs)
```

Bases: `django.forms.models.ModelForm`

A generic form for Xthemes; populates itself based on `fields` in the theme class.

```
class Meta
```

Bases: `object`

```
model
```

alias of `ThemeSettings`

```
fields = ()
```

```
GenericThemeForm.base_fields = OrderedDict()
```

```
GenericThemeForm.declared_fields = OrderedDict()
```

```
GenericThemeForm.media
```

```
GenericThemeForm.save(commit=True)
```

Save theme settings into the `ThemeSettings` instance

Parameters **commit** (*bool*) – Commit flag. Ignored, but there for compatibility with the superclass.

Returns The now saved `ThemeSettings` instance

Return type *shoop.xtheme.models.ThemeSettings*

shoop.xtheme.layout module

```
class shoop.xtheme.layout.LayoutCell(plugin_identifier, config=None, sizes=None)
```

Bases: `object`

A single cell in a layout. Maps to Bootstrap's `col-XX-XX` classes.

Initialize a layout cell with a given plugin, config and sizing configuration.

Parameters

- **plugin_identifier** (*str*) – Plugin identifier string
- **config** (*dict*/*None*) – Config dict
- **sizes** (*dict*/*None*) – Size dict

plugin_class

Get the actual plugin class for this cell, or None if the plugin class isn't available.

Returns Plugin or None.

Return type Plugin|None

plugin_name

Get the name of the plugin in this cell for display purposes.

Returns Plugin name string

Return type str

instantiate_plugin()

Instantiate the plugin with the current config.

Returns Instantiated plugin (if a class is available)

Return type Plugin|None

render(context)

Return the plugin's rendered contents.

Parameters context (*jinja2.runtime.Context*) – Jinja2 rendering context.

Returns string of content

Return type str

classmethod unserialize(data)

Unserialize a dict of layout cell data into a new cell.

Parameters data (*dict*) – Layout cell data dict

Returns New cell

Return type *LayoutCell*

serialize()

Serialize this cell into a dict.

Returns Layout cell data dict

Return type dict

class shoop.xtheme.layout.**LayoutRow**(cells=None)

Bases: *object*

A single row in a layout. Maps to Bootstrap's row class.

Parameters cells – Optional iterable of LayoutCells to populate this LayoutRow with.

classmethod unserialize(data)

Unserialize a dict of layout row data into a new row, along with all cell children.

Parameters data (*dict*) – Layout row data dict

Returns New row

Return type *LayoutRow*

serialize()

Serialize this row into a dict.

Returns Layout row data dict

Return type dict

add_cell (*sizes=None*)

Add an empty cell to this row. Used by the editor API.

Parameters **sizes** (*dict* | *None*) – An optional size dict, see [LayoutCell](#)

Returns The new layout cell

Return type [LayoutCell](#)

class `shoop.xtheme.layout.Layout` (*placeholder_name, rows=None*)

Bases: [object](#)

The layout (row, cell and plugin configuration) for a single placeholder.

Parameters

- **placeholder_name** (*str* | *None*) – The name of the placeholder. Could be None.
- **rows** (*Iterable*[[LayoutRow](#)] | *None*) – Optional iterable of [LayoutRows](#) to populate this Layout with.

row_class = 'row'

cell_class_template = 'col-%(breakpoint)s-%(width)s'

hide_cell_class_template = 'hidden-%(breakpoint)s'

classmethod **unserialize** (*data, placeholder_name=None*)

Unserialize a dict of layout data into a new layout, with all rows and cells.

Parameters

- **data** (*dict*) – Layout data dict
- **placeholder_name** (*str*) – Placeholder name if none is specified in the data

Returns New layout

Return type [Layout](#)

serialize ()

Serialize this layout into a dict.

Returns Layout data dict

Return type [dict](#)

begin_row ()

Begin a new row in the layout.

This is internally used by [LayoutPartExtension](#), but could just as well be used to programmatically create layouts for whichever purpose.

Returns The newly created row

Return type [LayoutRow](#)

begin_column (*sizes=None*)

Begin a new column (cell) in the layout, in the last row.

If no rows exist, one is implicitly created, for your convenience. The newly created cell has no plugin or configuration.

This is internally used by [LayoutPartExtension](#), but could just as well be used to programmatically create layouts for whichever purpose.

Parameters **sizes** – The size dictionary to pass to [LayoutCell](#).

Returns The newly created cell

Return type *LayoutCell*

add_plugin (*plugin_identifier*, *config*)

Configure a plugin in the last row and cell of the layout.

If no rows or cells exist, one row and one cell is implicitly created.

This is internally used by `LayoutPartExtension`, but could just as well be used to programmatically create layouts for whichever purpose.

Parameters

- **plugin_identifier** (*str*) – Plugin identifier string
- **config** (*dict*) – Configuration dict

Returns The configured cell

Return type *LayoutCell*

get_cell (*x*, *y*)

Get a layout cell indicated by the given (zero-based) coordinates.

If the coordinates are out of range, returns `None`.

Parameters

- **x** (*int*) – X (horizontal) coordinate
- **y** (*int*) – Y (vertical) coordinate

Returns Layout cell

Return type `LayoutCell|None`

insert_row (*y=None*)

Insert a new row at the given zero-based row and return it.

If *y* is `None`, the row is inserted at the end.

Parameters **y** (*int*) – Y coordinate

Returns The new layout row

Return type *LayoutRow*

delete_row (*y*)

Delete the *y*'th (zero-based) row.

If *y* is out of bounds, nothing is done.

Parameters **y** (*int*) – Y coordinate

Returns Was something done?

Return type `bool`

delete_cell (*x*, *y*)

Delete a layout cell indicated by the given (zero-based) coordinates.

If the coordinates are out of range, nothing is done.

Parameters

- **x** (*int*) – X (horizontal) coordinate
- **y** (*int*) – Y (vertical) coordinate

Returns Was something done?

Return type `bool`

shoop.xtheme.models module

class `shoop.xtheme.models.SavedViewConfigQuerySet` (*model=None, query=None, using=None, hints=None*)

Bases: `django.db.models.query.QuerySet`

appropriate (*theme, view_name, draft*)

Get an “appropriate” *SavedViewConfig* for the parameters given.

When draft mode is off:

- A PUBLIC *SavedViewConfig* is returned, or a new one in CURRENT_DRAFT status.

When draft mode is on:

- A CURRENT_DRAFT *SavedViewConfig* is returned, if one exists.
- If a PUBLIC *SavedViewConfig* exists, its data is copied into a new, unsaved CURRENT_DRAFT *SavedViewConfig*.

Parameters

- **theme** (`shoop.xtheme.Theme`) – Theme instance
- **view_name** (`str`) – View name string
- **draft** (`bool`) – Draft mode flag

Returns *SavedViewConfig* (possibly not saved)

Return type *SavedViewConfig*

class `shoop.xtheme.models.SavedViewConfigStatus`

Bases: `enumfields.enums.Enum`

Stati for *SavedViewConfigs*.

The lifecycle for *SavedViewConfigs* (SVCs) for a given (theme, view) pair is as follows:

- Initially, there’s zero SVCs.
- When a placeholder layout is saved in edit mode, an SVC in the CURRENT_DRAFT status is saved.
- When an SVC in CURRENT_DRAFT status is published, all other SVCs for the theme/view pair are “demoted” to being OLD_VERSIONs and the CURRENT_DRAFT SVC is promoted to being the PUBLIC one (and there should always be zero or one PUBLIC *SavedViewConfigs* per (theme, view) pair).
- When an SVC in CURRENT_DRAFT status is reverted, it is simply deleted.
- When an SVC has been published and edit mode is entered again, the current PUBLIC SVC is copied into a new CURRENT_DRAFT version.

CURRENT_DRAFT = `<SavedViewConfigStatus.CURRENT_DRAFT: 1>`

OLD_VERSION = `<SavedViewConfigStatus.OLD_VERSION: 2>`

PUBLIC = `<SavedViewConfigStatus.PUBLIC: 3>`

class `shoop.xtheme.models.SavedViewConfig` (*id, theme_identifier, view_name, created_on, status, _data*)

Bases: `django.db.models.base.Model`

status

A placeholder class that provides a way to set the attribute on the model.

objects = <django.db.models.manager.ManagerFromSavedViewConfigQuerySet object>

draft

publish()

revert()

set_layout_data (*placeholder_name*, *layout*)

get_layout_data (*placeholder_name*)

clear_layout_data (*placeholder_name*)

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception SavedViewConfig.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`SavedViewConfig.get_next_by_created_on` (**moreargs*, ***morekwargs*)

`SavedViewConfig.get_previous_by_created_on` (**moreargs*, ***morekwargs*)

`SavedViewConfig.get_status_display` (**moreargs*, ***morekwargs*)

class `shoop.xtheme.models.ThemeSettings` (*id*, *theme_identifier*, *active*, *data*)

Bases: `django.db.models.base.Model`

data

A placeholder class that provides a way to set the attribute on the model.

activate()

get_setting (*key*, *default=None*)

get_settings()

update_settings (*update_values*)

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception ThemeSettings.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`ThemeSettings.objects` = <django.db.models.manager.Manager object>

shoop.xtheme.parsing module

exception `shoop.xtheme.parsing.Unflattenable`

Bases: `Exception`

Exception raised when a node list can't be flattened into a constant.

exception `shoop.xtheme.parsing.NonConstant`

Bases: `ValueError`

Exception raised when something expected to be constant... is not.

exception `shoop.xtheme.parsing.NestingError`

Bases: `ValueError`

Exception raised when a template's placeholder/column/row/plugin hierarchy is out of whack.

`shoop.xtheme.parsing.flatten_const_node_list(environment, node_list)`

Try to flatten the given node list into a single string.

Parameters

- **environment** (`jinja2.environment.Environment`) – Jinja2 environment
- **node_list** (`list[jinja2.nodes.Node]`) – List of nodes

Returns String of content

Return type `str`

Raises **Unflattenable** – Raised when the node list can’t be flattened into a constant

`shoop.xtheme.parsing.parse_constantlike(environment, parser)`

Parse the next expression as a “constantlike” expression.

Expression trees that fold into constants are constantlike, as are bare variable names.

Parameters

- **environment** (`jinja2.environment.Environment`) – Jinja2 environment
- **parser** (`jinja2.parser.Parser`) – Template parser

Returns constant value of any type

Return type `object`

`shoop.xtheme.parsing.noop_node(lineno)`

Return a no-op node (compiled into a single 0).

Parameters **lineno** (`int`) – Line number for the node

Returns Node

Return type `jinja2.nodes.ExprStmt`

`class shoop.xtheme.parsing.PlaceholderExtension(environment)`

Bases: `shoop.xtheme.parsing._PlaceholderManagingExtension`

PlaceholderExtension manages `{% placeholder <NAME> [global] %} ... {% endplaceholder %}`.

- The name can be any Jinja2 expression that can be folded into a constant, with the addition of bare variable names such as `name` meaning the same as `"name"`. This makes it slightly easier to write templates.
- The body of this block is actually discarded; only the inner `column`, `row` and `plugin` directives have any meaning. (A parser-time `Layout` object is created and populated during parsing of this block.)
- An optional `global` parameter can be used to specify that the configuration for this placeholder should be “global” across different views (although currently that only applies to placeholders within the same template, i.e, if you defined the same global placeholder in different templates they will not share configuration, but an included or base template that is rendered by different views will).

tags = {'placeholder'}

parse (`parser`)

Parse a placeholder!

Parameters **parser** (`jinja2.parser.Parser`) – Template parser

Returns Output node for rendering a placeholder.

Return type `jinja2.nodes.Output`

identifier = 'shoop.xtheme.parsing.PlaceholderExtension'

class shoop.xtheme.parsing.**LayoutPartExtension** (*environment*)
Bases: shoop.xtheme.parsing._PlaceholderManagingExtension

Parser for row and column tags.

Syntax for the row and column tags is:

```
{% row %}  
    {% column [SIZES] %}...{% endcolumn %}  
{% endrow %}
```

- Rows map to `LayoutRow` objects and columns map to `LayoutCell`.

- For a single-cell layout, these are not necessary. `{% plugin %}` invocations without preceding `{% row %}/{% column %}` directives imply a single row and a single column.

tags = {'column', 'row'}

parse (*parser*)
Parse a column or row.

Parameters **parser** (*jinja2.parser.Parser*) – Template parser

Returns A null output node.

Return type `jinja2.nodes.Node`

identifier = 'shoop.xtheme.parsing.LayoutPartExtension'

class shoop.xtheme.parsing.**PluginExtension** (*environment*)
Bases: shoop.xtheme.parsing._PlaceholderManagingExtension

Parser for plugin tags.

Syntax for plugin tag is:

```
{% plugin <NAME> %}...{% endplugin %}
```

- The (optional) body of the plugin block is expected to be a Jinja2 AST that can be folded into a constant. Generally this means a single block of text (`{% raw %}/{% endraw %}` is okay!).

- The contents of the body, if set, must be valid [TOML markup](#). The TOML is parsed during Jinja2 parse time into a dict, which in turn is folded into the layout description object. This means only the initial parsing of the template incurs whatever performance hit there is in parsing TOML; the Jinja2 bccache should take care of the rest.

tags = {'plugin'}

parse (*parser*)
Parse a column or row.

Parameters **parser** (*jinja2.parser.Parser*) – Template parser

Returns A null output node.

Return type `jinja2.nodes.Node`

identifier = 'shoop.xtheme.parsing.PluginExtension'

shoop.xtheme.rendering module

`shoop.xtheme.rendering.get_view_config(context, global_type=False)`

Get a view configuration object for a Jinja2 rendering context.

Parameters

- **context** (*jinja2.runtime.Context*) – Rendering context
- **global_type** (*bool* / *False*) – Boolean indicating whether this is a global type

Returns View config

Return type *shoop.xtheme.view_config.ViewConfig*

`shoop.xtheme.rendering.render_placeholder(context, placeholder_name, default_layout=None, template_name=None, global_type=False)`

Render a placeholder in a given context.

See *PlaceholderRenderer* for argument docs.

Returns Markup

Return type Markup

`class shoop.xtheme.rendering.PlaceholderRenderer(context, placeholder_name, default_layout=None, template_name=None, global_type=False)`

Bases: *object*

Main class for materializing a placeholder’s contents during template render time.

Parameters

- **context** (*jinja2.runtime.Context*) – Rendering context
- **placeholder_name** (*str*) – Placeholder name
- **default_layout** (*Layout* / *dict* / *None*) – Layout or serialized layout (from template configuration)
- **template_name** (*str* / *None*) – The actual template this node was in. Used to figure out whether the placeholder lives in an extends parent, or in a child.
- **global_type** (*bool* / *False*) – Boolean indicating whether this is a global placeholder

render()

Get this placeholder’s rendered contents.

Returns Rendered markup.

Return type *markupsafe.Markup*

shoop.xtheme.resources module

`class shoop.xtheme.resources.InlineScriptResource`

Bases: *str*

An inline script resource (a subclass of string).

The contents are rendered inside a `<script>` tag.

classmethod from_vars (*var_name, *args, **kwargs*)

Create an *InlineScriptResource* assigning an object of variables into a name in the window scope.

Aside from *var_name* the signature of this function is similar to that of *dict*. Useful for configuration options, etc.

Parameters `var_name` (*str*) – The variable to add into global scope

Returns An *InlineScriptResource* object

Return type *InlineScriptResource*

class `shoop.xtheme.resources.InlineMarkupResource`

Bases: *str*

An inline markup resource (a subclass of string).

The contents are rendered as-is.

class `shoop.xtheme.resources.ResourceContainer`

Bases: *object*

ResourceContainers deal with storing and rendering injected resources.

A ResourceContainer is injected into rendering contexts by *XthemeTemplate* (akin to how django-jinja's Template injects request and csrf_token).

add_resource (*location, resource*)

Add a resource into the given location.

Duplicate resources are ignored (and false is returned). Resource injection order is retained.

Parameters

- **location** (*str*) – The name of the location. See KNOWN_LOCATIONS.
- **resource** (*str/InlineMarkupResource/InlineScriptResource*) – The actual resource. Either an URL string or one of the inline resource classes.

Returns Success flag.

Return type *bool*

render_resources (*location, clean=True*)

Render the resources for the given location, then (by default) clean that list of resources.

Parameters

- **location** (*str*) – The name of the location. See KNOWN_LOCATIONS.
- **clean** (*bool*) – Whether or not to clean up the list of resources.

Returns String of HTML

`shoop.xtheme.resources.inject_resources` (*context, content, clean=True*)

Inject all the resources in the context's ResourceContainer into appropriate places in the content given.

Parameters

- **context** (*jinja2.runtime.Context*) – Rendering context
- **content** (*str*) – HTML content
- **clean** (*bool*) – Clean the resource container as we go?

Returns Possibly modified HTML content

Return type *str*

`shoop.xtheme.resources.get_resource_container` (*context*)

Get a *ResourceContainer* from a rendering context.

Parameters **context** (*jinja2.runtime.Context*) – Context

Returns Resource Container

Return type `shoop.xtheme.resources.ResourceContainer|None`

`shoop.xtheme.resources.add_resource(context, location, resource)`

Add an Xtheme resource into the given context.

Parameters

- **context** (`jinja2.runtime.Context`) – Context
- **location** (`str`) – Location string (see `KNOWN_LOCATIONS`)
- **resource** (`str|InlineMarkupResource|InlineScriptResource`) – Resource descriptor (URL or inline markup object)

Returns Success flag

Return type `bool`

shoop.xtheme.template_ns module

class `shoop.xtheme.template_ns.XthemeNamespace`

Bases: `object`

A template helper namespace for Xtheme-related functionality.

get_view_name (`context`)

Get the current view's view name (used for identifying view configurations).

Parameters **context** (`jinja2.runtime.Context`) – Implicit Jinja2 context

Returns View name string

Return type `str`

is_edit_mode (`context`)

Get the current edit mode status.

Parameters **context** (`jinja2.runtime.Context`) – Implicit Jinja2 context

Returns Edit mode enable flag

Return type `bool`

get (`context, name, default=None`)

Get a theme setting value.

Parameters

- **context** (`jinja2.runtime.Context`) – Implicit Jinja2 context
- **name** (`str`) – Setting name
- **default** (`object`) – Default value if setting is not found

Returns Value

Return type `object`

shoop.xtheme.testing module

`shoop.xtheme.testing.override_current_theme_class(theme_class=<object object>)`

Context manager for overriding the currently active theme class for testing.

An instance of this class is then returned by `get_current_theme`.

A falsy value means `None` is returned from `get_current_theme`, which is also useful for testing.

Parameters **theme_class** (`class[Theme]`) – A theme class object

shoop.xtheme.urls module**shoop.xtheme.utils module**

`shoop.xtheme.utils.join_css_classes` (*class_list*)

Join an iterable of truthy values by spaces, effectively creating a list of CSS classes.

The retval is sorted for cleanliness.

Parameters `class_list` (*Iterable[str]*) – Iterable of classes

Returns String

Return type `str`

`shoop.xtheme.utils.get_html_attrs` (*attrs*)

Flatten a dict into HTML attributes (it's `django.forms.utils.flatatt` on steroids!).

Only truthy keys and values are taken into account; list-like values are flattened with `join_css_classes`

Parameters `attrs` (*dict[str, object]*) – Attribute dict

Returns string ready to paste after a HTML tag open. `<foo%s>!`

Return type `str`

shoop.xtheme.view_config module

`class` `shoop.xtheme.view_config.ViewConfig` (*theme, view_name, draft, global_type=False*)

Bases: `object`

A view configuration.

Contains layout and plugin configuration for all placeholders in a given view.

This class does not directly correspond to a database model; it may act as a container for `SavedViewConfig` objects, and wraps the `SavedViewConfig` API.

Initialize a view configuration.

Parameters

- **theme** (*shoop.xtheme.Theme* | *None*) – Theme object (could be `None` to not touch the database)
- **view_name** (*str*) – View name (the class name of the view)
- **draft** (*bool*) – Load in draft mode?
- **global_type** (*bool* | *False*) – Boolean indicating whether this is a global config

saved_view_config

Get a saved view config model depending on the current parameters.

Returns A `SavedViewConfig` object for the current theme/view/draft mode, or `None`

Return type `shoop.xtheme.models.SavedViewConfig` | `None`

`get_placeholder_layout` (*placeholder_name, default_layout=None*)

Get a `Layout` object for the given placeholder.

Parameters

- **placeholder_name** (*str*) – The name of the placeholder to load.
- **default_layout** (*dict* | *Layout*) – Default layout configuration (either a dict or an actual `Layout`)

Returns Layout

Return type *Layout*

save_default_placeholder_layout (*placeholder_name*, *layout*)

Save a default placeholder layout (only if no data for the PH already exists).

Parameters

- **placeholder_name** (*str*) – Placeholder name
- **layout** (*Layout* / *dict*) – Layout or layout data

Returns True if saved

Return type *bool*

publish ()

Publish this revision of the view configuration as the currently public one.

Returns Success flag

Return type *bool*

revert ()

Revert this revision of the view configuration, if it's a draft.

Returns Success flag

Return type *bool*

save_placeholder_layout (*placeholder_name*, *layout*)

Save the given layout as the layout for the given placeholder.

Parameters

- **placeholder_name** (*str*) – The placeholder name.
- **layout** (*Layout* / *dict*) – Layout object (or dict)

Module contents

class `shoop.xtheme.Plugin` (*config*)

Bases: *object*

A plugin that can be instantiated within a `shoop.xtheme.layout.LayoutCell`.

Other plugins should inherit from this class and register themselves in the `xtheme_plugin` provide category.

Instantiate a Plugin with the given `config` dictionary.

Parameters **config** (*dict*) – Dictionary of freeform configuration data

editor_form_class

alias of `GenericPluginForm`

fields = []

get_editor_form_class ()

Return the form class for editing this plugin.

The form class should either derive from `PluginForm`, or at least have a `get_config()` method.

Form classes without *fields* are treated the same way as if you'd return `None`, i.e. no configuration form is presented to the user.

Returns Editor form class

Return type `class[forms.Form]|None`

classmethod `get_plugin_choices` (*empty_label=None*)

Get a sorted list of 2-tuples (identifier and name) of available Xtheme plugins.

Handy for `<select>` boxes.

Parameters `empty_label` (*str|None*) – Label for the “empty” choice. If falsy, no empty choice is prepended

Returns List of 2-tuples

Return type `Iterable[tuple[str, str]]`

get_translated_value (*key, default=None, language=None*)

Get a translated value from the plugin’s configuration.

It’s assumed that translated values are stored in a `{language: data, ...}` dictionary in the plugin configuration blob. This is the protocol that `shoop.xtheme.plugins.forms.TranslatableField` uses.

If the configuration blob contains such a dictionary, but it does not contain a translated value in the requested language does not exist, the fallback value, if any, within that dictionary is tried next. Failing that, the `default` value is returned.

Parameters

- **key** (*str*) – Configuration key
- **default** – Default value to return when all else fails.
- **language** (*str|None*) – Requested language. Defaults to the active language.

Returns A translated value.

identifier = None

is_context_valid (*context*)

Check that the given rendering context is valid for rendering this plugin.

By default, just checks `required_context_variables`.

Parameters `context` (*jinja2.runtime.Context*) – Rendering context

Returns True if we should bother trying to render this

Return type `bool`

classmethod `load` (*identifier*)

Get a plugin class based on the identifier from the `xtheme_plugin` provides registry.

Parameters `identifier` (*str*) – Plugin class identifier

Returns A plugin class, or None

Return type `class[Plugin]|None`

name = <django.utils.functional.lazy.<locals>.__proxy__ object>

render (*context*)

Return the HTML for a plugin in a given rendering context.

Parameters `context` (*jinja2.runtime.Context*) – Rendering context

Returns String of rendered content.

Return type `str`

```

    required_context_variables = set()
class shoop.xtheme.TemplatedPlugin (config)
    Bases: shoop.xtheme.Plugin

    Convenience base class for plugins that just render a “sub-template” with a given context.

    Instantiate a Plugin with the given config dictionary.

    Parameters config (dict) – Dictionary of freeform configuration data

    config_copied_variables = set()

    engine = None

    get_context_data (context)
        Get a context dictionary from a Jinja2 context.

        Parameters context (jinja2.runtime.Context) – Jinja2 rendering context

        Returns Dict of vars

        Return type dict[str, object]

    inherited_variables = set()

    render (context)

    template_name = ''
class shoop.xtheme.Theme (settings_obj=None)
    Bases: object

    Base class for all Xtheme themes.

    This class does not directly correspond to a database object; it’s used in the rendering, etc. process.

    It does, however, act as a container for a ThemeSettings object that contains the actual persisted settings
    etc.

    Initialize this theme, with an optional ThemeSettings object

    Parameters settings_obj (ThemeSettings/None) – A theme settings object for this
        theme, if one exists. Passing this in will avoid extraneous database queries.

    author = ''

    fields = []

    get_configuration_form (form_kwargs)
        Return a ModelForm instance (model=ThemeSettings) for configuring this theme.

        By default, returns a GenericThemeForm (a ModelForm populated from theme.fields).

        Parameters form_kwargs (dict) – The keyword arguments that should be used for initial-
            izing the form

        Return type django.forms.ModelForm

    get_setting (key, default=None)
        Get a setting value for this theme.

        Parameters

        • key (str) – Setting name

        • default (object) – Default value, if the setting is not set

        Returns Setting value

```

Return type *object*

get_settings ()

Get all the currently set settings for the theme as a dict.

Returns Dict of settings

Return type *dict*

get_view (*view_name*)

Get an extra view for this theme.

Views may be either normal Django functions or CBVs (or anything that has `as_view()` really). Falsy values are considered “not found”.

Parameters **view_name** (*str*) – View name

Returns The extra view, if one exists for the given name.

Return type *dict[str, View|callable]|None*

global_placeholders = []

identifier = None

name = ‘’

set_current ()

Set this theme as the active theme.

set_setting (*key*, *value*)

Set a theme setting *key* to the value *value*.

Parameters

- **key** (*str*) – Setting name
- **value** (*object*) – Setting value

set_settings (**args*, ***kwargs*)

Set a number of settings for this theme.

The arguments are exactly the same as those to *dict*.

Note: It’s better to call this once than *set_setting* several times.

settings_obj

Get a saved settings model for this theme. If one does not yet exist, an unsaved one is returned.

If one was passed in the ctor, naturally that one is returned.

Return type *shoop.xtheme.models.ThemeSettings*

template_dir = None

shoop.xtheme.get_current_theme (*request=None*)

Get the currently active theme object.

Parameters **request** (*HttpRequest|None*) – Request, if available

Returns Theme object or None

Return type *Theme*

shoop.xtheme.get_theme_by_identifier (*identifier*, *settings_obj=None*)

Get an instantiated theme by identifier.

Parameters

- **identifier** (*str*) – Theme identifier
- **settings_obj** (`shoop.xtheme.models.ThemeSettings`) – Optional Theme-Settings object for the theme constructor

Returns Theme object or None

Return type *Theme*

`shoop.xtheme.set_current_theme(identifier)`

Activate a theme based on identifier.

Parameters **identifier** (*str*) – Theme identifier

`shoop.xtheme.templated_plugin_factory(identifier, template_name, **kwargs)`

A factory (akin to `modelform_factory`) to quickly create simple plugins.

Parameters

- **identifier** (*str*) – The unique identifier for the new plugin.
- **template_name** (*str*) – The template file path this plugin should render
- **kwargs** (*dict*) – Other arguments for the *TemplatedPlugin/Plugin* classes.

Returns New *TemplatedPlugin* subclass

Return type `class[TemplatedPlugin]`

Submodules**shoop.configuration module**

API for Shoop's Dynamic Configuration.

Idea of the Dynamic Configuration is to allow storing configuration values similarly as *Django settings* allows, but in a more flexible way: Dynamic Configuration can be changed with a simple API and there is no need restart the application server after changing a value.

Dynamic configuration values are permanent. Current implementation stores the values with *ConfigurationItem* model into database, but that may change in the future.

Configuration values are get and set by a key string. There is a global configuration and a shop specific configuration for each shop. Values in shop specific configuration override the values in global configuration.

`shoop.configuration.set(shop, key, value)`

Set configuration item value for a shop or globally.

If given `shop` is `None`, the value of given `key` is set globally for all shops. Otherwise sets a shop specific value which overrides the global value in configuration of the specified shop.

Parameters

- **shop** (`shoop.core.models.Shop|None`) – Shop to set value for, or `None` to set a global value
- **key** (*str*) – Name of the key to set
- **value** (*Any*) – Value to set. Note: Must be JSON serializable.

`shoop.configuration.get(shop, key, default=None)`

Get configuration value by shop and key.

Global configuration can be accessed with `shop=None`.

Parameters

- **shop** (`shoop.core.models.Shop|None`) – Shop to get configuration value for, or `None`
- **key** (`str`) – Configuration item key
- **default** (`Any`) – Default value returned if no value is set for given key (globally or in given shop).

Returns Configuration value or the default value

Return type Any

Module contents

Shoop Web APIs

REST API

The Shoop REST API is built on [Django REST Framework](#) with additional functionality built on *The Provides system* to auto-discover available viewsets.

Setting up the Shoop REST API

First, add `rest_framework` and `shoop.api` to your `INSTALLED_APPS`.

Then – and this differs from Django REST Framework’s defaults – you *must* add the `REST_FRAMEWORK` configuration dict to your settings. Django REST Framework defaults to no permission checking whatsoever (`rest_framework.permissions.AllowAny`), which would make all of your data world-readable and writable.

This is not what we want to accidentally happen, so configuration is enforced.

For the sake of demonstration, let’s make the API only accessible for superusers with the `IsAdminUser` permission policy. (Authentication is enabled by the default settings.)

```
REST_FRAMEWORK = {
    'DEFAULT_PERMISSION_CLASSES': ('rest_framework.permissions.IsAdminUser',)
}
```

Now just add the API to your root `urlpatterns`.

```
urlpatterns = patterns(
    # *snip*
    url(r'^api/', include('shoop.api.urls')),
    # *snip*
)
```

All done! If you visit the `/api/` URL (as a suitably authenticated user), you should be presented with Django REST Framework’s human-friendly user interface.

REST API Usage Example

This sketch of a script illustrates how to use the Shoop REST API.

```
# -*- coding: utf-8 -*-
import json
import uuid
import requests

base_url = "http://127.0.0.1:8000/api/"
s = requests.session()
s.auth = ("admin", "admin")

def send(endpoint, data, method="post"):
    data = json.dumps(data)
    resp = s.request(method, base_url + endpoint, data=data, headers={
        "Content-Type": "application/json",
        "Accept": "application/json;indent=4",
        "X-Requested-With": "XMLHttpRequest" # For `request.is_ajax()`
    })
    if resp.status_code > 300:
        raise Exception(resp.text)
    return resp.json()

def create_product():
    product = send("shoop/product/", {
        "tax_class": 1,
        "sku": str(uuid.uuid4()),
        "type": 1,
        "translations": {
            "en": {
                "name": "Hello"
            }
        }
    })
    return product

def create_shop_product(product):
    product_id = product["id"]

    shop_product = send("shoop/shop_product/", {
        "product": product_id,
        "shop": 1,
    })
    assert not shop_product.get("primary_category")

    shop_product = send("shoop/shop_product/%d/" % shop_product["id"], {
        "primary_category": 1,
        "purchase_multiple": 38
    }, "patch")
    assert shop_product.get("primary_category") == 1
    return shop_product

def create_product_price(product):
    price = send("shoop/cgp_price/", {
        "product": product["id"],
```

```
        "shop": None,
        "group": None,
        "price": 180
    })
    return price

def main():
    product = create_product()
    shop_product = create_shop_product(product)
    price = create_product_price(product)
    print(product["id"])

if __name__ == "__main__":
    main()
```

Indices and tables

- `genindex`
- `modindex`
- `search`

S

- shoop, 378
- shoop.addons, 42
- shoop.addons.admin_module, 40
- shoop.addons.admin_module.views, 39
- shoop.addons.admin_module.views.list, 38
- shoop.addons.admin_module.views.reload, 38
- shoop.addons.admin_module.views.upload, 39
- shoop.addons.installer, 41
- shoop.addons.manager, 41
- shoop.addons.reloader, 41
- shoop.admin, 130
- shoop.admin.base, 124
- shoop.admin.breadcrumbs, 126
- shoop.admin.currencybound, 126
- shoop.admin.dashboard, 44
- shoop.admin.dashboard.blocks, 42
- shoop.admin.dashboard.charts, 43
- shoop.admin.dashboard.utils, 43
- shoop.admin.form_part, 127
- shoop.admin.forms, 46
- shoop.admin.forms.fields, 45
- shoop.admin.forms.widgets, 45
- shoop.admin.menu, 127
- shoop.admin.module_registry, 127
- shoop.admin.modules, 116
- shoop.admin.modules.attributes, 48
- shoop.admin.modules.attributes.views, 47
- shoop.admin.modules.attributes.views.edit, 47
- shoop.admin.modules.attributes.views.list, 47
- shoop.admin.modules.categories, 50
- shoop.admin.modules.categories.views, 49
- shoop.admin.modules.categories.views.edit, 48
- shoop.admin.modules.categories.views.list, 49
- shoop.admin.modules.contact_groups, 53
- shoop.admin.modules.contact_groups.views, 52
- shoop.admin.modules.contact_groups.views.delete, 50
- shoop.admin.modules.contact_groups.views.edit, 51
- shoop.admin.modules.contact_groups.views.forms, 51
- shoop.admin.modules.contact_groups.views.list, 52
- shoop.admin.modules.contacts, 57
- shoop.admin.modules.contacts.dashboard, 57
- shoop.admin.modules.contacts.views, 56
- shoop.admin.modules.contacts.views.detail, 54
- shoop.admin.modules.contacts.views.edit, 54
- shoop.admin.modules.contacts.views.list, 55
- shoop.admin.modules.contacts.views.reset, 56
- shoop.admin.modules.demo, 58
- shoop.admin.modules.manufacturers, 60
- shoop.admin.modules.manufacturers.views, 59
- shoop.admin.modules.manufacturers.views.edit, 58
- shoop.admin.modules.manufacturers.views.list, 59
- shoop.admin.modules.media, 61
- shoop.admin.modules.media.utils, 60
- shoop.admin.modules.media.views, 60
- shoop.admin.modules.orders, 71
- shoop.admin.modules.orders.dashboard, 69
- shoop.admin.modules.orders.json_order_creator,

70	shoop.admin.modules.products.views.edit_media,
shoop.admin.modules.orders.receivers,	85
70	shoop.admin.modules.products.views.edit_package,
shoop.admin.modules.orders.sections, 71	86
shoop.admin.modules.orders.views, 66	shoop.admin.modules.products.views.edit_parent,
shoop.admin.modules.orders.views.detail,	87
61	shoop.admin.modules.products.views.edit_variation,
shoop.admin.modules.orders.views.edit,	88
62	shoop.admin.modules.products.views.list,
shoop.admin.modules.orders.views.list,	88
63	shoop.admin.modules.products.views.toolbars,
shoop.admin.modules.orders.views.log,	88
63	shoop.admin.modules.sales_units, 92
shoop.admin.modules.orders.views.payment,	shoop.admin.modules.sales_units.views,
63	92
shoop.admin.modules.orders.views.refund,	shoop.admin.modules.sales_units.views.edit,
63	91
shoop.admin.modules.orders.views.shipments,	shoop.admin.modules.sales_units.views.list,
65	92
shoop.admin.modules.permission_groups,	shoop.admin.modules.service_providers,
73	94
shoop.admin.modules.permission_groups.views,	shoop.admin.modules.service_providers.forms,
73	94
shoop.admin.modules.permission_groups.views.edit,	shoop.admin.modules.service_providers.views,
72	93
shoop.admin.modules.permission_groups.views.list,	shoop.admin.modules.services, 101
73	shoop.admin.modules.services.base_form_part,
shoop.admin.modules.product_types, 75	96
shoop.admin.modules.product_types.views,	shoop.admin.modules.services.behavior_form_part,
74	96
shoop.admin.modules.product_types.views.edit,	shoop.admin.modules.services.forms, 97
74	shoop.admin.modules.services.views, 95
shoop.admin.modules.product_types.views.list,	shoop.admin.modules.services.weight_based_pricing,
74	95
shoop.admin.modules.products, 91	shoop.admin.modules.shops, 104
shoop.admin.modules.products.forms, 79	shoop.admin.modules.shops.views, 104
shoop.admin.modules.products.forms.base_form,	shoop.admin.modules.shops.views.edit,
75	102
shoop.admin.modules.products.forms.package_form,	shoop.admin.modules.shops.views.list,
78	103
shoop.admin.modules.products.forms.parent_form,	shoop.admin.modules.suppliers, 106
78	shoop.admin.modules.suppliers.views, 106
shoop.admin.modules.products.forms.simple_variation_form,	shoop.admin.modules.suppliers.views.edit,
78	105
shoop.admin.modules.products.forms.variable_form,	shoop.admin.modules.suppliers.views.list,
79	105
shoop.admin.modules.products.utils, 91	shoop.admin.modules.system, 107
shoop.admin.modules.products.views, 89	shoop.admin.modules.system.views, 107
shoop.admin.modules.products.views.delete,	shoop.admin.modules.system.views.telemetry,
83	107
shoop.admin.modules.products.views.edit,	shoop.admin.modules.taxes, 111
83	shoop.admin.modules.taxes.views, 109
shoop.admin.modules.products.views.edit_shop,	shoop.admin.modules.taxes.views.edit,
84	107

shoop.admin.modules.taxes.views.list, 109
 shoop.admin.modules.users, 116
 shoop.admin.modules.users.views, 114
 shoop.admin.modules.users.views.detail, 111
 shoop.admin.modules.users.views.list, 112
 shoop.admin.modules.users.views.password, 112
 shoop.admin.modules.users.views.permissions, 113
 shoop.admin.template_helpers, 116
 shoop.admin.template_helpers.shoop_admin, 116
 shoop.admin.templatetags, 117
 shoop.admin.templatetags.shoop_admin, 117
 shoop.admin.toolbar, 128
 shoop.admin.urls, 130
 shoop.admin.utils, 123
 shoop.admin.utils.bs3_renderers, 117
 shoop.admin.utils.forms, 117
 shoop.admin.utils.permissions, 118
 shoop.admin.utils.picotable, 118
 shoop.admin.utils.search, 120
 shoop.admin.utils.urls, 121
 shoop.admin.utils.views, 122
 shoop.admin.views, 124
 shoop.admin.views.dashboard, 123
 shoop.admin.views.menu, 123
 shoop.admin.views.search, 123
 shoop.admin.views.select, 124
 shoop.api, 132
 shoop.api.encoders, 131
 shoop.api.factories, 131
 shoop.api.urls, 131
 shoop.apps, 134
 shoop.apps.provides, 132
 shoop.apps.settings, 133
 shoop.campaigns, 178
 shoop.campaigns.admin_module, 145
 shoop.campaigns.admin_module.form_parts, 142
 shoop.campaigns.admin_module.form_sets, 143
 shoop.campaigns.admin_module.forms, 135
 shoop.campaigns.admin_module.views, 140
 shoop.campaigns.apps, 177
 shoop.campaigns.models, 169
 shoop.campaigns.models.basket_conditions, 145
 shoop.campaigns.models.basket_effects, 152
 shoop.campaigns.models.basket_line_effects, 154
 shoop.campaigns.models.campaigns, 157
 shoop.campaigns.models.catalog_filters, 161
 shoop.campaigns.models.contact_group_sales_ranges, 164
 shoop.campaigns.models.context_conditions, 164
 shoop.campaigns.models.product_effects, 167
 shoop.campaigns.modules, 178
 shoop.campaigns.signal_handlers, 178
 shoop.campaigns.templates, 177
 shoop.campaigns.utils, 178
 shoop.configuration, 377
 shoop.core, 264
 shoop.core.api, 180
 shoop.core.api.orders, 178
 shoop.core.api.products, 179
 shoop.core.cache, 182
 shoop.core.cache.impl, 180
 shoop.core.defaults, 183
 shoop.core.defaults.order_statuses, 183
 shoop.core.excs, 260
 shoop.core.fields, 184
 shoop.core.fields.tagged_json, 183
 shoop.core.management, 187
 shoop.core.management.commands, 187
 shoop.core.management.commands.compute_bought_with, 185
 shoop.core.management.commands.makemessages, 185
 shoop.core.management.commands.shoop_init, 186
 shoop.core.management.commands.shoop_makemessages, 186
 shoop.core.management.commands.shoop_show_settings, 186
 shoop.core.middleware, 260
 shoop.core.models, 187
 shoop.core.modules, 238
 shoop.core.modules.interface, 237
 shoop.core.order_creator, 238
 shoop.core.order_creator.signals, 238
 shoop.core.pricing, 243
 shoop.core.pricing.default_pricing, 242
 shoop.core.settings, 261
 shoop.core.shortcuts, 250
 shoop.core.signals, 263
 shoop.core.stocks, 250
 shoop.core.suppliers, 251
 shoop.core.suppliers.base, 251
 shoop.core.taxing, 253

shoop.core.taxing.utils, 252
shoop.core.telemetry, 263
shoop.core.templatetags, 256
shoop.core.templatetags.prices, 255
shoop.core.templatetags.shoop_common, 256
shoop.core.utils, 260
shoop.core.utils.form_mixins, 257
shoop.core.utils.maintenance, 257
shoop.core.utils.model_caching_descriptor, 257
shoop.core.utils.name_mixin, 257
shoop.core.utils.price_display, 258
shoop.core.utils.prices, 258
shoop.core.utils.product_bought_with_relations, 258
shoop.core.utils.product_caching_object, 259
shoop.core.utils.query, 259
shoop.core.utils.slugs, 259
shoop.core.utils.tax_numbers, 259
shoop.core.utils.users, 259
shoop.core.utils.vat, 259
shoop.customer_group_pricing, 265
shoop.customer_group_pricing.admin_form_handler, 264
shoop.customer_group_pricing.api, 264
shoop.customer_group_pricing.models, 265
shoop.customer_group_pricing.module, 265
shoop.default_tax, 268
shoop.default_tax.admin_module, 266
shoop.default_tax.admin_module.views, 266
shoop.default_tax.models, 267
shoop.default_tax.module, 267
shoop.front, 296
shoop.front.admin_module, 269
shoop.front.admin_module.carts, 269
shoop.front.admin_module.carts.views, 268
shoop.front.apps, 277
shoop.front.apps.auth, 271
shoop.front.apps.auth.forms, 269
shoop.front.apps.auth.urls, 270
shoop.front.apps.auth.views, 270
shoop.front.apps.customer_information, 273
shoop.front.apps.customer_information.forms, 271
shoop.front.apps.customer_information.urls, 272
shoop.front.apps.customer_information.views, 272
shoop.front.apps.personal_order_history, 273
shoop.front.apps.personal_order_history.urls, 273
shoop.front.apps.personal_order_history.views, 273
shoop.front.apps.registration, 274
shoop.front.apps.registration.settings, 274
shoop.front.apps.registration.urls, 274
shoop.front.apps.registration.views, 274
shoop.front.apps.simple_order_notification, 275
shoop.front.apps.simple_order_notification.admin_module, 275
shoop.front.apps.simple_order_notification.template_helpers, 275
shoop.front.apps.simple_search, 277
shoop.front.apps.simple_search.template_helpers, 276
shoop.front.apps.simple_search.urls, 276
shoop.front.apps.simple_search.views, 276
shoop.front.basket, 281
shoop.front.basket.command_dispatcher, 277
shoop.front.basket.commands, 278
shoop.front.basket.objects, 279
shoop.front.basket.order_creator, 280
shoop.front.basket.storage, 280
shoop.front.basket.update_methods, 281
shoop.front.checkout, 285
shoop.front.checkout.addresses, 282
shoop.front.checkout.confirm, 283
shoop.front.checkout.empty, 283
shoop.front.checkout.methods, 284
shoop.front.checkout.single_page, 285
shoop.front.error_handling, 293
shoop.front.middleware, 293
shoop.front.models, 287
shoop.front.models.stored_basket, 287
shoop.front.notify_events, 294
shoop.front.settings, 295
shoop.front.signals, 295
shoop.front.template_helpers, 290
shoop.front.template_helpers.category, 288
shoop.front.template_helpers.general, 288
shoop.front.template_helpers.product, 289
shoop.front.template_helpers.urls, 289
shoop.front.templatetags, 290

- shoop.front.templatetags.shoop_front, 290
- shoop.front.templatetags.thumbnails, 290
- shoop.front.urls, 296
- shoop.front.utils, 291
- shoop.front.utils.product_sorting, 290
- shoop.front.utils.product_statistics, 290
- shoop.front.utils.views, 291
- shoop.front.views, 293
- shoop.front.views.basket, 291
- shoop.front.views.category, 291
- shoop.front.views.checkout, 292
- shoop.front.views.index, 292
- shoop.front.views.order, 292
- shoop.front.views.payment, 293
- shoop.front.views.product, 293
- shoop.guide, 297
- shoop.guide.admin_module, 296
- shoop.guide.apps, 296
- shoop.guide.settings, 296
- shoop.notify, 313
- shoop.notify.actions, 298
- shoop.notify.actions.debug, 297
- shoop.notify.actions.email, 297
- shoop.notify.actions.notification, 297
- shoop.notify.actions.order, 298
- shoop.notify.admin_module, 301
- shoop.notify.admin_module.forms, 301
- shoop.notify.admin_module.utils, 301
- shoop.notify.admin_module.views, 300
- shoop.notify.admin_module.views.edit, 299
- shoop.notify.admin_module.views.editor, 299
- shoop.notify.admin_module.views.list, 300
- shoop.notify.base, 306
- shoop.notify.conditions, 303
- shoop.notify.conditions.simple, 302
- shoop.notify.enums, 309
- shoop.notify.models, 305
- shoop.notify.models.notification, 304
- shoop.notify.models.script, 305
- shoop.notify.runner, 310
- shoop.notify.script, 310
- shoop.notify.settings, 311
- shoop.notify.template, 311
- shoop.notify.typology, 311
- shoop.order_printouts, 316
- shoop.order_printouts.admin_module, 316
- shoop.order_printouts.admin_module.toolbars, 316
- shoop.order_printouts.admin_module.views, 316
- shoop.order_printouts.pdf_export, 316
- shoop.simple_cms, 321
- shoop.simple_cms.admin_module, 318
- shoop.simple_cms.admin_module.views, 317
- shoop.simple_cms.models, 318
- shoop.simple_cms.plugins, 319
- shoop.simple_cms.template_helpers, 320
- shoop.simple_cms.urls, 320
- shoop.simple_cms.views, 320
- shoop.simple_supplier, 325
- shoop.simple_supplier.admin_module, 322
- shoop.simple_supplier.admin_module.forms, 321
- shoop.simple_supplier.admin_module.views, 322
- shoop.simple_supplier.forms, 322
- shoop.simple_supplier.models, 323
- shoop.simple_supplier.module, 324
- shoop.simple_supplier.utils, 324
- shoop.testing, 336
- shoop.testing.admin_module, 326
- shoop.testing.admin_module.mock_view, 325
- shoop.testing.factories, 329
- shoop.testing.image_generator, 332
- shoop.testing.management, 326
- shoop.testing.management.commands, 326
- shoop.testing.management.commands.shoop_populate_mocks, 326
- shoop.testing.mock_population, 333
- shoop.testing.models, 327
- shoop.testing.service_forms, 333
- shoop.testing.simple_checkout_phase, 334
- shoop.testing.soup_utils, 336
- shoop.testing.text_data, 336
- shoop.testing.utils, 336
- shoop.themes, 338
- shoop.themes.classic_gray, 337
- shoop.themes.classic_gray.plugins, 337
- shoop.themes.classic_gray.views, 336
- shoop.utils, 350
- shoop.utils.analog, 338
- shoop.utils.dates, 338
- shoop.utils.decorators, 339
- shoop.utils.excs, 339
- shoop.utils.fields, 340
- shoop.utils.filer, 340
- shoop.utils.form_group, 341
- shoop.utils.forms, 341
- shoop.utils.http, 342
- shoop.utils.i18n, 342
- shoop.utils.importing, 343

- shoop.utils.iterables, [343](#)
- shoop.utils.models, [343](#)
- shoop.utils.money, [344](#)
- shoop.utils.multilanguage_model_form,
[344](#)
- shoop.utils.numbers, [344](#)
- shoop.utils.objects, [345](#)
- shoop.utils.patterns, [346](#)
- shoop.utils.properties, [347](#)
- shoop.utils.serialization, [348](#)
- shoop.utils.settings_doc, [349](#)
- shoop.utils.setup, [349](#)
- shoop.utils.text, [349](#)
- shoop.utils.translation, [350](#)
- shoop.xtheme, [373](#)
- shoop.xtheme.admin_module, [352](#)
- shoop.xtheme.admin_module.views, [351](#)
- shoop.xtheme.editing, [359](#)
- shoop.xtheme.engine, [360](#)
- shoop.xtheme.forms, [361](#)
- shoop.xtheme.layout, [361](#)
- shoop.xtheme.models, [365](#)
- shoop.xtheme.parsing, [366](#)
- shoop.xtheme.plugins, [357](#)
- shoop.xtheme.plugins.category_links, [353](#)
- shoop.xtheme.plugins.consts, [353](#)
- shoop.xtheme.plugins.forms, [353](#)
- shoop.xtheme.plugins.image, [354](#)
- shoop.xtheme.plugins.snippets, [355](#)
- shoop.xtheme.plugins.social_media_links,
[355](#)
- shoop.xtheme.plugins.text, [356](#)
- shoop.xtheme.plugins.widgets, [356](#)
- shoop.xtheme.rendering, [369](#)
- shoop.xtheme.resources, [369](#)
- shoop.xtheme.template_ns, [371](#)
- shoop.xtheme.templatetags, [357](#)
- shoop.xtheme.templatetags.xtheme_tags,
[357](#)
- shoop.xtheme.testing, [371](#)
- shoop.xtheme.urls, [372](#)
- shoop.xtheme.utils, [372](#)
- shoop.xtheme.view_config, [372](#)
- shoop.xtheme.views, [359](#)
- shoop.xtheme.views.command, [357](#)
- shoop.xtheme.views.editor, [358](#)
- shoop.xtheme.views.extra, [358](#)
- shoop.xtheme.views.forms, [358](#)

A

- absolute_max (shoop.admin.modules.products.forms.base_forms.BaseProductMediaFormSet attribute), 76
- absolute_max (shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSellFormSet attribute), 85
- absolute_max (shoop.admin.modules.products.views.edit_media.ProductMediaFormSet attribute), 86
- absolute_max (shoop.admin.modules.services.behavior_form_part.BehaviorFormSet attribute), 96
- absolute_max (shoop.campaigns.admin_module.form_sets.BaseFormSet attribute), 144
- abstract (shoop.campaigns.models.Campaign.Meta attribute), 173
- abstract (shoop.campaigns.models.campaigns.Campaign.Meta attribute), 157
- abstract (shoop.core.models.PolymorphicShoopModel.Meta attribute), 210
- abstract (shoop.core.models.PolymorphicTranslatableShoopModel.Meta attribute), 210
- abstract (shoop.core.models.Service.Meta attribute), 224
- abstract (shoop.core.models.ShoopModel.Meta attribute), 229
- abstract (shoop.core.models.TranslatableShoopModel.Meta attribute), 235
- abstract (shoop.utils.analog.BaseLogEntry.Meta attribute), 338
- accepts_any_type (shoop.notify.base.Binding attribute), 307
- accepts_any_type (shoop.notify.Binding attribute), 314
- account_manager (shoop.core.models.Contact attribute), 192
- Action (class in shoop.notify), 313
- Action (class in shoop.notify.base), 308
- activate() (shoop.xtheme.models.ThemeSettings method), 366
- activation_complete() (in module shoop.front.apps.registration.views), 274
- ActivationForm (class in shoop.xtheme.admin_module.views), 351
- ActivationView (class in shoop.front.apps.registration.views), 274
- active() (shoop.campaigns.models.contact_group_sales_ranges.SalesRange method), 164
- Activity (class in shoop.admin.base), 125
- add_arguments() (shoop.core.management.commands.makemessages.Command method), 185
- add_arguments() (shoop.core.management.commands.shoop_makemessages.Command method), 186
- add_bought_with_relations_for_product() (in module shoop.core.utils.product_bought_with_relations), 258
- add_cell() (shoop.xtheme.layout.LayoutRow method), 362
- add_class_attrs() (shoop.admin.utils.bs3_renderers.AdminFieldRenderer method), 117
- add_code() (shoop.core.order_creator.OrderSource method), 239
- add_code() (shoop.front.basket.objects.BaseBasket method), 279
- add_create_or_change_message() (in module shoop.admin.utils.views), 122
- add_data() (shoop.admin.dashboard.BarChart method), 44
- add_data() (shoop.admin.dashboard.charts.BarChart method), 43
- add_edit_resources() (in module shoop.xtheme.editing), 359
- add_enabled_addons() (in module shoop.addons), 42
- add_enabled_addons() (in module shoop.addons.manager), 41
- add_error() (shoop.admin.modules.orders.json_order_creator.JsonOrderCreator method), 70
- add_form_def() (shoop.utils.form_group.FormGroup method), 341
- add_form_errors_as_messages (shoop.admin.modules.permission_groups.views.edit.PermissionGroup attribute), 72
- add_form_errors_as_messages (shoop.admin.modules.permission_groups.views.PermissionGroup attribute), 73
- add_form_errors_as_messages

- (shoop.admin.modules.service_providers.views.ServiceProviderEditView attribute), 93
- add_form_errors_as_messages (shoop.admin.modules.taxes.views.edit.TaxEditView attribute), 108
- add_form_errors_as_messages (shoop.admin.modules.taxes.views.TaxEditView attribute), 110
- add_form_errors_as_messages (shoop.admin.utils.views.CreateOrUpdateView attribute), 122
- add_form_errors_as_messages (shoop.campaigns.admin_module.views.CouponEditView attribute), 141
- add_form_errors_as_messages (shoop.default_tax.admin_module.views.TaxRuleEditView attribute), 266
- add_form_errors_as_messages (shoop.simple cms.admin_module.views.PageEditView attribute), 317
- add_form_errors_as_messages (shoop.xtheme.admin_module.views.ThemeConfigDetailView attribute), 352
- add_form_errors_as_messages() (in module shoop.admin.utils.forms), 117
- add_help_attrs() (shoop.admin.utils.bs3_renderers.AdminFieldRenderers method), 117
- add_line() (shoop.core.order_creator.OrderSource method), 239
- add_line() (shoop.front.basket.objects.BaseBasket method), 279
- add_log_entry() (shoop.core.models.Category method), 190
- add_log_entry() (shoop.core.models.Order method), 199
- add_log_entry() (shoop.core.models.Product method), 210, 214
- add_log_entry_on_log_target() (shoop.notify.Context method), 314
- add_log_entry_on_log_target() (shoop.notify.script.Context method), 310
- add_phase_attributes() (shoop.front.checkout.CheckoutProcess method), 286
- add_plugin() (shoop.xtheme.layout.Layout method), 364
- add_product() (shoop.front.basket.objects.BaseBasket method), 279
- add_product_to_order() (in module shoop.testing.factories), 331
- add_product_with_child_product() (shoop.front.basket.objects.BaseBasket method), 279
- add_resource() (in module shoop.xtheme.resources), 371
- add_resource() (shoop.xtheme.resources.ResourceContainer method), 370
- add_taxes() (shoop.core.taxing.TaxModule method), 253
- add_notification (class in shoop.notify.actions), 298
- AddNotification (class in shoop.notify.actions.notification), 297
- AddonEnableDisableForm (class in shoop.addons.admin_module.views.list), 38
- AddonListView (class in shoop.addons.admin_module.views), 39
- AddonListView (class in shoop.addons.admin_module.views.list), 38
- AddonModule (class in shoop.addons.admin_module), 40
- AddonUploadConfirmView (class in shoop.addons.admin_module.views), 40
- AddonUploadConfirmView (class in shoop.addons.admin_module.views.upload), 39
- AddonUploadForm (class in shoop.addons.admin_module.views.upload), 40
- AddonUploadView (class in shoop.addons.admin_module.views), 40
- AddonUploadView (class in shoop.addons.admin_module.views.upload), 39
- AddOrderLogEntry (class in shoop.notify.actions), 298
- AddOrderLogEntry (class in shoop.notify.actions.order), 298
- address (shoop.core.models.SavedAddress attribute), 223
- address_form_class (shoop.front.checkout.addresses.AddressesPhase attribute), 282
- address_form_classes (shoop.front.checkout.addresses.AddressesPhase attribute), 282
- address_kinds (shoop.front.checkout.addresses.AddressesPhase attribute), 282
- AddressesPhase (class in shoop.front.checkout.addresses), 282
- AddressForm (class in shoop.admin.modules.contacts.views.edit), 55
- AddressForm (class in shoop.front.apps.customer_information.forms), 272
- AddressForm (class in shoop.front.checkout.addresses), 282
- AddressForm.Meta (class in shoop.admin.modules.contacts.views.edit), 55
- AddressForm.Meta (class in shoop.front.apps.customer_information.forms), 272
- AddressForm.Meta (class in shoop.front.apps.customer_information.forms), 272
- AddressForm.Meta (class in shoop.front.apps.customer_information.forms), 272

- shoop.front.checkout.addresses), 282
- AddressSerializer (class in shoop.core.api.orders), 179
- AddressSerializer.Meta (class in shoop.core.api.orders), 179
- adjust_stock() (shoop.core.models.Supplier method), 233
- adjust_stock() (shoop.core.suppliers.base.BaseSupplierModule method), 251
- adjust_stock() (shoop.core.suppliers.BaseSupplierModule method), 251
- adjust_stock() (shoop.simple_supplier.module.SimpleSupplierModule method), 324
- admin, 18
- admin_form_class (shoop.campaigns.models.basket_effects.BasketDiscountEffect attribute), 152
- admin_form_class (shoop.campaigns.models.basket_effects.BasketDiscountEffect attribute), 154
- admin_form_class (shoop.campaigns.models.basket_line_effects.BasketLineEffect attribute), 155
- admin_form_class (shoop.campaigns.models.BasketDiscountEffect attribute), 171
- admin_form_class (shoop.campaigns.models.BasketLineEffect attribute), 170
- admin_form_class (shoop.campaigns.models.product_effects.ProductDiscountEffect attribute), 167
- admin_form_class (shoop.campaigns.models.product_effects.ProductDiscountEffect attribute), 169
- admin_form_class (shoop.campaigns.models.ProductDiscountEffect attribute), 173
- admin_url() (in module shoop.admin.utils.urls), 121
- admin_url_suffix (shoop.campaigns.models.BasketCampaign attribute), 170
- admin_url_suffix (shoop.campaigns.models.Campaign attribute), 173
- admin_url_suffix (shoop.campaigns.models.campaigns.BasketCampaign attribute), 158
- admin_url_suffix (shoop.campaigns.models.campaigns.Campaign attribute), 157
- admin_url_suffix (shoop.campaigns.models.campaigns.CatalogCampaign attribute), 157
- admin_url_suffix (shoop.campaigns.models.campaigns.Coupon attribute), 159
- admin_url_suffix (shoop.campaigns.models.CatalogCampaign attribute), 174
- admin_url_suffix (shoop.campaigns.models.Coupon attribute), 176
- AdminFieldRenderer (class in shoop.admin.utils.bs3_renderers), 117
- AdminModule (class in shoop.admin.base), 124
- AdminRegexURLPattern (class in shoop.admin.utils.urls), 121
- ADMINS (shoop.notify.enums.RecipientType attribute), 309
- ALL (shoop.notify.enums.StepConditionOperator attribute), 309
- allow_constant (shoop.notify.base.Binding attribute), 307
- allow_constant (shoop.notify.Binding attribute), 314
- allow_fractions (shoop.core.models.SalesUnit attribute), 223
- allow_unsaved_instance_assignment (shoop.core.fields.UnsavedForeignKey attribute), 185
- allow_variable (shoop.notify.base.Binding attribute), 307
- allow_variable (shoop.notify.Binding attribute), 314
- allowed_media_kinds (shoop.admin.modules.products.forms.base_forms.BaseForm attribute), 77
- allowed_media_kinds (shoop.admin.modules.products.forms.base_forms.ProductImageForm attribute), 77
- allowed_media_kinds (shoop.admin.modules.products.forms.base_forms.ProductMediaForm attribute), 77
- allowed_media_kinds (shoop.admin.modules.products.forms.ProductImageForm attribute), 77
- allowed_media_kinds (shoop.admin.modules.products.forms.ProductMediaForm attribute), 77
- AlwaysChangedModelForm (class in shoop.admin.modules.services.forms), 97
- amount (shoop.campaigns.models.basket_conditions.BasketMaxTotalAmount attribute), 160
- amount (shoop.campaigns.models.basket_conditions.BasketTotalAmount attribute), 160
- amount (shoop.core.models.OrderLineTax attribute), 204
- amount (shoop.core.models.Payment attribute), 205
- amount (shoop.core.models.Tax attribute), 234
- amount (shoop.core.pricing.Price attribute), 245
- amount (shoop.core.taxing.LineTax attribute), 253
- AnonymousContact (class in shoop.core.models), 187
- anonymouscontact (shoop.core.models.Contact attribute), 192
- AnonymousContact.DoesNotExist, 187
- AnonymousContact.MultipleObjectsReturned, 187
- AnonymousContact (shoop.notify.enums.StepConditionOperator attribute), 309
- AppConfig (class in shoop.apps), 134
- AppConfig (class in shoop.default_tax), 268
- AppConfig (class in shoop.front.apps.customer_information), 273
- AppConfig (class in shoop.front.apps.personal_order_history), 273
- AppConfig (class in shoop.guide.apps), 296
- AppConfig (class in shoop.order_printouts), 316
- AppConfig (class in shoop.simple_cms), 321
- apply_for_basket() (shoop.campaigns.models.basket_effects.BasketDiscountEffect method), 153
- apply_for_basket() (shoop.campaigns.models.basket_effects.BasketDiscountEffect method), 152
- apply_for_basket() (shoop.campaigns.models.basket_effects.BasketDiscountEffect method), 154
- apply_for_basket() (shoop.campaigns.models.BasketDiscountEffect method), 171

`apply_for_product()` (shoop.campaigns.models.product_effects.ProductDiscountEffect method), 168

`apply_for_product()` (shoop.campaigns.models.product_effects.ProductDiscountEffect method), 167

`apply_for_product()` (shoop.campaigns.models.product_effects.ProductDiscountPercentage method), 169

`apply_for_product()` (shoop.campaigns.models.ProductDiscountEffect method), 174

`apply_monkeypatch()` (in module shoop.api.encoders), 131

`apply_request_middleware()` (in module shoop.testing.utils), 336

`appropriate()` (shoop.xtheme.models.SavedViewConfigQuerySet method), 365

`as_dict()` (shoop.front.basket.storage.DictStoredBasket method), 281

`as_normalized()` (shoop.utils.patterns.Pattern method), 347

`assign_to_group_based_on_sales()` (in module shoop.campaigns.utils), 178

`attach_to_campaign()` (shoop.campaigns.models.campaigns.Coupon method), 160

`attach_to_campaign()` (shoop.campaigns.models.Coupon method), 176

`attached` (shoop.campaigns.models.campaigns.Coupon attribute), 160

`attached` (shoop.campaigns.models.Coupon attribute), 176

`Attribute` (class in shoop.core.models), 188

`attribute` (shoop.core.models.ProductAttribute attribute), 218

`Attribute.DoesNotExist`, 188

`Attribute.MultipleObjectsReturned`, 188

`AttributeEditView` (class in shoop.admin.modules.attributes.views), 47

`AttributeEditView` (class in shoop.admin.modules.attributes.views.edit), 47

`AttributeForm` (class in shoop.admin.modules.attributes.views.edit), 47

`AttributeForm.Meta` (class in shoop.admin.modules.attributes.views.edit), 47

`AttributeListView` (class in shoop.admin.modules.attributes.views), 48

`AttributeListView` (class in shoop.admin.modules.attributes.views.list), 47

`AttributeModule` (class in shoop.admin.modules.attributes), 48

`attributes` (shoop.core.models.Product attribute), 210, 214

`attributes` (shoop.core.models.ProductType attribute), 220

`AttributeType` (class in shoop.core.models), 188

`AttributeVisibility` (class in shoop.core.models), 189

`AUDIT` (shoop.utils.analog.LogEntryKind attribute), 338

`AuthAppConfig` (class in shoop.front.apps.auth), 271

`AutoRouter` (class in shoop.xthemeClassic_gray.ClassicGrayTheme attribute), 337

`AutoRouter` (class in shoop.xthemeClassic_gray.ClassicGrayTheme attribute), 375

`AutoRouter` (class in shoop.api.urls), 131

`AutoRouter` (class in shoop.api.urls), 131

B

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

`base_amount` (shoop.core.models.OrderLineTax attribute), 204

- `base_fields` (shoop.admin.modules.products.forms.base_form_base_fields attribute), 76
- `base_fields` (shoop.admin.modules.products.forms.base_form_base_fields attribute), 76
- `base_fields` (shoop.admin.modules.products.forms.base_form_base_fields attribute), 75
- `base_fields` (shoop.admin.modules.products.forms.base_form_base_fields attribute), 77
- `base_fields` (shoop.admin.modules.products.forms.base_form_base_fields attribute), 77
- `base_fields` (shoop.admin.modules.products.forms.base_form_base_fields attribute), 76
- `base_fields` (shoop.admin.modules.products.forms.BaseProductMediaForm attribute), 79
- `base_fields` (shoop.admin.modules.products.forms.package_base_fields attribute), 78
- `base_fields` (shoop.admin.modules.products.forms.PackageChildFields attribute), 80
- `base_fields` (shoop.admin.modules.products.forms.ProductAttributeFields attribute), 80
- `base_fields` (shoop.admin.modules.products.forms.ProductBaseForm attribute), 80
- `base_fields` (shoop.admin.modules.products.forms.ProductImageMediaForm attribute), 80
- `base_fields` (shoop.admin.modules.products.forms.ProductMediaForm attribute), 81
- `base_fields` (shoop.admin.modules.products.forms.ShopProductForm attribute), 81
- `base_fields` (shoop.admin.modules.products.forms.simple_variation_base_fields attribute), 78
- `base_fields` (shoop.admin.modules.products.forms.SimpleVariationChildForm attribute), 82
- `base_fields` (shoop.admin.modules.products.forms.variable_base_fields attribute), 79
- `base_fields` (shoop.admin.modules.products.forms.variable_base_fields attribute), 79
- `base_fields` (shoop.admin.modules.products.forms.VariableVariationChildForm attribute), 82
- `base_fields` (shoop.admin.modules.products.forms.VariationBaseForm attribute), 82
- `base_fields` (shoop.admin.modules.products.views.edit_cross_base_fields attribute), 84
- `base_fields` (shoop.admin.modules.products.views.edit_media_base_fields attribute), 85
- `base_fields` (shoop.admin.modules.sales_units.views.edit_sale_base_fields attribute), 91
- `base_fields` (shoop.admin.modules.service_providers.forms.CostFields attribute), 94
- `base_fields` (shoop.admin.modules.service_providers.forms.CostFields attribute), 94
- `base_fields` (shoop.admin.modules.services.forms.AlwaysChangeModelForm attribute), 97
- `base_fields` (shoop.admin.modules.services.forms.BaseMethodForm attribute), 97
- `base_fields` (shoop.admin.modules.services.forms.BaseMethodForm attribute), 97
- `base_fields` (shoop.admin.modules.services.forms.FixedCostBehaviorComp attribute), 98
- `base_fields` (shoop.admin.modules.services.forms.GroupAvailabilityBehav attribute), 100
- `base_fields` (shoop.admin.modules.services.forms.PaymentMethodForm attribute), 98
- `base_fields` (shoop.admin.modules.services.forms.RoundingBehaviorComp attribute), 101
- `base_fields` (shoop.admin.modules.services.forms.ShippingMethodForm attribute), 98
- `base_fields` (shoop.admin.modules.services.forms.StaffOnlyBehaviorComp attribute), 100
- `base_fields` (shoop.admin.modules.services.forms.WaivingCostBehaviorCor attribute), 99
- `base_fields` (shoop.admin.modules.services.forms.WeightLimitsBehaviorCo attribute), 99
- `base_fields` (shoop.admin.modules.shops.views.edit.ContactAddressForm attribute), 103
- `base_fields` (shoop.admin.modules.shops.views.edit.ShopBaseForm attribute), 102
- `base_fields` (shoop.admin.modules.suppliers.views.edit.SupplierForm attribute), 105
- `base_fields` (shoop.admin.modules.taxes.views.edit.CustomerTaxGroupForm attribute), 108
- `base_fields` (shoop.admin.modules.taxes.views.edit.TaxClassForm attribute), 108
- `base_fields` (shoop.admin.modules.taxes.views.edit.TaxForm attribute), 107
- `base_fields` (shoop.admin.modules.users.views.detail.BaseUserForm attribute), 111
- `base_fields` (shoop.admin.modules.users.views.detail.PasswordChangeF attribute), 112
- `base_fields` (shoop.admin.modules.variables.views.DataForm attribute), 113
- `base_fields` (shoop.admin.modules.campaigns.admin_module.form_parts.SalesRangesForm attribute), 142
- `base_fields` (shoop.admin.modules.campaigns.admin_module.forms.BasketCampaignForm attribute), 135
- `base_fields` (shoop.admin.modules.campaigns.admin_module.forms.BasketDiscountAmount attribute), 135
- `base_fields` (shoop.admin.modules.campaigns.admin_module.forms.BasketDiscountPercent attribute), 136
- `base_fields` (shoop.admin.modules.campaigns.admin_module.forms.BasketMaxTotalAmount attribute), 136
- `base_fields` (shoop.admin.modules.campaigns.admin_module.forms.BasketMaxTotalProduct attribute), 136
- `base_fields` (shoop.admin.modules.campaigns.admin_module.forms.BasketTotalAmountCo attribute), 136
- `base_fields` (shoop.admin.modules.campaigns.admin_module.forms.BasketTotalProductAm attribute), 137
- `base_fields` (shoop.admin.modules.campaigns.admin_module.forms.CatalogCampaignForm attribute), 137

base_fields (shoop.campaigns.admin_module.forms.CategoryFilterForm (shoop.notify.admin_module.forms.ScriptForm attribute), 137 attribute), 301

base_fields (shoop.campaigns.admin_module.forms.ContactBaseFieldForm (shoop.notify.admin_module.forms.ScriptItemEditForm attribute), 137 attribute), 301

base_fields (shoop.campaigns.admin_module.forms.ContactBaseFieldForm (shoop.simple_cms.admin_module.views.PageForm attribute), 137 attribute), 317

base_fields (shoop.campaigns.admin_module.forms.ContactBaseFieldForm (shoop.plugins.PageLinksConfigForm attribute), 138 attribute), 319

base_fields (shoop.campaigns.admin_module.forms.ContactBaseFieldForm (shoop.simple_supplier.admin_module.forms.SimpleSupplierForm attribute), 138 attribute), 321

base_fields (shoop.campaigns.admin_module.forms.CouponForm (shoop.simple_supplier.forms.StockAdjustmentForm attribute), 138 attribute), 322

base_fields (shoop.campaigns.admin_module.forms.DiscountForm (shoop.testing.admin_module.mock_view.MockForm attribute), 138 attribute), 325

base_fields (shoop.campaigns.admin_module.forms.FreeProductForm (shoop.testing.service_forms.CarrierWithCheckoutPhaseForm attribute), 139 attribute), 334

base_fields (shoop.campaigns.admin_module.forms.ProductBaseFieldForm (shoop.testing.service_forms.PaymentWithCheckoutPhaseForm attribute), 139 attribute), 333

base_fields (shoop.campaigns.admin_module.forms.ProductBaseFieldForm (shoop.testing.service_forms.PseudoPaymentProcessorForm attribute), 139 attribute), 333

base_fields (shoop.campaigns.admin_module.forms.ProductBaseFieldForm (shoop.testing.simple_checkout_phase.TestPaymentCheckoutPhaseForm attribute), 139 attribute), 334

base_fields (shoop.campaigns.admin_module.forms.ProductBaseFieldForm (shoop.testing.simple_checkout_phase.TestShipmentCheckoutPhaseForm attribute), 140 attribute), 334

base_fields (shoop.campaigns.admin_module.forms.ProductBaseFieldForm (shoop.utils.multilanguage_model_form.MultiLanguageModelForm attribute), 140 attribute), 344

base_fields (shoop.customer_group_pricing.admin_form_part_classes (shoop.admin_module.views.ActivationForm attribute), 264 attribute), 351

base_fields (shoop.default_tax.admin_module.views.TaxRuleForm (shoop.xtheme.forms.GenericThemeForm attribute), 266 attribute), 361

base_fields (shoop.front.apps.auth.forms.EmailAuthenticationForm (shoop.xtheme.plugins.category_links.CategoryLinksConfigForm attribute), 269 attribute), 353

base_fields (shoop.front.apps.auth.forms.RecoverPasswordForm (shoop.xtheme.plugins.forms.GenericPluginForm attribute), 270 attribute), 354

base_fields (shoop.front.apps.customer_information.forms.AddressForm (shoop.xtheme.plugins.forms.PluginForm attribute), 272 attribute), 353

base_fields (shoop.front.apps.customer_information.forms.CompanyForm (shoop.xtheme.plugins.social_media_links.SocialMediaLinksPluginForm attribute), 272 attribute), 355

base_fields (shoop.front.apps.customer_information.forms.PersonForm (shoop.xtheme.views.forms.LayoutCellGeneralInfoForm attribute), 271 attribute), 359

base_fields (shoop.front.apps.simple_search.views.SearchForm (shoop.admin.form_part.FormPartsViewMixin attribute), 276 attribute), 127

base_fields (shoop.front.checkout.addresses.AddressForm (shoop.admin.modules.categories.views.CategoryForm attribute), 282 attribute), 49

base_fields (shoop.front.checkout.addresses.CompanyForm (shoop.admin.modules.categories.views.edit.CategoryForm attribute), 282 attribute), 49

base_fields (shoop.front.checkout.confirm.ConfirmForm (shoop.admin.modules.contact_groups.views.ContactForm attribute), 283 attribute), 53

base_fields (shoop.front.checkout.methods.MethodsForm (shoop.admin.modules.contact_groups.views.edit.ContactForm attribute), 284 attribute), 51

base_fields (shoop.front.checkout.single_page.OrderForm (shoop.admin.modules.contacts.views.ContactEditForm attribute), 285 attribute), 57

base_fields (shoop.front.views.category.ProductListForm (shoop.admin.modules.contacts.views.edit.ContactForm attribute), 291 attribute), 55

395

base_objects (shoop.core.models.WeightBasedPricingBehaviorComponent attribute), 237	BaseProductMediaFormPart (class in shoop.admin.modules.products.views.edit), 83
base_objects (shoop.core.models.WeightLimitsBehaviorComponent attribute), 237	BaseProductMediaFormSet (class in shoop.admin.modules.products.forms.base_forms), 76
base_objects (shoop.testing.models.CarrierWithCheckoutPhase attribute), 327	BaseSupplierModule (class in shoop.core.suppliers), 251
base_objects (shoop.testing.models.ExpensiveSwedenBehaviorComponent attribute), 327	BaseSupplierModule (class in shoop.core.suppliers.base), 251
base_objects (shoop.testing.models.PaymentWithCheckoutPhase attribute), 328	BaseUserForm (class in shoop.admin.modules.users.views.detail), 111
base_objects (shoop.testing.models.PseudoPaymentProcessor attribute), 328	BasicServiceCheckoutPhaseProvider (class in shoop.front.checkout), 285
base_price (shoop.core.pricing.Priceful attribute), 246	BASKET (shoop.campaigns.models.campaigns.CampaignType attribute), 157
base_price (shoop.core.pricing.PriceInfo attribute), 248	basket_partial() (in shoop.themes.classic_gray.views), 336
base_translations (shoop.core.models.ServiceProvider attribute), 227	BasketBaseFormPart (class in shoop.campaigns.admin_module.form_parts), 142
base_unit_price (shoop.core.models.OrderLine attribute), 203	BasketCampaign (class in shoop.campaigns.models), 170
base_unit_price (shoop.core.order_creator.SourceLine attribute), 242	BasketCampaign (class in shoop.campaigns.models.campaigns), 158
base_unit_price (shoop.core.pricing.Priceful attribute), 246	BasketCampaign.DoesNotExist, 159, 170
BaseActionButton (class in shoop.admin.toolbar), 128	BasketCampaign.MultipleObjectsReturned, 159, 170
BaseBasket (class in shoop.front.basket.objects), 279	basketcampaign_set (shoop.core.models.Shop attribute), 230
BaseCheckoutView (class in shoop.front.views.checkout), 292	BasketCampaignEditView (class in shoop.campaigns.admin_module.views), 140
BaseEqual (class in shoop.notify.conditions.simple), 302	BasketCampaignForm (class in shoop.campaigns.admin_module.forms), 135
BaseFormPart (class in shoop.campaigns.admin_module.form_parts), 142	BasketCampaignForm.Meta (class in shoop.campaigns.admin_module.forms), 135
BaseFormset (class in shoop.campaigns.admin_module.form_sets), 143	BasketCampaignListView (class in shoop.campaigns.admin_module.views), 141
BaseImageGenerator (class in shoop.testing.image_generator), 332	BasketCampaignModule (class in shoop.campaigns.modules), 178
BaseLogEntry (class in shoop.utils.analog), 338	BasketCampaignTranslation (class in shoop.campaigns.models.campaigns), 160
BaseLogEntry.Meta (class in shoop.utils.analog), 338	BasketCampaignTranslation.DoesNotExist, 160
BaseMetaclass (class in shoop.notify.base), 306	BasketCampaignTranslation.MultipleObjectsReturned, 160
BaseMethodForm (class in shoop.admin.modules.services.forms), 97	BasketCommandDispatcher (class in shoop.front.basket.command_dispatcher), 277
BaseMethodForm.Meta (class in shoop.admin.modules.services.forms), 97	BasketCompatibilityError, 280
BasePopupChoiceWidget (class in shoop.admin.forms.widgets), 45	BasketCondition (class in shoop.campaigns.models), 172
BaseProductMediaForm (class in shoop.admin.modules.products.forms), 79	BasketCondition (class in shoop.campaigns.models.basket_conditions),
BaseProductMediaForm (class in shoop.admin.modules.products.forms.base_forms), 76	
BaseProductMediaForm.Meta (class in shoop.admin.modules.products.forms), 79	
BaseProductMediaForm.Meta (class in shoop.admin.modules.products.forms.base_forms), 76	

145

BasketCondition.DoesNotExist, 145, 172

BasketCondition.MultipleObjectsReturned, 146, 172

basketcondition_ptr (shoop.campaigns.models.basket_conditions.BasketCondition attribute), 149

basketcondition_ptr (shoop.campaigns.models.basket_conditions.BasketCondition attribute), 148

basketcondition_ptr (shoop.campaigns.models.basket_conditions.BasketCondition attribute), 147

basketcondition_ptr (shoop.campaigns.models.basket_conditions.BasketCondition attribute), 147

basketcondition_ptr (shoop.campaigns.models.basket_conditions.BasketCondition attribute), 152

basketcondition_ptr (shoop.campaigns.models.basket_conditions.BasketCondition attribute), 151

basketcondition_ptr (shoop.campaigns.models.basket_conditions.BasketCondition attribute), 150

BasketConditionsFormPart (class in shoop.campaigns.admin_module.form_parts), 143

BasketConditionsFormSet (class in shoop.campaigns.admin_module.form_sets), 144

BasketDiscountAmount (class in shoop.campaigns.models.basket_effects), 153

basketdiscountamount (shoop.campaigns.models.basket_effects.BasketDiscountAmount attribute), 152

basketdiscountamount (shoop.campaigns.models.BasketDiscountEffect attribute), 172

BasketDiscountAmount.DoesNotExist, 153

BasketDiscountAmount.MultipleObjectsReturned, 153

BasketDiscountAmountForm (class in shoop.campaigns.admin_module.forms), 135

BasketDiscountAmountForm.Meta (class in shoop.campaigns.admin_module.forms), 135

BasketDiscountEffect (class in shoop.campaigns.models), 171

BasketDiscountEffect (class in shoop.campaigns.models.basket_effects), 152

BasketDiscountEffect.DoesNotExist, 152, 171

BasketDiscountEffect.MultipleObjectsReturned, 152, 171

basketdiscounteffect_ptr (shoop.campaigns.models.basket_effects.BasketDiscountEffect attribute), 153

basketdiscounteffect_ptr (shoop.campaigns.models.basket_effects.BasketDiscountEffect attribute), 154

BasketDiscountEffectsFormPart (class in shoop.campaigns.admin_module.form_parts), 143

BasketDiscountEffectsFormSet (class in shoop.campaigns.admin_module.form_sets), 144

shoop.campaigns.admin_module.form_sets), 144

BasketDiscountPercentage (class in shoop.campaigns.models.basket_effects), 153

BasketDiscountPercentageForm (class in shoop.campaigns.admin_module.forms), 136

BasketDiscountPercentageForm.Meta (class in shoop.campaigns.admin_module.forms), 136

BasketLine (class in shoop.front.basket.objects), 279

BasketLineEffect (class in shoop.campaigns.models), 169

BasketLineEffect (class in shoop.campaigns.models.basket_line_effects), 154

BasketLineEffect.DoesNotExist, 155, 170

BasketLineEffect.MultipleObjectsReturned, 155, 170

basketlineeffect_ptr (shoop.campaigns.models.basket_line_effects.Discount attribute), 157

basketlineeffect_ptr (shoop.campaigns.models.basket_line_effects.FreeProduct attribute), 156

BasketLineEffectsFormPart (class in shoop.campaigns.admin_module.form_parts), 143

BasketLineEffectsFormSet (class in shoop.campaigns.admin_module.form_sets), 144

BasketMaxTotalAmountCondition (class in shoop.campaigns.models.basket_conditions), 148

basketmaxtotalamountcondition (shoop.campaigns.models.basket_conditions.BasketCondition attribute), 146

basketmaxtotalamountcondition (shoop.campaigns.models.BasketCondition attribute), 172

BasketMaxTotalAmountCondition.DoesNotExist, 149

BasketMaxTotalAmountCondition.MultipleObjectsReturned, 149

BasketMaxTotalAmountConditionForm (class in shoop.campaigns.admin_module.forms), 136

BasketMaxTotalAmountConditionForm.Meta (class in shoop.campaigns.admin_module.forms), 136

- `BasketMaxTotalProductAmountCondition` (class in `shoop.campaigns.models.basket_conditions`), 148
- `basketmaxtotalproductamountcondition` (`shoop.campaigns.models.basket_conditions.BasketCondition` module `shoop.utils.iterables`), 146
- `basketmaxtotalproductamountcondition` (`shoop.campaigns.models.BasketCondition` attribute), 172
- `BasketMaxTotalProductAmountCondition.DoesNotExist`, 148
- `BasketMaxTotalProductAmountCondition.MultipleObjectsReturned`, 148
- `BasketMaxTotalProductAmountConditionForm` (class in `shoop.campaigns.admin_module.forms`), 136
- `BasketMaxTotalProductAmountConditionForm.Meta` (class in `shoop.campaigns.admin_module.forms`), 136
- `BasketOrderCreator` (class in `shoop.front.basket.order_creator`), 280
- `BasketStorage` (class in `shoop.front.basket.storage`), 280
- `BasketTotalAmountCondition` (class in `shoop.campaigns.models.basket_conditions`), 147
- `baskettotalamountcondition` (`shoop.campaigns.models.basket_conditions.BasketCondition` attribute), 146
- `baskettotalamountcondition` (`shoop.campaigns.models.BasketCondition` attribute), 172
- `BasketTotalAmountCondition.DoesNotExist`, 147
- `BasketTotalAmountCondition.MultipleObjectsReturned`, 147
- `BasketTotalAmountConditionForm` (class in `shoop.campaigns.admin_module.forms`), 136
- `BasketTotalAmountConditionForm.Meta` (class in `shoop.campaigns.admin_module.forms`), 136
- `BasketTotalProductAmountCondition` (class in `shoop.campaigns.models.basket_conditions`), 146
- `basketttotalproductamountcondition` (`shoop.campaigns.models.basket_conditions.BasketCondition` attribute), 146
- `basketttotalproductamountcondition` (`shoop.campaigns.models.BasketCondition` attribute), 172
- `BasketTotalProductAmountCondition.DoesNotExist`, 147
- `BasketTotalProductAmountCondition.MultipleObjectsReturned`, 147
- `BasketTotalProductAmountConditionForm` (class in `shoop.campaigns.admin_module.forms`), 136
- `BasketTotalProductAmountConditionForm.Meta` (class in `shoop.campaigns.admin_module.forms`), 136
- `BasketUpdateMethods` (class in `shoop.front.basket.update_methods`), 281
- `BasketView` (class in `shoop.front.views.basket`), 291
- `begin_column()` (`shoop.xtheme.layout.Layout` method), 363
- `begin_row()` (`shoop.xtheme.layout.Layout` method), 363
- `behavior_components` (`shoop.core.models.PaymentMethod` attribute), 206
- `behavior_components` (`shoop.core.models.Service` attribute), 224
- `behavior_components` (`shoop.core.models.ShippingMethod` attribute), 228
- `BehaviorComponentFormPart` (class in `shoop.admin.modules.services.behavior_form_part`), 97
- `BehaviorFormSet` (class in `shoop.admin.modules.services.behavior_form_part`), 96
- `BILLING` (`shoop.core.models.SavedAddressRole` attribute), 224
- `billing_address` (`shoop.core.models.Order` attribute), 199
- `billing_address_form_class` (`shoop.front.checkout.single_page.SingleCheckoutPhase` attribute), 285
- `billing_orders` (`shoop.core.models.ImmutableAddress` attribute), 197
- `Binding` (class in `shoop.notify`), 314
- `Binding` (class in `shoop.notify.base`), 307
- `bindings` (`shoop.front.notify_events.OrderReceived` attribute), 294
- `bindings` (`shoop.notify.Action` attribute), 313
- `bindings` (`shoop.notify.actions.AddNotification` attribute), 298
- `bindings` (`shoop.notify.actions.AddOrderLogEntry` attribute), 298
- `bindings` (`shoop.notify.actions.debug.SetDebugFlag` attribute), 297
- `bindings` (`shoop.notify.actions.email.SendEmail` attribute), 297
- `bindings` (`shoop.notify.actions.notification.AddNotification` attribute), 297
- `bindings` (`shoop.notify.actions.order.AddOrderLogEntry` attribute), 298
- `bindings` (`shoop.notify.actions.SendEmail` attribute), 298
- `bindings` (`shoop.notify.actions.SetDebugFlag` attribute), 298
- `bindings` (`shoop.notify.base.Action` attribute), 308
- `bindings` (`shoop.notify.base.Base` attribute), 307
- `bindings` (`shoop.notify.base.Condition` attribute), 308
- `bindings` (`shoop.notify.base.Event` attribute), 307
- `bindings` (`shoop.notify.base.ScriptItem` attribute), 308
- `bindings` (`shoop.notify.Condition` attribute), 315

- bindings (shoop.notify.conditions.BooleanEqual attribute), 303
- bindings (shoop.notify.conditions.Empty attribute), 303
- bindings (shoop.notify.conditions.IntegerEqual attribute), 303
- bindings (shoop.notify.conditions.LanguageEqual attribute), 303
- bindings (shoop.notify.conditions.NonEmpty attribute), 303
- bindings (shoop.notify.conditions.simple.BaseEqual attribute), 302
- bindings (shoop.notify.conditions.simple.CaseInsensitiveStringEqual attribute), 303
- bindings (shoop.notify.conditions.simple.Empty attribute), 302
- bindings (shoop.notify.conditions.simple.NonEmpty attribute), 302
- bindings (shoop.notify.conditions.TextEqual attribute), 304
- bindings (shoop.notify.Event attribute), 315
- BLOCK_TEMPLATE (shoop.admin.dashboard.blocks.DashboardChatBlock attribute), 43
- BLOCK_TEMPLATE (shoop.admin.dashboard.DashboardChatBlock attribute), 44
- Boolean (class in shoop.notify.typology), 312
- BOOLEAN (shoop.core.models.AttributeType attribute), 188
- BooleanEqual (class in shoop.notify.conditions), 303
- Bootstrap3Namespace (class in shoop.admin.templatetags.shoop_admin), 117
- BOUGHT_WITH (shoop.core.models.ProductCrossSellType attribute), 219
- Breadcrumbs (class in shoop.admin.breadcrumbs), 126
- breadcrumbs_menu_entry (shoop.addons.admin_module.AddonModule attribute), 40
- breadcrumbs_menu_entry (shoop.admin.base.AdminModule attribute), 124
- breadcrumbs_menu_entry (shoop.admin.modules.attributes.AttributeModule attribute), 48
- breadcrumbs_menu_entry (shoop.admin.modules.categories.CategoryModule attribute), 50
- breadcrumbs_menu_entry (shoop.admin.modules.contact_groups.ContactGroupModule attribute), 53
- breadcrumbs_menu_entry (shoop.admin.modules.contacts.ContactModule attribute), 57
- breadcrumbs_menu_entry (shoop.admin.modules.manufacturers.ManufacturerModule attribute), 60
- breadcrumbs_menu_entry (shoop.admin.modules.orders.OrderModule attribute), 71
- breadcrumbs_menu_entry (shoop.admin.modules.permission_groups.PermissionGroupModule attribute), 73
- breadcrumbs_menu_entry (shoop.admin.modules.product_types.ProductTypeModule attribute), 75
- breadcrumbs_menu_entry (shoop.admin.modules.products.ProductModule attribute), 91
- breadcrumbs_menu_entry (shoop.admin.modules.sales_units.SalesUnitModule attribute), 92
- breadcrumbs_menu_entry (shoop.admin.modules.services.PaymentMethodModule attribute), 102
- breadcrumbs_menu_entry (shoop.admin.modules.services.ShippingMethodModule attribute), 101
- breadcrumbs_menu_entry (shoop.admin.modules.shops.ShopModule attribute), 104
- breadcrumbs_menu_entry (shoop.admin.modules.suppliers.SupplierModule attribute), 106
- breadcrumbs_menu_entry (shoop.admin.modules.users.UserModule attribute), 116
- breadcrumbs_menu_entry (shoop.default_tax.admin_module.TaxRulesAdminModule attribute), 266
- breadcrumbs_menu_entry (shoop.notify.admin_module.NotifyAdminModule attribute), 301
- breadcrumbs_menu_entry (shoop.simple_cms.admin_module.SimpleCMSAdminModule attribute), 318
- breadcrumbs_menu_entry (shoop.xtheme.admin_module.XthemeAdminModule attribute), 352
- browse_kind (shoop.admin.forms.widgets.BasePopupChoiceWidget attribute), 45
- browse_kind (shoop.admin.forms.widgets.ContactChoiceWidget attribute), 46
- browse_kind (shoop.admin.forms.widgets.MediaChoiceWidget attribute), 45
- browse_kind (shoop.admin.forms.widgets.ProductChoiceWidget attribute), 46
- build() (shoop.admin.modules.contacts.views.detail.ContactDetailToolBar method), 54
- button_activate_button()

[CacheDetailToolBar \(class in shoop.admin.modules.contacts.views.detail.ContactDetailToolBar\)](#), 54
[build_form\(\) \(shoop.xtheme.views.editor.EditorView method\)](#), 358
[build_new_order_button\(\) \(shoop.admin.modules.contacts.views.detail.ContactDetailToolBar method\)](#), 54
[build_new_user_button\(\) \(shoop.admin.modules.contacts.views.detail.ContactDetailToolBar method\)](#), 54
[build_potfiles\(\) \(shoop.core.management.commands.make_messages.C command\)](#), 185
[build_renew_password_button\(\) \(shoop.admin.modules.contacts.views.detail.ContactDetailToolBar method\)](#), 54
[bump_version\(\) \(shoop.core.cache.impl.VersionedCache method\)](#), 181
[bump_version\(\) \(shoop.core.cache.VersionedCache method\)](#), 182
[ButtonGroup \(class in shoop.admin.toolbar\)](#), 130
[by_rendering_template\(\) \(shoop.admin.dashboard.blocks.DashboardBlock class method\)](#), 42
[by_rendering_template\(\) \(shoop.admin.dashboard.DashboardBlock class method\)](#), 44
C
[cache_info\(\) \(shoop.front.basket.objects.BasketLine method\)](#), 279
[cache_prices\(\) \(shoop.core.models.Order method\)](#), 199
[cache_product_things\(\) \(in module shoop.front.utils.views\)](#), 291
[cache_translations\(\) \(in module shoop.utils.translation\)](#), 350
[cache_translations_for_tree\(\) \(in module shoop.utils.translation\)](#), 350
[cache_values\(\) \(shoop.core.models.Shipment method\)](#), 228
[cache_values\(\) \(shoop.core.models.ShipmentProduct method\)](#), 228
[cached_load\(\) \(in module shoop.utils.importing\)](#), 343
[calculate_amount\(\) \(shoop.core.models.Tax method\)](#), 235
[calculate_compounded_added_taxes\(\) \(in module shoop.core.taxing.utils\)](#), 252
[calculate_taxes\(\) \(shoop.core.order_creator.OrderSource method\)](#), 239
[calculate_taxes_or_raise\(\) \(shoop.core.order_creator.OrderSource method\)](#), 239
[calculating_isCheap \(shoop.core.taxing.TaxModule attribute\)](#), 253
[callback \(shoop.admin.utils.urls.AdminRegexURLPattern attribute\)](#), 121
[camel_case\(\) \(in module shoop.utils.text\)](#), 350
[Campaign \(class in shoop.campaigns.models\)](#), 173
[Campaign \(class in shoop.campaigns.models.campaigns\)](#), 157
[CampaignAdminModule \(class in shoop.campaigns.admin_module\)](#), 145
[CampaignAppConfig \(class in shoop.campaigns.apps\)](#), 177
[CampaignBaseFormPart \(class in shoop.campaigns.admin_module.form_parts\)](#), 142
[CampaignType \(class in shoop.campaigns.models.campaigns\)](#), 157
[can_change_quantity \(shoop.front.basket.objects.BasketLine attribute\)](#), 279
[can_create_refund\(\) \(shoop.core.models.Order method\)](#), 199
[can_delete \(shoop.admin.modules.products.forms.base_forms.BaseProductForm attribute\)](#), 77
[can_delete \(shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSellForm attribute\)](#), 85
[can_delete \(shoop.admin.modules.products.views.edit_media.ProductMediaForm attribute\)](#), 86
[can_delete \(shoop.admin.modules.services.behavior_form_part.BehaviorFormPart attribute\)](#), 96

[can_delete \(shoop.campaigns.admin_module.form_sets.BaseFormSet attribute\), 144](#)
[can_delete \(shoop.front.basket.objects.BasketLine attribute\), 279](#)
[can_delete\(\) \(shoop.core.models.ContactGroup method\), 193](#)
[can_delete\(\) \(shoop.core.models.PaymentMethod method\), 206](#)
[can_delete\(\) \(shoop.core.models.ShippingMethod method\), 228](#)
[can_edit\(\) \(shoop.core.models.Order method\), 199](#)
[can_manage_stock\(\) \(shoop.simple_supplier.admin_module.CasinoSimpleSupplierFormSet method\), 321](#)
[can_order \(shoop.admin.modules.products.forms.base_forms.CatalogCampaignFormSet attribute\), 77](#)
[can_order \(shoop.admin.modules.products.views.edit_cross_sell_base_forms.CrossSellFormSet attribute\), 85](#)
[can_order \(shoop.admin.modules.products.views.edit_media.ProductMediaFormSet attribute\), 86](#)
[can_order \(shoop.admin.modules.services.behavior_form_part.BehaviorFormSet attribute\), 96](#)
[can_order \(shoop.campaigns.admin_module.form_sets.BaseFormset attribute\), 144](#)
[can_set_canceled\(\) \(shoop.core.models.Order method\), 199](#)
[can_set_complete\(\) \(shoop.core.models.Order method\), 199](#)
[can_use_code\(\) \(shoop.campaigns.models.campaigns.Coupon method\), 160](#)
[can_use_code\(\) \(shoop.campaigns.models.Coupon method\), 177](#)
[can_use_code\(\) \(shoop.campaigns.modules.BasketCampaignModule method\), 178](#)
[can_use_code\(\) \(shoop.core.order_creator.OrderSourceModifierModule method\), 241](#)
[CANCELED \(shoop.core.models.OrderStatusRole attribute\), 205](#)
[CANCELED \(shoop.core.models.PaymentStatus attribute\), 208](#)
[Carrier \(class in shoop.core.models\), 189](#)
[carrier \(shoop.core.models.ServiceProvider attribute\), 227](#)
[carrier \(shoop.core.models.ShippingMethod attribute\), 228](#)
[Carrier.DoesNotExist, 189](#)
[Carrier.MultipleObjectsReturned, 189](#)
[carrier_ptr \(shoop.core.models.CustomCarrier attribute\), 195](#)
[CarrierWithCheckoutPhase \(class in shoop.testing.models\), 327](#)
[carrierwithcheckoutphase \(shoop.core.models.CustomCarrier attribute\), 195](#)
[CarrierWithCheckoutPhase.DoesNotExist, 327](#)
[CarrierWithCheckoutPhase.MultipleObjectsReturned, 327](#)
[CarrierWithCheckoutPhaseForm \(class in shoop.testing.service_forms\), 333](#)
[CarrierWithCheckoutPhaseForm.Meta \(class in shoop.testing.service_forms\), 333](#)
[CartAdminModule \(class in shoop.front.admin_module\), 269](#)
[CartListView \(class in shoop.front.admin_module.carts.views\), 268](#)
[CatalogCampaign \(class in shoop.campaigns.models\), 157](#)
[CatalogCampaign.DoesNotExist, 158, 174](#)
[CatalogCampaign.MultipleObjectsReturned, 158, 174](#)
[CatalogCampaignSet \(shoop.core.models.Shop attribute\), 230](#)
[CatalogCampaignEditView \(class in shoop.campaigns.admin_module.views\), 140](#)
[CatalogCampaignForm \(class in shoop.campaigns.admin_module.forms\), 137](#)
[CatalogCampaignForm.Meta \(class in shoop.campaigns.admin_module.forms\), 137](#)
[CatalogCampaignListView \(class in shoop.campaigns.admin_module.views\), 140](#)
[CatalogCampaignModule \(class in shoop.campaigns.modules\), 178](#)
[CatalogCampaignTranslation \(class in shoop.campaigns.models.campaigns\), 161](#)
[CatalogCampaignTranslation.DoesNotExist, 161](#)
[CatalogCampaignTranslation.MultipleObjectsReturned, 161](#)
[CatalogConditionsFormPart \(class in shoop.campaigns.admin_module.form_parts\), 143](#)
[CatalogConditionsFormSet \(class in shoop.campaigns.admin_module.form_sets\), 144](#)
[CatalogEffectsFormPart \(class in shoop.campaigns.admin_module.form_parts\), 143](#)

[CatalogEffectsFormSet](#) (class in [shoop.campaigns.admin_module.form_sets](#)), [145](#)
[CatalogFilter](#) (class in [shoop.campaigns.models](#)), [175](#)
[CatalogFilter](#) (class in [shoop.campaigns.models.catalog_filters](#)), [161](#)
[CatalogFilter.DoesNotExist](#), [161](#), [175](#)
[CatalogFilter.MultipleObjectsReturned](#), [161](#), [175](#)
[catalogfilter_ptr](#) ([shoop.campaigns.models.catalog_filters.CategoryFilter](#) attribute), [164](#)
[catalogfilter_ptr](#) ([shoop.campaigns.models.catalog_filters.ProductTypeFilter](#) attribute), [163](#)
[catalogfilter_ptr](#) ([shoop.campaigns.models.catalog_filters.ProductTypeFilter](#) attribute), [162](#)
[CatalogFiltersFormPart](#) (class in [shoop.campaigns.admin_module.form_parts](#)), [143](#)
[CatalogFiltersFormSet](#) (class in [shoop.campaigns.admin_module.form_sets](#)), [144](#)
[categories](#) ([shoop.campaigns.models.catalog_filters.CategoryFilter](#) attribute), [163](#)
[categories](#) ([shoop.core.models.Shop](#) attribute), [230](#)
[categories](#) ([shoop.core.models.ShopProduct](#) attribute), [231](#)
[Category](#) (class in [shoop.core.models](#)), [190](#)
[category](#) ([shoop.addons.admin_module.AddonModule](#) attribute), [40](#)
[category](#) ([shoop.admin.modules.categories.CategoryModule](#) attribute), [50](#)
[category](#) ([shoop.admin.modules.contact_groups.ContactGroupModule](#) attribute), [53](#)
[category](#) ([shoop.admin.modules.contacts.ContactModule](#) attribute), [57](#)
[category](#) ([shoop.admin.modules.permission_groups.PermissionGroupModule](#) attribute), [73](#)
[category](#) ([shoop.admin.modules.service_providers.ServiceProviderModule](#) attribute), [94](#)
[category](#) ([shoop.admin.modules.services.ServiceModule](#) attribute), [101](#)
[category](#) ([shoop.admin.modules.system.SystemModule](#) attribute), [107](#)
[category](#) ([shoop.admin.modules.taxes.TaxModule](#) attribute), [111](#)
[category](#) ([shoop.admin.modules.users.UserModule](#) attribute), [116](#)
[category](#) ([shoop.core.models.Product](#) attribute), [210](#), [214](#)
[category](#) ([shoop.default_tax.admin_module.TaxRulesAdminModule](#) attribute), [266](#)
[category](#) ([shoop.front.apps.simple_order_notification.admin_module.SimpleOrderNotificationModule](#) attribute), [275](#)
[category](#) ([shoop.simple_supplier.admin_module.StocksAdminModule](#) attribute), [322](#)
[Category.DoesNotExist](#), [190](#)
[Category.MultipleObjectsReturned](#), [190](#)
[CategoryBaseForm](#) (class in [shoop.admin.modules.categories.views.edit](#)), [48](#)
[CategoryBaseForm.Meta](#) (class in [shoop.admin.modules.categories.views.edit](#)), [48](#)
[CategoryBaseFormPart](#) (class in [shoop.admin.modules.categories.views.edit](#)), [49](#)
[CategoryEditView](#) (class in [shoop.admin.modules.categories.views](#)), [49](#)
[CategoryFactory](#) (class in [shoop.testing.factories](#)), [330](#)
[CategoryFilter](#) (class in [shoop.campaigns.models.catalog_filters](#)), [163](#)
[categoryfilter](#) ([shoop.campaigns.models.catalog_filters.CatalogFilter](#) attribute), [161](#)
[categoryfilter](#) ([shoop.campaigns.models.CatalogFilter](#) attribute), [175](#)
[CategoryFilter.DoesNotExist](#), [164](#)
[CategoryFilter.MultipleObjectsReturned](#), [164](#)
[categoryfilter_set](#) ([shoop.core.models.Category](#) attribute), [190](#)
[CategoryFilterForm](#) (class in [shoop.campaigns.admin_module.forms](#)), [137](#)
[CategoryFilterForm.Meta](#) (class in [shoop.campaigns.admin_module.forms](#)), [137](#)
[CategoryModuleConfigForm](#) (class in [shoop.xtheme.plugins.category_links](#)), [353](#)
[CategoryModuleLinkPlugin](#) (class in [shoop.xtheme.plugins.category_links](#)), [353](#)
[CategoryListView](#) (class in [shoop.admin.modules.categories.views](#)), [50](#)
[CategoryListView](#) (class in [shoop.admin.modules.categories.views.list](#)), [49](#)
[CategoryModule](#) (class in [shoop.admin.modules.categories](#)), [50](#)
[CategoryStatus](#) (class in [shoop.core.models](#)), [191](#)
[CategoryView](#) (class in [shoop.front.views.category](#)), [291](#)
[CategoryVisibility](#) (class in [shoop.core.models](#)), [191](#)
[cell_class_template](#) ([shoop.xtheme.layout.Layout](#) attribute), [358](#)
[CELL_FULL_WIDTH](#) ([shoop.xtheme.views.forms.LayoutCellGeneralInfo](#) attribute), [358](#)
[CELL_WIDTH_CHOICES](#)

- (shoop.xtheme.views.forms.LayoutCellGeneralInformationForm attribute), 358
- CgpPrice (class in shoop.customer_group_pricing.models), 265
- CgpPrice.DoesNotExist, 265
- CgpPrice.MultipleObjectsReturned, 265
- cgpprice_set (shoop.core.models.ContactGroup attribute), 193
- cgpprice_set (shoop.core.models.Shop attribute), 230
- change_password() (in module shoop.front.apps.customer_information.views), 272
- change_protect_field_text (shoop.admin.modules.shops.views.edit.ShopBaseForm attribute), 102
- change_protect_field_text (shoop.core.utils.form_mixins.ProtectedFieldsMixin attribute), 257
- change_protect_message (shoop.core.models.Shop attribute), 230
- change_protect_message (shoop.core.models.Tax attribute), 235
- changed (shoop.xtheme.views.editor.EditorView attribute), 358
- Chart (class in shoop.admin.dashboard.charts), 43
- check_all_verified() (shoop.core.models.Order method), 199
- check_and_raise_if_only_one_allowed() (in module shoop.admin.utils.views), 123
- check_demo_optin() (shoop.admin.modules.demo.DemoModel method), 58
- checkout_process (shoop.front.checkout.CheckoutPhaseView attribute), 285
- CheckoutPhaseViewMixin (class in shoop.front.checkout), 285
- CheckoutProcess (class in shoop.front.checkout), 286
- child (shoop.core.models.ProductPackageLink attribute), 220
- child_lines (shoop.core.models.OrderLine attribute), 203
- children (shoop.core.models.Category attribute), 190
- children (shoop.simple_cms.models.Page attribute), 318
- choice() (shoop.front.checkout.methods.MethodChoiceIterator method), 284
- ChoicesFilter (class in shoop.admin.utils.picotable), 118
- class_for_identifier() (shoop.notify.base.Base class method), 307
- ClassicGrayTheme (class in shoop.themes.classic_gray), 337
- ClassicGrayThemeAppConfig (class in shoop.themes.classic_gray), 337
- clean() (shoop.admin.modules.orders.views.shipment.ShipmentForm method), 66
- clean() (shoop.admin.modules.permission_groups.views.edit.PermissionGroupForm method), 72
- clean() (shoop.admin.modules.users.views.permissions.PermissionChangeForm method), 113
- clean() (shoop.campaigns.models.contact_group_sales_ranges.ContactGroupSalesRange method), 164
- clean() (shoop.campaigns.models.ContactGroupSalesRange method), 175
- clean() (shoop.core.models.Tax method), 235
- clean() (shoop.core.utils.form_mixins.ProtectedFieldsMixin method), 257
- clean() (shoop.default_tax.admin_module.views.TaxRuleForm method), 266
- clean() (shoop.front.apps.auth.forms.RecoverPasswordForm method), 270
- clean() (shoop.front.apps.simple_search.views.SearchForm method), 276
- clean() (shoop.simple_cms.admin_module.views.PageForm method), 317
- clean() (shoop.simple_cms.plugins.PageLinksConfigForm method), 319
- clean() (shoop.xtheme.plugins.category_links.CategoryLinksConfigForm method), 353
- clean() (shoop.xtheme.plugins.forms.TranslatableField method), 354
- clean() (shoop.xtheme.plugins.image.ImageIDField method), 354
- clean() (shoop.xtheme.plugins.social_media_links.SocialMediaLinksPlugin method), 355
- clean_amount() (shoop.admin.modules.orders.views.refund.RefundForm method), 64
- clean_code() (shoop.campaigns.admin_module.forms.CouponForm method), 138
- clean_coupon() (shoop.campaigns.admin_module.forms.BasketCampaignForm method), 135
- clean_delta() (shoop.simple_supplier.forms.StockAdjustmentForm method), 322
- clean_empty_lines() (shoop.front.basket.objects.BaseBasket method), 279
- clean_external_url() (shoop.admin.modules.products.forms.base_forms.ProductMediaForm method), 77
- clean_external_url() (shoop.admin.modules.products.forms.ProductMediaForm method), 81
- clean_file() (shoop.admin.modules.products.forms.base_forms.ProductImageForm method), 77
- clean_file() (shoop.admin.modules.products.forms.ProductImageMediaForm method), 80
- clean_hook() (shoop.admin.modules.orders.views.shipment.ShipmentForm method), 65
- clean_line_number() (shoop.admin.modules.orders.views.refund.RefundForm method), 64
- clean_members() (shoop.admin.modules.permission_groups.views.edit.PermissionGroupForm method), 72
- clean_members() (shoop.admin.modules.users.views.permissions.PermissionChangeForm method), 113
- clean_minimum_purchase_quantity()

(`shoop.admin.modules.products.forms.base_forms.ShopProductForm` (in module `shoop.admin.modules.products.forms`), 76

`clean_minimum_purchase_quantity()` (`shoop.admin.modules.products.forms.ShopProductForm` (class in `shoop.admin.modules.products.forms`), 81

`clean_old_password()` (`shoop.admin.modules.users.views.password.PasswordChangeForm` (class in `shoop.admin.modules.users.views.password`), 112

`clean_old_password()` (`shoop.admin.modules.users.views.permissions.PermissionsChangeFormBase` (class in `shoop.admin.modules.users.views.permissions`), 113

`clean_password2()` (`shoop.admin.modules.users.views.password.PasswordChangeForm` (class in `shoop.admin.modules.users.views.password`), 112

`clean_permission_groups()` (`shoop.admin.modules.users.views.permissions.PermissionsChangeFormBase` (class in `shoop.admin.modules.users.views.permissions`), 113

`clean_protected_fields()` (`shoop.core.utils.form_mixins.ProtectedFieldsMixin` (class in `shoop.core.utils.form_mixins`), 257

`clean_quantity()` (`shoop.admin.modules.orders.views.refunds.RefundForm` (class in `shoop.admin.modules.orders.views.refunds`), 64

`clean_username()` (`shoop.front.apps.auth.forms.EmailAuthenticationForm` (class in `shoop.front.apps.auth.forms`), 269

`clear()` (`shoop.core.cache.impl.VersionedCache` (class in `shoop.core.cache.impl`), 181

`clear()` (`shoop.core.cache.VersionedCache` (class in `shoop.core.cache`), 182

`clear_all()` (`shoop.front.basket.objects.BaseBasket` (class in `shoop.front.basket.objects`), 279

`clear_codes()` (`shoop.campaigns.modules.BasketCampaignModule` (class in `shoop.campaigns.modules`), 178

`clear_codes()` (`shoop.core.order_creator.OrderSource` (class in `shoop.core.order_creator`), 239

`clear_codes()` (`shoop.core.order_creator.OrderSourceModifierModule` (class in `shoop.core.order_creator`), 241

`clear_codes()` (`shoop.front.basket.objects.BaseBasket` (class in `shoop.front.basket.objects`), 279

`clear_existing_package()` (in module `shoop.admin.modules.products.utils`), 91

`clear_layout_data()` (`shoop.xtheme.models.SavedViewConfig` (class in `shoop.xtheme.models`), 366

`clear_load_cache()` (in module `shoop.utils.importing`), 343

`clear_provides_cache()` (in module `shoop.apps.provides`), 132

`clear_variation()` (`shoop.core.models.Product` (class in `shoop.core.models`), 210, 214

`clear_view_cache()` (in module `shoop.xtheme.views.extra`), 358

`code` (`shoop.core.utils.vat.VatCannotIdentifyValidationError` (class in `shoop.core.utils.vat`), 259

`code` (`shoop.core.utils.vat.VatInvalidValidationError` (class in `shoop.core.utils.vat`), 260

`code` (`shoop.core.utils.vat.VatValidationError` (class in `shoop.core.utils.vat`), 259

`codes` (`shoop.core.models.Order` (class in `shoop.core.models`), 200

`codes` (`shoop.core.order_creator.OrderSource` (class in `shoop.core.order_creator`), 239

`collect_settings_from_app()` (in module `shoop.apps.settings`), 133

`columns` (`shoop.admin.modules.attributes.views.AttributeListView` (class in `shoop.admin.modules.attributes.views`), 119

`columns` (`shoop.admin.modules.attributes.views.list.AttributeListView` (class in `shoop.admin.modules.attributes.views.list`), 119

`columns` (`shoop.admin.modules.categories.views.CategoryListView` (class in `shoop.admin.modules.categories.views`), 119

`columns` (`shoop.admin.modules.categories.views.list.CategoryListView` (class in `shoop.admin.modules.categories.views.list`), 119

`columns` (`shoop.admin.modules.contact_groups.views.ContactGroupListView` (class in `shoop.admin.modules.contact_groups.views`), 119

`columns` (`shoop.admin.modules.contacts.views.ContactListView` (class in `shoop.admin.modules.contacts.views`), 56

`columns` (`shoop.admin.modules.manufacturers.views.ManufacturerListView` (class in `shoop.admin.modules.manufacturers.views`), 59

`columns` (`shoop.admin.modules.manufacturers.views.ManufacturerListView` (class in `shoop.admin.modules.manufacturers.views`), 59

`columns` (`shoop.admin.modules.orders.views.list.OrderListView` (class in `shoop.admin.modules.orders.views.list`), 63

`columns` (`shoop.admin.modules.orders.views.OrderListView` (class in `shoop.admin.modules.orders.views`), 67

`columns` (`shoop.admin.modules.permission_groups.views.list.PermissionGroupListView` (class in `shoop.admin.modules.permission_groups.views.list`), 73

`columns` (`shoop.admin.modules.permission_groups.views.PermissionGroupListView` (class in `shoop.admin.modules.permission_groups.views`), 73

`columns` (`shoop.admin.modules.product_types.views.list.ProductTypeListView` (class in `shoop.admin.modules.product_types.views.list`), 74

`columns` (`shoop.admin.modules.product_types.views.ProductTypeListView` (class in `shoop.admin.modules.product_types.views`), 75

`columns` (`shoop.admin.modules.products.views.list.ProductListView` (class in `shoop.admin.modules.products.views.list`), 88

`columns` (`shoop.admin.modules.products.views.ProductListView` (class in `shoop.admin.modules.products.views`), 89

`columns` (`shoop.admin.modules.sales_units.views.list.SalesUnitListView` (class in `shoop.admin.modules.sales_units.views.list`), 92

`columns` (`shoop.admin.modules.sales_units.views.SalesUnitListView` (class in `shoop.admin.modules.sales_units.views`), 92

`columns` (`shoop.admin.modules.service_providers.views.ServiceProviderListView` (class in `shoop.admin.modules.service_providers.views`), 93

`columns` (`shoop.admin.modules.shops.views.list.ShopListView` (class in `shoop.admin.modules.shops.views.list`), 104

`columns` (`shoop.admin.modules.shops.views.ShopListView` (class in `shoop.admin.modules.shops.views`), 104

`columns` (`shoop.admin.modules.suppliers.views.list.SupplierListView` (class in `shoop.admin.modules.suppliers.views.list`), 105

`columns` (`shoop.admin.modules.suppliers.views.SupplierListView` (class in `shoop.admin.modules.suppliers.views`), 106

405

- attribute), 174
- config_copied_variables (shoop.xtheme.TemplatedPlugin attribute), 375
- ConfigurationItem (class in shoop.core.models), 191
- ConfigurationItem.DoesNotExist, 191
- ConfigurationItem.MultipleObjectsReturned, 191
- configure() (shoop.utils.setup.Setup class method), 349
- confirm_login_allowed()
 - (shoop.front.apps.auth.forms.EmailAuthenticationForm method), 269
- ConfirmForm (class in shoop.front.checkout.confirm), 283
- ConfirmPhase (class in shoop.front.checkout.confirm), 283
- CONSTANT_ONLY (shoop.notify.ConstantUse attribute), 315
- CONSTANT_ONLY (shoop.notify.enums.ConstantUse attribute), 309
- ConstantUse (class in shoop.notify), 315
- ConstantUse (class in shoop.notify.enums), 309
- construct_simple() (in module shoop.notify.conditions.simple), 303
- Contact (class in shoop.core.models), 192
- Contact.DoesNotExist, 192
- Contact.MultipleObjectsReturned, 192
- contact_address (shoop.core.models.Shop attribute), 230
- contact_groups (shoop.campaigns.models.basket_conditions attribute), 151
- contact_groups (shoop.campaigns.models.context_conditions attribute), 166
- contact_ptr (shoop.core.models.AnonymousContact attribute), 187
- contact_ptr (shoop.core.models.CompanyContact attribute), 191
- contact_ptr (shoop.core.models.PersonContact attribute), 209
- contact_set (shoop.core.models.CustomerTaxGroup attribute), 195
- contact_set (shoop.core.models.PaymentMethod attribute), 206
- contact_set (shoop.core.models.PersonContact attribute), 209
- contact_set (shoop.core.models.ShippingMethod attribute), 228
- ContactAddressesFormPart (class in shoop.admin.modules.contacts.views.edit), 55
- ContactAddressForm (class in shoop.admin.modules.shops.views.edit), 102
- ContactAddressForm.Meta (class in shoop.admin.modules.shops.views.edit), 103
- ContactAddressFormPart (class in shoop.admin.modules.shops.views.edit), 103
- ContactBaseForm (class in shoop.admin.modules.contacts.views.edit), 54
- ContactBaseFormPart (class in shoop.admin.modules.contacts.views.edit), 54
- ContactBasketCondition (class in shoop.campaigns.models.basket_conditions), 151
- contactbasketcondition (shoop.campaigns.models.basket_conditions.BasketCondition attribute), 146
- contactbasketcondition (shoop.campaigns.models.BasketCondition attribute), 172
- ContactBasketCondition.DoesNotExist, 151
- ContactBasketCondition.MultipleObjectsReturned, 151
- contactbasketcondition_set (shoop.core.models.Contact attribute), 192
- ContactBasketConditionForm (class in shoop.campaigns.admin_module.forms), 137
- ContactBasketConditionForm.Meta (class in shoop.campaigns.admin_module.forms), 137
- ContactChoiceWidget (class in shoop.campaigns.admin_module.widgets), 46
- ContactCondition (class in shoop.campaigns.models.context_conditions), 166
- contactcondition (shoop.campaigns.models.context_conditions.ContextCondition attribute), 165
- contactcondition (shoop.campaigns.models.ContextCondition attribute), 176
- ContactCondition.DoesNotExist, 166
- ContactCondition.MultipleObjectsReturned, 167
- contactcondition_set (shoop.core.models.Contact attribute), 192
- ContactConditionForm (class in shoop.campaigns.admin_module.forms), 137
- ContactConditionForm.Meta (class in shoop.campaigns.admin_module.forms), 137
- ContactDetailToolbar (class in shoop.admin.modules.contacts.views.detail), 54
- ContactDetailView (class in shoop.admin.modules.contacts.views), 56
- ContactDetailView (class in shoop.admin.modules.contacts.views.detail), 54
- ContactEditView (class in shoop.admin.modules.contacts.views), 57

[ContactEditView](#) (class in [shoop.admin.modules.contact_groups.views](#)), [52](#)
[shoop.admin.modules.contacts.views.edit](#), [55](#)
[ContactGroup](#) (class in [shoop.core.models](#)), [193](#)
[ContactGroup.DoesNotExist](#), [193](#)
[ContactGroup.MultipleObjectsReturned](#), [193](#)
[ContactGroupBaseForm](#) (class in [shoop.admin.modules.contact_groups.views](#)), [53](#)
[shoop.admin.modules.contact_groups.views.forms](#), [51](#)
[ContactGroupBaseForm.Meta](#) (class in [shoop.admin.modules.contact_groups.views](#)), [51](#)
[shoop.admin.modules.contact_groups.views.forms](#), [51](#)
[ContactGroupBaseFormPart](#) (class in [shoop.admin.modules.contact_groups.views](#)), [53](#)
[shoop.admin.modules.contact_groups.views.forms](#), [51](#)
[ContactGroupBasketCondition](#) (class in [shoop.campaigns.models.basket_conditions](#)), [150](#)
[shoop.campaigns.models.basket_conditions](#), [150](#)
[contactgroupbasketcondition](#) ([shoop.campaigns.models.basket_conditions.BasketCondition](#) attribute), [146](#)
[contactgroupbasketcondition](#) ([shoop.campaigns.models.BasketCondition](#) attribute), [172](#)
[ContactGroupBasketCondition.DoesNotExist](#), [151](#)
[ContactGroupBasketCondition.MultipleObjectsReturned](#), [151](#)
[contactgroupbasketcondition_set](#) ([shoop.core.models.ContactGroup](#) attribute), [193](#)
[ContactGroupBasketConditionForm](#) (class in [shoop.campaigns.admin_module.forms](#)), [138](#)
[ContactGroupBasketConditionForm.Meta](#) (class in [shoop.campaigns.admin_module.forms](#)), [138](#)
[ContactGroupCondition](#) (class in [shoop.campaigns.models.context_conditions](#)), [165](#)
[shoop.campaigns.models.context_conditions](#), [165](#)
[contactgroupcondition](#) ([shoop.campaigns.models.ContextCondition](#) attribute), [176](#)
[ContactGroupCondition.DoesNotExist](#), [166](#)
[ContactGroupCondition.MultipleObjectsReturned](#), [166](#)
[contactgroupcondition_set](#) ([shoop.core.models.ContactGroup](#) attribute), [193](#)
[ContactGroupConditionForm](#) (class in [shoop.campaigns.admin_module.forms](#)), [138](#)
[ContactGroupConditionForm.Meta](#) (class in [shoop.campaigns.admin_module.forms](#)), [138](#)
[ContactGroupDeleteView](#) (class in [shoop.admin.modules.orders.sections](#)), [71](#)
[shoop.admin.modules.orders.sections](#), [71](#)
[ContactGroupDeleteView](#) (class in [shoop.admin.modules.contact_groups.views](#)), [50](#)
[shoop.admin.modules.contact_groups.views.delete](#), [50](#)
[ContactGroupEditView](#) (class in [shoop.admin.modules.contact_groups.views](#)), [53](#)
[shoop.admin.modules.contact_groups.views.edit](#), [51](#)
[ContactGroupListView](#) (class in [shoop.admin.modules.contact_groups.views](#)), [53](#)
[shoop.admin.modules.contact_groups.views.list](#), [52](#)
[ContactGroupMembersForm](#) (class in [shoop.admin.modules.contact_groups.views](#)), [52](#)
[shoop.admin.modules.contact_groups.views.forms](#), [52](#)
[ContactGroupMembersFormPart](#) (class in [shoop.admin.modules.contact_groups.views](#)), [52](#)
[shoop.admin.modules.contact_groups.views.forms](#), [52](#)
[ContactGroupMembersFormSet](#) (class in [shoop.admin.modules.contact_groups.views](#)), [52](#)
[shoop.admin.modules.contact_groups.views.forms](#), [52](#)
[ContactGroupModule](#) (class in [shoop.admin.modules.contact_groups](#)), [53](#)
[ContactGroupSalesRange](#) (class in [shoop.campaigns.models](#)), [175](#)
[shoop.campaigns.models.contact_group_sales_ranges](#), [164](#)
[ContactGroupSalesRange](#) (class in [shoop.campaigns.models](#)), [175](#)
[shoop.campaigns.models.contact_group_sales_ranges](#), [164](#)
[ContactGroupSalesRange.DoesNotExist](#), [164](#), [175](#)
[ContactGroupSalesRange.MultipleObjectsReturned](#), [164](#), [175](#)
[ContactListView](#) (class in [shoop.admin.modules.contacts.views](#)), [56](#)
[shoop.admin.modules.contacts.views.list](#), [55](#)
[ContactModule](#) (class in [shoop.admin.modules.contacts](#)), [57](#)
[ContactResetPasswordView](#) (class in [shoop.admin.modules.contacts.views](#)), [56](#)
[shoop.admin.modules.contacts.views.reset](#), [56](#)
[ContactResetPasswordView](#) (class in [shoop.campaigns.models.basket_conditions](#)), [152](#)
[shoop.campaigns.models.basket_conditions.ContactBasketCondition](#) attribute), [152](#)
[contacts](#) ([shoop.campaigns.models.context_conditions.ContactCondition](#) attribute), [166](#)
[shoop.campaigns.models.context_conditions.ContactCondition](#) attribute), [166](#)
[content](#) ([shoop.simple_cms.models.Page](#) attribute), [319](#)
[ContentsOrderSection](#) (class in [shoop.admin.modules.orders.sections](#)), [71](#)

62
create_order_from_state()
 (shoop.admin.modules.orders.json_order_creator.JsonOrderCreator
 method), 70
create_order_with_product() (in module
 shop.testing.factories), 331
create_payment() (shop.core.api.orders.OrderViewSet
 method), 179
create_payment() (shop.core.models.Order method),
 200
create_price() (shop.core.models.Shop method), 230
create_product() (in module shop.testing.factories), 331
create_random_address() (in module
 shop.testing.factories), 331
create_random_company() (in module
 shop.testing.factories), 332
create_random_order() (in module
 shop.testing.factories), 332
create_random_person() (in module
 shop.testing.factories), 331
create_refund() (shop.core.models.Order method), 200
create_service() (shop.core.models.ServiceProvider
 method), 227
create_shipment() (shop.core.models.Order method),
 200
create_shipment_of_all_products()
 (shop.core.models.Order method), 201
create_source_from_state() (in module
 shop.admin.modules.orders.views.edit),
 62
create_source_from_state()
 (shoop.admin.modules.orders.json_order_creator.JsonOrderCreator
 method), 70
created_by (shop.campaigns.models.BasketCampaign
 attribute), 170
created_by (shop.campaigns.models.Campaign at-
 tribute), 173
created_by (shop.campaigns.models.campaigns.BasketCampaign
 attribute), 159
created_by (shop.campaigns.models.campaigns.Campaign
 attribute), 157
created_by (shop.campaigns.models.campaigns.CatalogCampaign
 attribute), 158
created_by (shop.campaigns.models.campaigns.Coupon
 attribute), 160
created_by (shop.campaigns.models.campaigns.CouponUsage
 attribute), 159
created_by (shop.campaigns.models.CatalogCampaign
 attribute), 174
created_by (shop.campaigns.models.Coupon attribute),
 177
created_by (shop.simple cms.models.Page attribute),
 318
created_by (shop.simple_supplier.models.StockAdjustment
 attribute), 323
CreateOrUpdateView (class in shop.admin.utils.views),
 201
creator (shop.core.models.Order attribute), 201
creator (shop.core.order_creator.OrderSource attribute),
 239
creator (shop.front.models.stored_basket.StoredBasket
 attribute), 287
creator (shop.front.models.StoredBasket attribute), 287
CRITICAL (shop.notify.enums.Priority attribute), 310
cross_sell_1 (shop.core.models.Product attribute), 211,
 214
cross_sell_2 (shop.core.models.Product attribute), 211,
 214
currency (shop.admin.currencybound.CurrencyBound
 attribute), 126
currency (shop.simple_supplier.models.StockAdjustment
 attribute), 323
currency (shop.simple_supplier.models.StockCount at-
 tribute), 323
CurrencyBound (class in shop.admin.currencybound),
 126
CurrencyField (class in shop.core.fields), 184
CURRENT_DRAFT (shop.xtheme.models.SavedViewConfigStatus
 attribute), 365
CustomCarrier (class in shop.core.models), 195
customcarrier (shop.core.models.Carrier attribute), 189
CustomCarrier.DoesNotExist, 195
CustomCarrier.MultipleObjectsReturned, 195
customcarrier_ptr (shop.testing.models.CarrierWithCheckoutPhase
 attribute), 327
CustomOrderForm (class in
 shop.admin.modules.service_providers.forms),
 94
CustomOrderForm.Meta (class in
 shop.admin.modules.service_providers.forms),
 94
customer (shop.core.models.Order attribute), 201
customer (shop.core.order_creator.OrderSource at-
 tribute), 239
customer (shop.front.models.stored_basket.StoredBasket
 attribute), 287
customer (shop.front.models.StoredBasket attribute),
 287
customer_baskets (shop.core.models.Contact attribute),
 192
customer_orders (shop.core.models.Contact attribute),
 192
customer_tax_groups (shop.default_tax.models.TaxRule
 attribute), 267
CustomerEditView (class in
 shop.front.apps.customer_information.views),
 272

CustomerGroupPricingAppConfig (class in shoop.customer_group_pricing), 265	in	DashboardContentBlock (class in shoop.admin.dashboard.blocks), 42	in
CustomerGroupPricingForm (class in shoop.customer_group_pricing.admin_form_part), 264	in	DashboardMoneyBlock (class in shoop.admin.dashboard), 44	in
CustomerGroupPricingFormPart (class in shoop.customer_group_pricing.admin_form_part), 264	in	DashboardMoneyBlock (class in shoop.admin.dashboard.blocks), 43	in
CustomerGroupPricingModule (class in shoop.customer_group_pricing.module), 265	in	DashboardNumberBlock (class in shoop.admin.dashboard), 44	in
CustomerTaxGroup (class in shoop.core.models), 195		DashboardNumberBlock (class in shoop.admin.dashboard.blocks), 42	in
CustomerTaxGroup.DoesNotExist, 195		DashboardValueBlock (class in shoop.admin.dashboard), 44	
CustomerTaxGroup.MultipleObjectsReturned, 195		DashboardValueBlock (class in shoop.admin.dashboard.blocks), 42	in
CustomerTaxGroupEditView (class in shoop.admin.modules.taxes.views), 110	in	DashboardView (class in shoop.admin.views.dashboard), 123	
CustomerTaxGroupEditView (class in shoop.admin.modules.taxes.views.edit), 108	in	data (shoop.core.models.PersistentCacheEntry attribute), 208	
CustomerTaxGroupForm (class in shoop.admin.modules.taxes.views.edit), 107	in	data (shoop.front.models.stored_basket.StoredBasket attribute), 287	
CustomerTaxGroupForm.Meta (class in shoop.admin.modules.taxes.views.edit), 108	in	data (shoop.front.models.StoredBasket attribute), 288	
CustomerTaxGroupListView (class in shoop.admin.modules.taxes.views), 111	in	data (shoop.notify.models.Notification attribute), 305	
CustomerTaxGroupListView (class in shoop.admin.modules.taxes.views.list), 109	in	data (shoop.notify.models.notification.Notification attribute), 304	
CustomPaymentProcessor (class in shoop.core.models), 195		data (shoop.xtheme.models.ThemeSettings attribute), 366	
custompaymentprocessor (shoop.core.models.PaymentProcessor attribute), 207	at-	data_attribute (shoop.testing.simple_checkout_phase.TestMethodCheckout attribute), 335	
CustomPaymentProcessor.DoesNotExist, 196		data_attribute (shoop.testing.simple_checkout_phase.TestPaymentCheckout attribute), 335	
CustomPaymentProcessor.MultipleObjectsReturned, 196		data_attribute (shoop.testing.simple_checkout_phase.TestShipmentCheckout attribute), 335	
custompaymentprocessor_ptr (shoop.testing.models.PaymentWithCheckoutPhase attribute), 328		DatabaseBasketStorage (class in shoop.front.basket.storage), 281	in
CustomPaymentProcessorForm (class in shoop.admin.modules.service_providers.forms), 94	in	DATE (shoop.core.models.AttributeType attribute), 189	
CustomPaymentProcessorForm.Meta (class in shoop.admin.modules.service_providers.forms), 94	in	DateRangeFilter (class in shoop.admin.utils.picotable), 119	
		DATETIME (shoop.core.models.AttributeType attribute), 189	
		datetime() (in shoop.core.templatetags.shoop_common), 256	module
		Decimal (class in shoop.notify.typology), 312	
		DECIMAL (shoop.core.models.AttributeType attribute), 189	
D		declared_fields (shoop.addons.admin_module.views.list.AddonEnableDisable attribute), 38	
DashboardBlock (class in shoop.admin.dashboard), 44		declared_fields (shoop.addons.admin_module.views.reload.ReloadMethodForm attribute), 38	
DashboardBlock (class in shoop.admin.dashboard.blocks), 42	in	declared_fields (shoop.addons.admin_module.views.upload.AddonUploadForm attribute), 39	
DashboardChartBlock (class in shoop.admin.dashboard), 44		declared_fields (shoop.admin.forms.ShoopAdminForm attribute), 46	
DashboardChartBlock (class in shoop.admin.dashboard.blocks), 43	in	declared_fields (shoop.admin.modules.attributes.views.edit.AttributeForm attribute), 47	
DashboardContentBlock (class in shoop.admin.dashboard), 44	in		

declared_fields (shoop.admin.modules.categories.views.edit.CategoryBaseForm attribute), 48

declared_fields (shoop.admin.modules.categories.views.edit.CategoryForm attribute), 48

declared_fields (shoop.admin.modules.contact_groups.views.edit.ContactGroupForm attribute), 51

declared_fields (shoop.admin.modules.contact_groups.views.edit.ContactGroupForm attribute), 52

declared_fields (shoop.admin.modules.contacts.views.edit.AddressForm attribute), 55

declared_fields (shoop.admin.modules.manufacturers.views.edit.ManufacturerForm attribute), 59

declared_fields (shoop.admin.modules.orders.views.refund.RefundForm attribute), 64

declared_fields (shoop.admin.modules.orders.views.refund.RefundForm attribute), 64

declared_fields (shoop.admin.modules.orders.views.shipment.ShipmentForm attribute), 66

declared_fields (shoop.admin.modules.permission_groups.views.edit.PermissionGroupForm attribute), 72

declared_fields (shoop.admin.modules.product_types.views.edit.ProductTypeForm attribute), 74

declared_fields (shoop.admin.modules.products.forms.base_BaseForm attribute), 76

declared_fields (shoop.admin.modules.products.forms.base_BaseForm attribute), 76

declared_fields (shoop.admin.modules.products.forms.base_BaseForm attribute), 75

declared_fields (shoop.admin.modules.products.forms.base_BaseForm attribute), 77

declared_fields (shoop.admin.modules.products.forms.base_BaseForm attribute), 77

declared_fields (shoop.admin.modules.products.forms.base_BaseForm attribute), 76

declared_fields (shoop.admin.modules.products.forms.BaseProductForm attribute), 79

declared_fields (shoop.admin.modules.products.forms.package_form.PackageForm attribute), 78

declared_fields (shoop.admin.modules.products.forms.PackageForm attribute), 80

declared_fields (shoop.admin.modules.products.forms.ProductAttributeForm attribute), 80

declared_fields (shoop.admin.modules.products.forms.ProductBaseForm attribute), 80

declared_fields (shoop.admin.modules.products.forms.ProductBaseForm attribute), 80

declared_fields (shoop.admin.modules.products.forms.ProductChangeMediaForm attribute), 80

declared_fields (shoop.admin.modules.products.forms.ProductMediaForm attribute), 81

declared_fields (shoop.admin.modules.products.forms.ShopProductForm attribute), 81

declared_fields (shoop.admin.modules.products.forms.simple_variation.SimpleVariationForm attribute), 78

declared_fields (shoop.admin.modules.products.forms.SimpleVariationForm attribute), 82

declared_fields (shoop.admin.modules.products.forms.variable_variation_Form attribute), 79

declared_fields (shoop.admin.modules.products.forms.variable_variation_Form attribute), 79

declared_fields (shoop.admin.modules.products.forms.variable_variation_Form attribute), 79

declared_fields (shoop.admin.modules.products.forms.VariableVariationForm attribute), 82

declared_fields (shoop.admin.modules.products.forms.VariationVariablesForm attribute), 82

declared_fields (shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSellForm attribute), 84

declared_fields (shoop.admin.modules.products.views.edit_media.ProductMediaForm attribute), 85

declared_fields (shoop.admin.modules.sales_units.views.edit.SalesUnitForm attribute), 91

declared_fields (shoop.admin.modules.service_providers.forms.CustomCarrierForm attribute), 94

declared_fields (shoop.admin.modules.service_providers.forms.CustomPaymentForm attribute), 94

declared_fields (shoop.admin.modules.services.forms.AlwaysChangedModelForm attribute), 97

declared_fields (shoop.admin.modules.services.forms.BaseMethodForm attribute), 97

declared_fields (shoop.admin.modules.services.forms.FixedCostBehaviorForm attribute), 98

declared_fields (shoop.admin.modules.services.forms.GroupAvailabilityBehaviorForm attribute), 100

declared_fields (shoop.admin.modules.services.forms.PaymentMethodForm attribute), 98

declared_fields (shoop.admin.modules.services.forms.RoundingBehaviorForm attribute), 101

declared_fields (shoop.admin.modules.services.forms.ShippingMethodForm attribute), 98

declared_fields (shoop.admin.modules.services.forms.StaffOnlyBehaviorForm attribute), 100

declared_fields (shoop.admin.modules.services.forms.WaivingCostBehaviorForm attribute), 99

declared_fields (shoop.admin.modules.services.forms.WeightLimitsBehaviorForm attribute), 99

declared_fields (shoop.admin.modules.shops.views.edit.ContactAddressForm attribute), 103

declared_fields (shoop.admin.modules.shops.views.edit.ShopBaseForm attribute), 102

declared_fields (shoop.admin.modules.suppliers.views.edit.SupplierForm attribute), 105

declared_fields (shoop.admin.modules.taxes.views.edit.CustomerTaxGroupForm attribute), 108

declared_fields (shoop.admin.modules.taxes.views.edit.TaxClassForm attribute), 108

declared_fields (shoop.admin.modules.taxes.views.edit.TaxForm attribute), 107

declared_fields (shoop.admin.modules.users.views.detail.BaseUserForm attribute), 111

declared_fields (shoop.admin.modules.users.views.password.PasswordCharacterForm attribute), 112

declared_fields (shoop.admin.modules.permissions.PermissionForm attribute), 113

[declared_fields \(shoop.xtheme.plugins.social_media_links.SocialMediaLinksPluginForm attribute\), 355](#)
[declared_fields \(shoop.xtheme.views.forms.LayoutCellGeneratorForm attribute\), 359](#)
[decode\(\) \(shoop.core.fields.tagged_json.TagRegistry method\), 183](#)
[decode_enum\(\) \(in module shoop.core.fields.tagged_json\), 183](#)
[decompress\(\) \(shoop.xtheme.plugins.widgets.TranslatableFieldWidget method\), 356](#)
[deconstruct\(\) \(shoop.core.fields.InternalIdentifierField method\), 184](#)
[deconstruct\(\) \(shoop.core.fields.MeasurementField method\), 185](#)
[default\(\) \(shoop.api.encoders.ExtJSONEncoder method\), 131](#)
[default\(\) \(shoop.core.fields.tagged_json.TaggedJSONEncoder method\), 184](#)
[default\(\) \(shoop.utils.serialization.ExtendedJSONEncoder method\), 349](#)
[default_billing_address \(shoop.core.models.Contact attribute\), 192](#)
[default_by_identifier\(\) \(in module shoop.testing.factories\), 330](#)
[default_contact_group_identifier \(shoop.core.models.AnonymousContact attribute\), 187](#)
[default_contact_group_identifier \(shoop.core.models.CompanyContact attribute\), 191](#)
[default_contact_group_identifier \(shoop.core.models.Contact attribute\), 192](#)
[default_contact_group_identifier \(shoop.core.models.PersonContact attribute\), 209](#)
[default_contact_group_name \(shoop.core.models.AnonymousContact attribute\), 187](#)
[default_contact_group_name \(shoop.core.models.CompanyContact attribute\), 191](#)
[default_contact_group_name \(shoop.core.models.Contact attribute\), 192](#)
[default_contact_group_name \(shoop.core.models.PersonContact attribute\), 209](#)
[default_module_spec \(shoop.core.models.Supplier attribute\), 233](#)
[default_module_spec \(shoop.core.modules.interface.ModuleInterface attribute\), 237](#)
[default_module_spec \(shoop.core.modules.ModuleInterface attribute\), 238](#)
[default_payment_method \(shoop.core.models.Contact attribute\), 192](#)
[default_plugin_id \(shoop.plugins.models.ShopProduct attribute\), 231](#)
[default_plugin_value \(shoop.testing.factories.ShopProductFactory attribute\), 330](#)
[default_settings_module \(shoop.apps.AppConfig attribute\), 134](#)
[default_shipping_address \(shoop.core.models.Contact attribute\), 192](#)
[default_shipping_method \(shoop.core.models.Contact attribute\), 192](#)
[default_size \(shoop.admin.dashboard.blocks.DashboardBlock attribute\), 42](#)
[default_size \(shoop.admin.dashboard.blocks.DashboardChartBlock attribute\), 43](#)
[default_size \(shoop.admin.dashboard.blocks.DashboardValueBlock attribute\), 42](#)
[default_size \(shoop.admin.dashboard.DashboardBlock attribute\), 44](#)
[default_size \(shoop.admin.dashboard.DashboardChartBlock attribute\), 44](#)
[default_size \(shoop.admin.dashboard.DashboardValueBlock attribute\), 44](#)
[default_tax_group_getter \(shoop.core.models.Contact attribute\), 192](#)
[default_tax_group_getter\(\) \(shoop.core.models.CompanyContact method\), 191](#)
[default_tax_group_getter\(\) \(shoop.core.models.PersonContact method\), 209](#)
[DefaultBasketView \(class in shoop.front.views.basket\), 291](#)
[DefaultPricingModule \(class in shoop.core.pricing.default_pricing\), 242](#)
[DefaultTaxModule \(class in shoop.default_tax.module\), 267](#)
[DEFERRED \(shoop.core.models.PaymentStatus attribute\), 208](#)
[define_log_model\(\) \(in module shoop.utils.analog\), 338](#)
[delete\(\) \(shoop.core.models.AnonymousContact method\), 187](#)
[delete\(\) \(shoop.core.models.Carrier method\), 189](#)
[delete\(\) \(shoop.core.models.ContactGroup method\), 193](#)
[delete\(\) \(shoop.core.models.Order method\), 201](#)
[delete\(\) \(shoop.core.models.PaymentProcessor method\), 207](#)
[delete\(\) \(shoop.core.models.Product method\), 211, 214](#)
[delete\(\) \(shoop.front.basket.objects.BaseBasket method\), 279](#)
[delete\(\) \(shoop.front.basket.storage.BasketStorage method\), 280](#)
[delete\(\) \(shoop.front.basket.storage.DatabaseBasketStorage method\), 281](#)
[delete\(\) \(shoop.front.basket.storage.DirectSessionBasketStorage method\), 281](#)

- method), 281
- delete_cell() (shoop.xtheme.layout.Layout method), 364
- delete_folder() (in module shoop.admin.modules.media.utils), 60
- delete_line() (shoop.front.basket.objects.BaseBasket method), 279
- delete_line() (shoop.front.basket.update_methods.BasketUpdateMethod method), 281
- delete_row() (shoop.xtheme.layout.Layout method), 364
- DELETED (shoop.core.models.CategoryStatus attribute), 191
- DELETION (shoop.utils.analog.LogEntryKind attribute), 338
- deletion_label (shoop.admin.modules.products.forms.package_child_form_set.FixedCostBehaviorComponent attribute), 78
- deletion_label (shoop.admin.modules.products.forms.PackageChildFormSet.shoop.core.models.PaymentMethod attribute), 80
- deletion_label (shoop.admin.modules.products.forms.parent_form_set.Product attribute), 78
- deletion_label (shoop.admin.modules.products.forms.simple_variation_form_set.models.Service attribute), 79
- deletion_label (shoop.admin.modules.products.forms.SimpleVariationChildFormSet attribute), 82
- DemoModule (class in shoop.admin.modules.demo), 58
- derive_model_url() (in module shoop.admin.utils.urls), 121
- description (shoop.campaigns.models.basket_conditions.BasketMaxTotalAmountCondition attribute), 149
- description (shoop.campaigns.models.basket_conditions.BasketMaxTotalProductAffinityCondition attribute), 148
- description (shoop.campaigns.models.basket_conditions.BasketTotalAmountCondition attribute), 147
- description (shoop.campaigns.models.basket_conditions.BasketTotalProductAmountCondition attribute), 146
- description (shoop.campaigns.models.basket_conditions.ContactBasketCondition attribute), 152
- description (shoop.campaigns.models.basket_conditions.ContactGroupBasketCondition attribute), 151
- description (shoop.campaigns.models.basket_conditions.ProductsInBasketCondition attribute), 150
- description (shoop.campaigns.models.basket_effects.BasketDiscountAmount attribute), 153
- description (shoop.campaigns.models.basket_effects.BasketDiscountPercentage attribute), 154
- description (shoop.campaigns.models.basket_line_effects.DiscountFromProduct attribute), 156
- description (shoop.campaigns.models.basket_line_effects.FreeShipping attribute), 156
- description (shoop.campaigns.models.catalog_filters.CategoryFilter attribute), 164
- description (shoop.campaigns.models.catalog_filters.ProductFilter attribute), 163
- description (shoop.campaigns.models.catalog_filters.ProductTypeFilter attribute), 162
- description (shoop.campaigns.models.context_conditions.ContactCondition attribute), 166
- description (shoop.campaigns.models.context_conditions.ContactGroupCondition attribute), 166
- description (shoop.campaigns.models.context_conditions.ContextCondition attribute), 165
- description (shoop.campaigns.models.ContextCondition attribute), 176
- description (shoop.campaigns.models.product_effects.ProductDiscountAmount attribute), 168
- description (shoop.campaigns.models.product_effects.ProductDiscountPercentage attribute), 169
- description (shoop.core.models.Category attribute), 190
- description (shoop.core.models.FixedCostBehaviorComponent attribute), 196
- description (shoop.core.models.PaymentMethod attribute), 206
- description (shoop.core.models.Product attribute), 211
- description (shoop.core.models.SimpleVariationChildFormSet attribute), 219
- description (shoop.core.models.Service attribute), 224
- description (shoop.core.models.ShippingMethod attribute), 228
- description (shoop.core.models.WaivingCostBehaviorComponent attribute), 236
- description (shoop.campaigns.models.WeightBasedPriceRange attribute), 236
- description (shoop.notify.conditions.Empty attribute), 307
- description (shoop.notify.conditions.NonEmpty attribute), 308
- description (shoop.notify.conditions.simple.Empty attribute), 308
- description (shoop.notify.conditions.simple.NonEmpty attribute), 308
- DevServerReloadMethod (class in shoop.core.reloaders), 41
- DictStoredBasket (class in shoop.front.basket.storage), 280
- DirectSessionBasketStorage (class in shoop.front.basket.storage), 281
- disable_protected_fields() (shoop.core.utils.form_mixins.ProtectedFieldsMixin method), 257
- DISABLED (shoop.core.models.SavedAddressStatus attribute), 224
- DISABLED (shoop.core.models.ShopStatus attribute), 232
- DISCOUNT (shoop.core.models.OrderLineType attribute), 204
- display_amount (shoop.core.models.OrderLine attribute), 203

`discount_amount` (shoop.core.order_creator.SourceLine attribute), 242

`discount_amount` (shoop.core.pricing.Priceful attribute), 246

`discount_effects` (shoop.campaigns.models.BasketCampaign attribute), 170

`discount_effects` (shoop.campaigns.models.campaigns.BasketCampaign attribute), 159

`discount_percentage` (shoop.core.pricing.Priceful attribute), 246

`discount_price()` (shoop.campaigns.modules.CatalogCampaignModule method), 178

`discount_price()` (shoop.core.pricing.DiscountModule method), 243

`discount_prices()` (shoop.core.pricing.DiscountModule method), 243

`discount_rate` (shoop.core.pricing.Priceful attribute), 246

`discounted_unit_price` (shoop.core.pricing.Priceful attribute), 247

`DiscountFromProduct` (class in shoop.campaigns.models.basket_line_effects), 156

`discountfromproduct` (shoop.campaigns.models.basket_line_effects attribute), 155

`discountfromproduct` (shoop.campaigns.models.BasketLineEffects attribute), 170

`DiscountFromProduct.DoesNotExist`, 156

`DiscountFromProduct.MultipleObjectsReturned`, 156

`discountfromproduct_set` (shoop.core.models.Product attribute), 211, 215

`DiscountFromProductForm` (class in shoop.campaigns.admin_module.forms), 138

`DiscountFromProductForm.Meta` (class in shoop.campaigns.admin_module.forms), 138

`DiscountModule` (class in shoop.core.pricing), 243

`discover()` (in module shoop.admin.module_registry), 127

`dispatch()` (shoop.admin.modules.orders.views.edit.OrderEditView method), 62

`dispatch()` (shoop.admin.modules.orders.views.OrderEditView method), 67

`dispatch()` (shoop.admin.modules.products.views.edit_package.ProductPackageView method), 87

`dispatch()` (shoop.admin.modules.products.views.edit_parent.ProductParentView method), 87

`dispatch()` (shoop.admin.modules.products.views.edit_variation.ProductVariationView method), 88

`dispatch()` (shoop.admin.modules.products.views.ProductPackageView method), 90

`dispatch()` (shoop.admin.modules.products.views.ProductVariationView method), 90

`dispatch()` (shoop.admin.modules.users.views.detail.UserDetailView method), 112

`dispatch()` (shoop.admin.modules.users.views.UserDetailView method), 114

`dispatch()` (shoop.front.apps.auth.views.LogoutView method), 270

`dispatch()` (shoop.front.apps.auth.views.RecoverPasswordConfirmView method), 271

`dispatch()` (shoop.front.apps.registration.views.RegistrationView method), 274

`dispatch()` (shoop.front.apps.simple_search.views.SearchView method), 277

`dispatch()` (shoop.front.views.basket.BasketView method), 291

`dispatch()` (shoop.front.views.checkout.BaseCheckoutView method), 292

`dispatch()` (shoop.front.views.payment.ProcessPaymentView method), 293

`dispatch()` (shoop.notify.admin_module.views.editor.ScriptAPI method), 299

`dispatch()` (shoop.xtheme.views.editor.EditorView method), 358

`dispatch_add_cell()` (shoop.xtheme.views.editor.EditorView method), 358

`dispatch_change_plugin()` (shoop.xtheme.views.editor.EditorView method), 358

`dispatch_command()` (shoop.admin.modules.orders.views.edit.OrderEditView method), 62

`dispatch_command()` (shoop.admin.modules.orders.views.OrderEditView method), 67

`dispatch_command()` (shoop.admin.modules.products.views.edit_package.ProductPackageView method), 87

`dispatch_command()` (shoop.admin.modules.products.views.edit_parent.ProductParentView method), 87

`dispatch_command()` (shoop.admin.modules.products.views.edit_variation.ProductVariationView method), 88

`dispatch_command()` (shoop.admin.modules.products.views.ProductPackageView method), 90

`dispatch_command()` (shoop.admin.modules.products.views.ProductVariationView method), 90

`dispatch_del_cell()` (shoop.xtheme.views.editor.EditorView method), 358

`dispatch_del_row()` (shoop.xtheme.views.editor.EditorView method), 358

`dispatch_publish()` (shoop.xtheme.views.editor.EditorView method), 358

`dispatch_revert()` (shoop.xtheme.views.editor.EditorView method), 358

`DOCUMENTATION` (shoop.core.models.ProductMediaKind attribute), 220

`draft` (shoop.xtheme.models.SavedViewConfig attribute), 366

`draw_circle()` (shoop.testing.image_generator.BaseImageGenerator

- method), 332
- draw_rectangle() (shoop.testing.image_generator.BaseImageGenerator attribute), 292
- method), 332
- DropdownActionButton (class in shoop.admin.toolbar), 129
- DropdownDivider (class in shoop.admin.toolbar), 129
- DropdownHeader (class in shoop.admin.toolbar), 129
- DropdownItem (class in shoop.admin.toolbar), 129
- ## E
- easy_thumbnails_thumbnailer (shoop.core.models.ProductMedia attribute), 219
- EDIT (shoop.utils.analog.LogEntryKind attribute), 338
- editor_form_class (shoop.simple_cms.plugins.PageLinksPlugin attribute), 320
- editor_form_class (shoop.xtheme.Plugin attribute), 373
- editor_form_class (shoop.xtheme.plugins.category_links.CategoryLinksPlugin attribute), 353
- editor_form_class (shoop.xtheme.plugins.social_media_links.SocialMediaLinksPlugin attribute), 355
- EditorView (class in shoop.xtheme.views.editor), 358
- EditProductToolBar (class in shoop.admin.modules.products.views.toolbars), 88
- EditScriptContentView (class in shoop.notify.admin_module.views), 300
- EditScriptContentView (class in shoop.notify.admin_module.views.editor), 299
- effective_title (shoop.core.models.ProductMedia attribute), 219
- effects (shoop.campaigns.admin_module.views.BasketCampaignEditView attribute), 141
- effects (shoop.campaigns.admin_module.views.CatalogCampaignEditView attribute), 140
- effects (shoop.campaigns.models.campaigns.CatalogCampaign attribute), 158
- effects (shoop.campaigns.models.CatalogCampaign attribute), 174
- EffectsFormset (class in shoop.campaigns.admin_module.form_sets), 144
- Email (class in shoop.notify.typology), 312
- email (shoop.testing.factories.CompanyFactory attribute), 329
- email (shoop.testing.factories.UserFactory attribute), 329
- EMAIL (shoop.utils.analog.LogEntryKind attribute), 338
- email_template_name (shoop.front.apps.auth.forms.RecoverPasswordForm attribute), 269
- EmailAuthenticationForm (class in shoop.front.apps.auth.forms), 269
- Empty (class in shoop.notify.conditions), 303
- Empty (class in shoop.notify.conditions.simple), 302
- empty_phase_spec (shoop.front.views.checkout.BaseCheckoutView attribute), 292
- EmptyPhase (class in shoop.front.checkout.empty), 283
- ENABLED (shoop.core.models.SavedAddressStatus attribute), 224
- ENABLED (shoop.core.models.ShopStatus attribute), 232
- enabled (shoop.notify.script.Step attribute), 310
- encode() (shoop.core.fields.tagged_json.TagRegistry method), 183
- encode_address() (in module shoop.admin.modules.orders.views.edit), 62
- encode_enum() (in module shoop.core.fields.tagged_json), 183
- encode_line() (in module shoop.admin.modules.orders.views.edit), 62
- encode_method() (in module shoop.admin.modules.orders.views.edit), 62
- encode_shop() (in module shoop.admin.modules.orders.views.edit), 62
- engine (shoop.xtheme.TemplatedPlugin attribute), 375
- Enum (class in shoop.notify.typology), 313
- enum_class (shoop.notify.typology.Enum attribute), 313
- ERROR (shoop.utils.analog.LogEntryKind attribute), 338
- error_messages (shoop.admin.modules.users.views.password.PasswordChangeForm attribute), 112
- error_messages (shoop.front.apps.auth.forms.EmailAuthenticationForm attribute), 269
- errors (shoop.admin.modules.orders.json_order_creator.JsonOrderCreator attribute), 70
- errors (shoop.utils.form_group.FormGroup attribute), 341
- Event (class in shoop.notify), 315
- Event (class in shoop.notify.base), 307
- event_class (shoop.notify.models.Script attribute), 306
- event_class (shoop.notify.models.script.Script attribute), 305
- ExceptionalResponse, 339
- ExceptionMiddleware (class in shoop.core.middleware), 260
- exclude (shoop.admin.modules.attributes.views.edit.AttributeForm.Meta attribute), 47
- exclude (shoop.admin.modules.manufacturers.views.edit.ManufacturerForm attribute), 59
- exclude (shoop.admin.modules.permission_groups.views.edit.PermissionGroupForm attribute), 72
- exclude (shoop.admin.modules.product_types.views.edit.ProductTypeForm attribute), 74
- exclude (shoop.admin.modules.sales_units.views.edit.SalesUnitForm.Meta attribute), 91

exclude (shoop.admin.modules.service_providers.forms.CustomCarrierForm.Meta attribute), 94
 exclude (shoop.admin.modules.service_providers.forms.CustomPaymentProcessorForm.Meta attribute), 94
 exclude (shoop.admin.modules.services.forms.BaseMethodForm.Meta attribute), 97
 exclude (shoop.admin.modules.services.forms.FixedCostBehaviorComponentForm.Meta attribute), 98
 exclude (shoop.admin.modules.services.forms.GroupAvailabilityBehaviorComponentForm.Meta attribute), 100
 exclude (shoop.admin.modules.services.forms.RoundingBehaviorComponentForm.Meta attribute), 101
 exclude (shoop.admin.modules.services.forms.StaffOnlyBehaviorComponentForm.Meta attribute), 100
 exclude (shoop.admin.modules.services.forms.WaivingCostBehaviorComponentForm.Meta attribute), 99
 exclude (shoop.admin.modules.services.forms.WeightLimitsBehaviorComponentForm.Meta attribute), 99
 exclude (shoop.admin.modules.shops.views.edit.ShopBaseForm.Meta attribute), 102
 exclude (shoop.admin.modules.suppliers.views.edit.SupplierForm.Meta attribute), 105
 exclude (shoop.campaigns.admin_module.forms.CatalogCampaignForm.Meta attribute), 137
 exclude (shoop.campaigns.admin_module.forms.ProductDiscountAmountForm.Meta attribute), 139
 exclude (shoop.campaigns.admin_module.forms.ProductDiscountPercentageForm.Meta attribute), 139
 exclude (shoop.testing.service_forms.CarrierWithCheckoutForm.Meta attribute), 334
 exclude (shoop.testing.service_forms.PaymentWithCheckoutForm.Meta attribute), 333
 exclude (shoop.testing.service_forms.PseudoPaymentProcessorForm.Meta attribute), 333
 execute() (shoop.addons.reloader.DevServerReloadMethod method), 41
 execute() (shoop.addons.reloader.GunicornReloadMethod method), 42
 execute() (shoop.addons.reloader.ModWSGIReloadMethod method), 42
 execute() (shoop.addons.reloader.ReloadMethod method), 41
 execute() (shoop.addons.reloader.UwsgiReloadMethod method), 41
 execute() (shoop.notify.Action method), 313
 execute() (shoop.notify.actions.AddNotification method), 298
 execute() (shoop.notify.actions.AddOrderLogEntry method), 298
 execute() (shoop.notify.actions.debug.SetDebugFlag method), 297
 execute() (shoop.notify.actions.email.SendEmail method), 297
 execute() (shoop.notify.actions.notification.AddNotification method), 297
 execute() (shoop.notify.actions.order.AddOrderLogEntry method), 298
 execute() (shoop.notify.actions.SendEmail method), 298
 execute() (shoop.notify.actions.SetDebugFlag method), 298
 execute() (shoop.notify.models.Script method), 306
 execute() (shoop.notify.script.Step method), 310
 execute() (shoop.models.campaigns.Coupon attribute), 160
 execute() (shoop.models.Coupon attribute), 177
 ExpensiveSwedenBehaviorComponent (class in shoop.testing.models), 327
 ExpensiveSwedenBehaviorComponent.DoesNotExist, 327
 ExpensiveSwedenBehaviorComponent.MultipleObjectsReturned, 327
 ExtendedJSONEncoder (class in shoop.core.models.SupplierType attribute), 348
 EXTERNAL (shoop.core.models.SupplierType attribute), 348
 ExtJSONEncoder (class in shoop.api.encoders), 131
 BaseForm.Meta (shoop.admin.modules.products.forms.base_forms.BaseProductMediaForm attribute), 77
 BaseForm.Meta (shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSellForm attribute), 85
 BaseForm.Meta (shoop.admin.modules.products.views.edit_media.ProductMediaForm attribute), 86
 extra (shoop.admin.modules.services.behavior_form_part.BehaviorFormSet attribute), 96
 extra (shoop.campaigns.admin_module.form_sets.BaseFormset attribute), 144
 extra (shoop.core.models.OrderLogEntry attribute), 204
 extra (shoop.utils.analog.BaseLogEntry attribute), 338
 extra_data (shoop.core.models.Order attribute), 201
 extra_data (shoop.core.models.OrderLine attribute), 203
 extra_js (shoop.admin.base.OrderSection attribute), 126
 extra_js (shoop.admin.modules.orders.sections.LogEntriesOrderSection attribute), 71
 extra_kwargs (shoop.core.api.products.ShopProductSerializer.Meta attribute), 180
 extra_view_dispatch() (in module shoop.xtheme.views.extra), 358
 extract_form_fields() (in module shoop.testing.soup_utils), 336
 extract_inner_value() (in module shoop.utils.objects), 345
 extract_jinja() (in module shoop.core.management.commands.makemessages), 345

186

extract_messages() (in module shoop.utils.excs), 339

F

FALLBACK_LANGUAGE_CODE (in module shoop.xtheme.plugins.consts), 353

FEMALE (shoop.core.models.Gender attribute), 197

field() (shoop.admin.templatetags.shoop_admin.Bootstrap3Namespace method), 117

fields (shoop.admin.form_part.FormPartsViewMixin attribute), 127

fields (shoop.admin.modules.categories.views.edit.CategoryBaseForm.Meta attribute), 48

fields (shoop.admin.modules.contact_groups.views.forms.ContactGroupBaseForm.Meta attribute), 51

fields (shoop.admin.modules.contacts.views.edit.AddressForm.Meta attribute), 55

fields (shoop.admin.modules.orders.views.edit.OrderEditView attribute), 62

fields (shoop.admin.modules.orders.views.OrderEditView attribute), 67

fields (shoop.admin.modules.products.forms.base_forms.BaseProductMediaForm.Meta attribute), 76

fields (shoop.admin.modules.products.forms.base_forms.ProductBaseForm.Meta attribute), 75

fields (shoop.admin.modules.products.forms.base_forms.ShopProductForm.Meta attribute), 76

fields (shoop.admin.modules.products.forms.BaseProductMediaForm.Meta attribute), 79

fields (shoop.admin.modules.products.forms.ProductBaseForm.Meta attribute), 80

fields (shoop.admin.modules.products.forms.ShopProductForm.Meta attribute), 81

fields (shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSellForm.Meta attribute), 84

fields (shoop.admin.modules.products.views.edit_media.ProductMediaForm.Meta attribute), 85

fields (shoop.admin.modules.services.forms.PaymentMethodForm.Meta attribute), 98

fields (shoop.admin.modules.services.forms.ShippingMethodForm.Meta attribute), 98

fields (shoop.admin.modules.shops.views.edit.ContactAddressForm.Meta attribute), 103

fields (shoop.admin.modules.taxes.views.edit.CustomerTaxGroupForm.Meta attribute), 108

fields (shoop.admin.modules.taxes.views.edit.TaxClassForm.Meta attribute), 108

fields (shoop.admin.modules.taxes.views.edit.TaxForm.Meta attribute), 107

fields (shoop.admin.modules.users.views.detail.UserDetailView attribute), 112

fields (shoop.admin.modules.users.views.UserDetailView attribute), 114

fields (shoop.campaigns.admin_module.form_parts.SalesRangesForm.Meta attribute), 142

fields (shoop.campaigns.admin_module.forms.CouponForm.Meta attribute), 138

fields (shoop.core.api.orders.PaymentSerializer.Meta attribute), 179

fields (shoop.default_tax.admin_module.views.TaxRuleForm.Meta attribute), 266

fields (shoop.front.apps.customer_information.forms.AddressForm.Meta attribute), 272

fields (shoop.front.apps.customer_information.forms.CompanyContactForm.Meta attribute), 272

fields (shoop.front.apps.customer_information.forms.PersonContactForm.Meta attribute), 271

fields (shoop.front.checkout.addresses.AddressForm.Meta attribute), 282

fields (shoop.front.checkout.addresses.CompanyForm.Meta attribute), 282

fields (shoop.notify.admin_module.forms.ScriptForm.Meta attribute), 301

fields (shoop.simple_cms.admin_module.views.PageForm.Meta attribute), 317

fields (shoop.simple_cms.plugins.PageLinksPlugin attribute), 320

fields (shoop.themes.classic_gray.ClassicGrayTheme attribute), 337

fields (shoop.themes.classic_gray.plugins.ProductCrossSellsPlugin attribute), 337

fields (shoop.themes.classic_gray.plugins.ProductHighlightPlugin attribute), 337

fields (shoop.xtheme.forms.GenericThemeForm.Meta attribute), 361

fields (shoop.xtheme.Plugin attribute), 373

fields (shoop.xtheme.plugins.category_links.CategoryLinksPlugin attribute), 353

fields (shoop.xtheme.plugins.image.ImagePlugin attribute), 354

fields (shoop.xtheme.plugins.snippets.SnippetsPlugin attribute), 355

fields (shoop.xtheme.plugins.social_media_links.SocialMediaLinksPlugin attribute), 355

fields (shoop.xtheme.plugins.text.TextPlugin attribute), 356

fields (shoop.xtheme.Theme attribute), 375

FIELDS_BY_MODEL_NAME

(shoop.admin.modules.contacts.views.edit.ContactBaseForm attribute), 54

file (shoop.core.models.ProductMedia attribute), 219

filer_file_from_upload() (in module shoop.utils.filer), 340

filer_folder_from_path() (in module shoop.utils.filer), 340

filer_image_from_data() (in module shoop.utils.filer), 341

filer_image_from_upload() (in module shoop.utils.filer),

- 340
- fill_model_instance() (in module shoop.utils.forms), 342
- Filter (class in shoop.admin.utils.picotable), 118
- filter (shoop.admin.forms.widgets.BasePopupChoiceWidget attribute), 45
- filter (shoop.admin.forms.widgets.ImageChoiceWidget attribute), 45
- filter (shoop.admin.forms.widgets.PackageProductChoiceWidget attribute), 46
- filter (shoop.admin.forms.widgets.PersonContactChoiceWidget attribute), 46
- filter_form_field_choices() (in module shoop.admin.utils.forms), 117
- filter_key (shoop.campaigns.admin_module.views.CatalogCampaignEditView attribute), 140
- filter_queryset() (shoop.admin.utils.picotable.ChoicesFilter method), 118
- filter_queryset() (shoop.admin.utils.picotable.ColumnFilter method), 120
- filter_queryset() (shoop.admin.utils.picotable.DateRangeFilter method), 119
- filter_queryset() (shoop.admin.utils.picotable.Filter method), 118
- filter_queryset() (shoop.admin.utils.picotable.MultiFieldTextFilter method), 119
- filter_queryset() (shoop.admin.utils.picotable.RangeFilter method), 119
- filter_queryset() (shoop.admin.utils.picotable.TextFilter method), 119
- filter_queryset() (shoop.campaigns.models.catalog_filters.CatalogFilter method), 161
- filter_queryset() (shoop.campaigns.models.catalog_filters.CategoryFilter method), 163
- filter_queryset() (shoop.campaigns.models.catalog_filters.ProductFilter method), 163
- filter_queryset() (shoop.campaigns.models.catalog_filters.ProductTypeFilter method), 162
- filter_queryset() (shoop.campaigns.models.CatalogFilter method), 175
- filters (shoop.campaigns.models.campaigns.CatalogCampaign attribute), 158
- filters (shoop.campaigns.models.CatalogCampaign attribute), 174
- final (shoop.front.checkout.CheckoutPhaseViewMixin attribute), 285
- final (shoop.front.checkout.single_page.SingleCheckoutPhase attribute), 285
- finalize() (shoop.front.basket.objects.BaseBasket method), 279
- finalize() (shoop.front.basket.storage.BasketStorage method), 280
- finalize() (shoop.front.basket.storage.DatabaseBasketStorage method), 281
- finalize_installation_for_enabled_apps() (in module shoop.addons.admin_module.views.reload), 38
- find_line_by_line_id() (shoop.front.basket.objects.BaseBasket method), 279
- find_lines_by_parent_line_id() (shoop.front.basket.objects.BaseBasket method), 280
- first() (in module shoop.utils.iterables), 343
- first_name (shoop.core.utils.name_mixin.NameMixin attribute), 257
- first_name (shoop.testing.factories.UserFactory attribute), 329
- FixedCostBehaviorComponent (class in shoop.core.models), 196
- FixedCostBehaviorComponent (shoop.core.models.ServiceBehaviorComponent attribute), 225
- FixedCostBehaviorComponent.DoesNotExist, 196
- FixedCostBehaviorComponent.MultipleObjectsReturned, 196
- FixedCostBehaviorComponentForm (class in shoop.admin.modules.services.forms), 98
- FixedCostBehaviorComponentForm.Meta (class in shoop.admin.modules.services.forms), 98
- filter() (in module shoop.admin.utils.forms), 117
- flatten() (in module shoop.utils.text), 349
- flatten_const_node_list() (in module shoop.xtheme.parsing), 366
- for_model() (shoop.admin.toolbar.NewActionButton class method), 128
- forget() (shoop.notify.models.notification.NotificationManager method), 304
- foreign_amount (shoop.core.models.Payment attribute), 206
- form (shoop.admin.modules.services.base_form_part.PaymentMethodBase attribute), 96
- form (shoop.admin.modules.services.base_form_part.ServiceBaseFormPart attribute), 96
- form (shoop.admin.modules.services.base_form_part.ShippingMethodBase attribute), 96
- form (shoop.campaigns.admin_module.form_parts.BasketBaseFormPart attribute), 142
- form (shoop.campaigns.admin_module.form_parts.CampaignBaseFormPart attribute), 142
- form (shoop.campaigns.admin_module.form_parts.CatalogBaseFormPart attribute), 142
- form (shoop.campaigns.admin_module.form_parts.SalesRangesFormPart attribute), 142
- form (shoop.simple_supplier.admin_module.forms.SimpleSupplierFormPart attribute), 321
- form() (shoop.admin.modules.products.forms.base_forms.BaseProductMed method), 77
- form() (shoop.admin.modules.products.views.edit_cross_sell.ProductCrossS method), 85
- form() (shoop.admin.modules.products.views.edit_media.ProductMediaFor

method), 86

form() (shoop.admin.modules.services.behavior_form_part.BehaviorFormSet attribute), 97

form() (shoop.admin.modules.product_types.views.ProductTypeEditView attribute), 75

form() (shoop.admin.template_tags.shoop_admin.Bootstrap3Form class (shoop.admin.modules.products.forms.base_forms.ProductImageForm attribute), 77

form() (shoop.admin.modules.products.forms.base_forms.ProductMediaFormSet attribute), 77

form() (shoop.campaigns.admin_module.form_sets.BaseFormset class (shoop.admin.modules.products.forms.ProductImageMediaForm attribute), 81

form() (shoop.campaigns.admin_module.form_sets.EffectsFormset class (shoop.admin.modules.products.forms.ProductImageMediaForm attribute), 81

form_class (shoop.addons.admin_module.views.AddonListView class (shoop.admin.modules.products.forms.ProductMediaFormSet attribute), 81

form_class (shoop.addons.admin_module.views.AddonUploadForm class (shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSellForm attribute), 85

form_class (shoop.addons.admin_module.views.AddonUploadForm class (shoop.admin.modules.products.views.edit_media.ProductMediaForm attribute), 86

form_class (shoop.addons.admin_module.views.list.AddonListView class (shoop.admin.modules.products.views.edit_parent.ProductParentForm attribute), 87

form_class (shoop.addons.admin_module.views.reload.ReloadForm class (shoop.admin.modules.products.views.ProductCrossSellEditView attribute), 89

form_class (shoop.addons.admin_module.views.ReloadForm class (shoop.admin.modules.products.views.ProductMediaEditView attribute), 90

form_class (shoop.addons.admin_module.views.upload.AddonUploadForm class (shoop.admin.modules.sales_units.views.edit.SalesUnitEditView attribute), 92

form_class (shoop.addons.admin_module.views.upload.AddonUploadForm class (shoop.admin.modules.sales_units.views.SalesUnitEditView attribute), 92

form_class (shoop.admin.modules.attributes.views.AttributeEditView class (shoop.admin.modules.service_providers.views.ServiceProviderForm attribute), 93

form_class (shoop.admin.modules.attributes.views.edit.AttributeEditView class (shoop.admin.modules.services.behavior_form_part.BehaviorForm attribute), 96

form_class (shoop.admin.modules.manufacturers.views.edit.ManufacturerEditView class (shoop.admin.modules.suppliers.views.edit.SupplierEditView attribute), 105

form_class (shoop.admin.modules.manufacturers.views.ManufacturerEditView class (shoop.admin.modules.suppliers.views.SupplierEditView attribute), 106

form_class (shoop.admin.modules.orders.views.OrderCreatedForm class (shoop.admin.modules.taxes.views.CustomerTaxGroupEditView attribute), 110

form_class (shoop.admin.modules.orders.views.OrderCreatedForm class (shoop.admin.modules.taxes.views.edit.CustomerTaxGroupEditView attribute), 108

form_class (shoop.admin.modules.orders.views.OrderCreatedForm class (shoop.admin.modules.taxes.views.edit.TaxClassEditView attribute), 109

form_class (shoop.admin.modules.orders.views.OrderCreatedForm class (shoop.admin.modules.taxes.views.edit.TaxEditView attribute), 108

form_class (shoop.admin.modules.orders.views.payment.OrderCreatedForm class (shoop.admin.modules.taxes.views.TaxClassEditView attribute), 110

form_class (shoop.admin.modules.orders.views.refund.OrderCreatedForm class (shoop.admin.modules.taxes.views.TaxEditView attribute), 110

form_class (shoop.admin.modules.orders.views.refund.OrderCreatedForm class (shoop.admin.modules.users.views.password.UserChangePasswordView attribute), 113

form_class (shoop.admin.modules.orders.views.shipment.OrderCreatedForm class (shoop.admin.modules.users.views.UserChangePasswordView attribute), 115

form_class (shoop.admin.modules.permission_groups.views.PermissionGroupEditView class (shoop.admin_module.form_sets.BaseFormset attribute), 144

form_class (shoop.admin.modules.permission_groups.views.PermissionGroupEditView class (shoop.campaigns.admin_module.views.CouponEditView attribute), 141

form_class (shoop.admin.modules.product_types.views.edit.ProductTypeEditView class (shoop.admin_module.views.TaxRuleEditView attribute), 141

attribute), 266
form_class (shoop.front.apps.auth.views.LoginView attribute), 270
form_class (shoop.front.apps.auth.views.RecoverPasswordConfirmView attribute), 270
form_class (shoop.front.apps.auth.views.RecoverPasswordView attribute), 270
form_class (shoop.front.apps.simple_search.views.SearchView attribute), 276
form_class (shoop.front.checkout.confirm.ConfirmPhase attribute), 283
form_class (shoop.front.checkout.methods.MethodsPhase attribute), 284
form_class (shoop.notify.admin_module.views.edit.ScriptEditView attribute), 299
form_class (shoop.notify.admin_module.views.ScriptEditView attribute), 300
form_class (shoop.simple_cms.admin_module.views.PageEditView attribute), 317
form_class (shoop.testing.admin_module.mock_view.MockView attribute), 325
form_class (shoop.testing.simple_checkout_phase.TestPaymentCheckoutPhase attribute), 335
form_class (shoop.testing.simple_checkout_phase.TestShipmentCheckoutPhase attribute), 335
form_class (shoop.xtheme.admin_module.views.ThemeConfigDetailView attribute), 352
form_class (shoop.xtheme.admin_module.views.ThemeConfigView attribute), 351
form_def_name (shoop.admin.modules.products.views.edit_variation.AttributesForm attribute), 88
form_invalid() (shoop.admin.modules.orders.views.OrderCreatePaymentView method), 68
form_invalid() (shoop.admin.modules.orders.views.OrderCreateShipmentView method), 69
form_invalid() (shoop.admin.modules.orders.views.payment.OrderCreatePaymentView method), 63
form_invalid() (shoop.admin.modules.orders.views.shipment.OrderCreateShipmentView method), 66
form_invalid() (shoop.admin.utils.views.CreateOrUpdateView method), 122
form_name (shoop.admin.modules.products.views.edit_parent.ProductEditView attribute), 87
form_part_class_provide_key (shoop.admin.form_part.FormPartsViewMixin attribute), 127
form_part_class_provide_key (shoop.admin.modules.categories.views.CategoryEditView attribute), 49
form_part_class_provide_key (shoop.admin.modules.categories.views.edit.CategoryEditView attribute), 49
form_part_class_provide_key (shoop.admin.modules.contact_groups.views.ContactGroupEditView attribute), 53
form_part_class_provide_key (shoop.admin.modules.contact_groups.views.edit.ContactGroupEditView attribute), 51
form_part_class_provide_key (shoop.admin.modules.contacts.views.ContactEditView attribute), 57
form_part_class_provide_key (shoop.admin.modules.contacts.views.edit.ContactEditView attribute), 55
form_part_class_provide_key (shoop.admin.modules.products.views.edit.ProductEditView attribute), 84
form_part_class_provide_key (shoop.admin.modules.products.views.ProductEditView attribute), 89
form_part_class_provide_key (shoop.admin.modules.shops.views.edit.ShopEditView attribute), 103
form_part_class_provide_key (shoop.admin.modules.shops.views.ShopEditView attribute), 104
form_part_classes (shoop.admin.modules.products.views.edit_package.ProductEditView attribute), 86
form_part_classes (shoop.admin.modules.products.views.edit_parent.ProductEditView attribute), 87
form_part_classes (shoop.admin.modules.products.views.edit_variation.ProductEditView attribute), 88
form_part_classes (shoop.admin.modules.products.views.ProductPackageView attribute), 90
form_part_classes (shoop.admin.modules.products.views.ProductVariationView attribute), 93
form_provide_key (shoop.admin.modules.service_providers.views.ServiceProviderView attribute), 93
form_valid() (shoop.addons.admin_module.views.AddonListView method), 39
form_valid() (shoop.addons.admin_module.views.AddonUploadConfirmView method), 39
form_valid() (shoop.addons.admin_module.views.AddonUploadView method), 40
form_valid() (shoop.addons.admin_module.views.list.AddonListView method), 38
form_valid() (shoop.addons.admin_module.views.reload.ReloadView method), 38
form_valid() (shoop.addons.admin_module.views.ReloadView method), 40
form_valid() (shoop.addons.admin_module.views.upload.AddonUploadConfirmView method), 39
form_valid() (shoop.addons.admin_module.views.upload.AddonUploadView method), 39
form_valid() (shoop.admin.form_part.FormPart method), 127
form_valid() (shoop.admin.modules.categories.views.CategoryEditView method), 49

[form_valid\(\) \(shoop.admin.modules.categories.views.edit.CategoryEditFormPart.admin.modules.products.views.edit_variation.VariationEditFormPart method\), 49](#)
[form_valid\(\) \(shoop.admin.modules.categories.views.edit.CategoryEditView.shoop.admin.modules.products.views.ProductCrossSellEditView method\), 49](#)
[form_valid\(\) \(shoop.admin.modules.contact_groups.views.ContactGroupEditView.admin.modules.products.views.ProductEditView method\), 53](#)
[form_valid\(\) \(shoop.admin.modules.contact_groups.views.edit_contact_group_edit_view.admin.modules.products.views.ProductMediaEditView method\), 51](#)
[form_valid\(\) \(shoop.admin.modules.contact_groups.views.forms.ContactGroupBaseFormPart.services.base_form_part.ServiceBaseFormPart method\), 51](#)
[form_valid\(\) \(shoop.admin.modules.contact_groups.views.forms.ContactGroupMembersFormPart.services.behavior_form_part.BehaviorFormPart method\), 52](#)
[form_valid\(\) \(shoop.admin.modules.contacts.views.ContactEditView.admin.modules.services.weight_based_pricing.WeightBasedPricingFormPart method\), 57](#)
[form_valid\(\) \(shoop.admin.modules.contacts.views.edit.ContactAddressFormPart.admin.modules.shops.views.edit.ContactAddressFormPart method\), 55](#)
[form_valid\(\) \(shoop.admin.modules.contacts.views.edit.ContactBaseFormPart.admin.modules.shops.views.edit.ShopBaseFormPart method\), 55](#)
[form_valid\(\) \(shoop.admin.modules.contacts.views.edit.ContactEditView.admin.modules.shops.views.edit.ShopEditView method\), 55](#)
[form_valid\(\) \(shoop.admin.modules.orders.views.OrderCreateFormRefundView.shoop.admin.modules.shops.views.ShopEditView method\), 68](#)
[form_valid\(\) \(shoop.admin.modules.orders.views.OrderCreatePaymentView.shoop.admin.modules.users.views.password.UserChangePasswordView method\), 68](#)
[form_valid\(\) \(shoop.admin.modules.orders.views.OrderCreateRefundView.shoop.admin.modules.users.views.permissions.UserChangePermissionsView method\), 68](#)
[form_valid\(\) \(shoop.admin.modules.orders.views.OrderCreateShipView.shoop.admin.modules.users.views.UserChangePasswordView method\), 69](#)
[form_valid\(\) \(shoop.admin.modules.orders.views.payment.OrderCreatePaymentView.admin.modules.users.views.UserChangePermissionsView method\), 63](#)
[form_valid\(\) \(shoop.admin.modules.orders.views.refund.OrderCreateRefundView.admin.modules.shops.views.ShopEditView method\), 65](#)
[form_valid\(\) \(shoop.admin.modules.orders.views.refund.OrderCreateRefundView.campaigns.admin_module.form_parts.BaseFormPart method\), 64](#)
[form_valid\(\) \(shoop.admin.modules.orders.views.shipment.OrderCreateShipmentView.campaigns.admin_module.form_parts.CampaignBaseFormPart method\), 66](#)
[form_valid\(\) \(shoop.admin.modules.products.views.edit.BaseFormPart.campaigns.admin_module.form_parts.SalesRangesFormPart method\), 83](#)
[form_valid\(\) \(shoop.admin.modules.products.views.edit.ProductAvailabilityFormPart.customer_group_pricing.admin_form_part.CustomerGroupPricingFormPart method\), 83](#)
[form_valid\(\) \(shoop.admin.modules.products.views.edit.ProductBaseFormPart.shoop.front.apps.auth.views.LoginView method\), 83](#)
[form_valid\(\) \(shoop.admin.modules.products.views.edit.ProductEditView.shoop.front.apps.auth.views.RecoverPasswordConfirmView method\), 84](#)
[form_valid\(\) \(shoop.admin.modules.products.views.edit.ShopProductFormPart.shoop.front.apps.auth.views.RecoverPasswordView method\), 83](#)
[form_valid\(\) \(shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSellEditView.customer_information.views.CompanyEditView method\), 85](#)
[form_valid\(\) \(shoop.admin.modules.products.views.edit_media.ProductMediaEditView.customer_information.views.CustomerEditView method\), 86](#)
[form_valid\(\) \(shoop.admin.modules.products.views.edit_park.ProductParkFormPart.addresses.AddressesPhaseFormPart method\), 87](#)
[form_valid\(\) \(shoop.admin.modules.products.views.edit_park.ProductParkFormPart.shoop.front.apps.checkout.confirm.ConfirmPhaseFormPart method\), 87](#)

form_valid() (shoop.front.checkout.methods.MethodsPhaseFormInstantiationAttributeError, 341	
method), 284	FormPart (class in shoop.admin.form_part), 127
form_valid() (shoop.front.checkout.single_page.SingleCheckoutFormMixin (class in shoop.admin.form_part),	
method), 285	127
form_valid() (shoop.notify.admin_module.views.edit.ScriptEditView (shoop.utils.form_group.FormGroup attribute),	
method), 299	341
form_valid() (shoop.notify.admin_module.views.ScriptEditView (shoop.admin.modules.products.views.edit.ProductImageMediaForm	
method), 300	attribute), 84
form_valid() (shoop.simple_supplier.admin_module.forms.SimpleSupplierForm (shoop.admin.modules.products.views.edit.ProductMediaFormPart	
method), 321	attribute), 83
form_valid() (shoop.testing.admin_module.mocked_view.MockedView (shoop.admin.modules.services.behavior_form_part.BehaviorComp	
method), 326	attribute), 97
form_valid() (shoop.testing.simple_checkout_phase.TestMethodCheckoutPhase (shoop.admin.modules.services.weight_based_pricing.WeightBased	
method), 335	attribute), 95
form_valid() (shoop.xtheme.admin_module.views.ThemeConfigView (shoop.campaigns.admin_module.form_parts.BaseFormPart	
method), 352	attribute), 142
form_valid_hook() (shoop.admin.modules.orders.views.shipments.ShipmentForm (shoop.campaigns.admin_module.form_parts.BasketConditionsForm	
method), 65	attribute), 143
format_abandoned_status()	formset (shoop.campaigns.admin_module.form_parts.BasketDiscountEffectsForm (class in shoop.campaigns.admin_module.form_parts), 143
(shoop.front.admin_module.carts.views.CartListView attribute), 143	
method), 268	formset (shoop.campaigns.admin_module.form_parts.BasketLineEffectsForm (class in shoop.campaigns.admin_module.form_parts), 143
format_created_date() (shoop.front.admin_module.carts.views.CartListView attribute), 143	
method), 268	formset (shoop.campaigns.admin_module.form_parts.CatalogConditionsForm (class in shoop.campaigns.admin_module.form_parts), 143
format_decimal() (shoop.core.fields.FormattedDecimalField attribute), 143	
method), 184	formset (shoop.campaigns.admin_module.form_parts.CatalogEffectsForm (class in shoop.campaigns.admin_module.form_parts), 143
format_money() (in module shoop.utils.i18n), 342	attribute), 143
format_order_date() (shoop.admin.modules.orders.views.list.OrderListView (shoop.campaigns.admin_module.form_parts.CatalogFiltersFormPa	
method), 63	attribute), 143
format_order_date() (shoop.admin.modules.orders.views.OrderListView (shoop.campaigns.admin_module.form_parts.CatalogFiltersFormPa	
method), 67	FreeProductLine (class in shoop.campaigns.models.basket_line_effects),
format_output() (shoop.xtheme.plugins.widgets.TranslatableFieldWidget, 355	
method), 356	freeproductline (shoop.campaigns.models.basket_line_effects.BasketLineEffect (class in shoop.campaigns.models.basket_line_effects), 155
format_percent() (in module shoop.utils.i18n), 342	attribute), 155
format_taxful_total_price()	freeproductline (shoop.campaigns.models.BasketLineEffect (class in shoop.campaigns.models.basket_line_effects), 170
(shoop.admin.modules.orders.views.list.OrderListView attribute), 170	
method), 63	FreeProductLine.DoesNotExist, 156
format_taxful_total_price()	FreeProductLine.MultipleObjectsReturned, 156
(shoop.admin.modules.orders.views.OrderListView attribute), 211, 215	
method), 67	FreeProductLineForm (class in shoop.campaigns.admin_module.forms), 139
format_taxful_total_price()	
(shoop.front.admin_module.carts.views.CartListView attribute), 139	
method), 268	FreeProductLineForm.Meta (class in shoop.campaigns.admin_module.forms), 139
format_updated_date() (shoop.front.admin_module.carts.views.CartListView attribute), 139	
method), 268	
FormattedDecimalField (class in shoop.core.fields), 184	
FormattedDecimalFormField (class in shoop.core.fields), 184	from_basket_and_data() (shoop.front.basket.storage.DictStoredBasket class method), 281
FormDef (class in shoop.utils.form_group), 341	from_context() (shoop.core.pricing.PriceDisplayOptions class method), 245
formfield() (shoop.core.fields.FormattedDecimalField method), 184	from_data() (shoop.core.models.ImmutableAddress class method), 198
formfield() (shoop.core.models.Attribute method), 188	from_data() (shoop.core.models.MutableAddress class method), 199
formfield_callback() (shoop.admin.modules.contacts.views.front_contacts.BasketForm (class in shoop.admin.modules.contacts.views.front_contacts), 54	
method), 54	from_data() (shoop.core.pricing.Price class method), 245
FormGroup (class in shoop.utils.form_group), 341	

- from_data() (shoop.utils.money.Money class method), 344
- from_dict() (shoop.core.order_creator.SourceLine class method), 242
- from_dict() (shoop.front.basket.storage.DictStoredBasket class method), 281
- from_email (shoop.front.apps.auth.forms.RecoverPasswordForm attribute), 269
- from_event() (shoop.notify.Context class method), 314
- from_event() (shoop.notify.script.Context class method), 310
- from_line_taxes() (shoop.core.taxing.TaxSummary class method), 254
- from_tax() (shoop.core.taxing.LineTax class method), 253
- from_variables() (shoop.notify.Context class method), 314
- from_variables() (shoop.notify.script.Context class method), 310
- from_vars() (shoop.xtheme.resources.InlineScriptResource class method), 369
- full_clean() (shoop.core.models.Order method), 201
- full_clean() (shoop.utils.form_group.FormGroup method), 341
- full_name (shoop.core.models.Contact attribute), 192
- full_name (shoop.core.utils.name_mixin.NameMixin attribute), 257
- FullRefundConfirmationForm (class in shoop.admin.modules.orders.views.refund), 64
- FULLY_PAID (shoop.core.models.PaymentStatus attribute), 208
- FULLY_SHIPPED (shoop.core.models.ShippingStatus attribute), 229
- fuzz() (shoop.testing.factories.FuzzyBoolean method), 329
- fuzz() (shoop.testing.factories.FuzzyName method), 330
- FuzzyBoolean (class in shoop.testing.factories), 329
- FuzzyMatcher (class in shoop.admin.utils.search), 120
- FuzzyName (class in shoop.testing.factories), 330
- ## G
- Gender (class in shoop.core.models), 197
- gender (shoop.core.models.PersonContact attribute), 209
- generate() (shoop.testing.image_generator.BaseImageGenerator method), 332
- generate() (shoop.testing.image_generator.ModernArtImageGenerator method), 332
- generate() (shoop.testing.image_generator.RandomImageGenerator method), 332
- generate() (shoop.testing.image_generator.RingImageGenerator method), 333
- generate_code() (shoop.campaigns.models.campaigns.Coupon class method), 160
- generate_code() (shoop.campaigns.models.Coupon class method), 177
- generate_fields() (shoop.admin.modules.contacts.views.edit.ContactBaseForm method), 54
- generate_image() (in module shoop.testing.image_generator), 333
- generate_key() (in module shoop.front.models.stored_basket), 287
- generate_multilanguage_slugs() (in module shoop.core.utils.slugs), 259
- generate_pricing() (shoop.testing.mock_population.Populator method), 333
- GENERIC_FILE (shoop.core.models.ProductMediaKind attribute), 220
- GenericPluginForm (class in shoop.xtheme.plugins.forms), 354
- GenericThemeForm (class in shoop.xtheme.forms), 361
- GenericThemeForm.Meta (class in shoop.xtheme.forms), 361
- get() (in module shoop.configuration), 377
- get() (shoop.addons.admin_module.views.reload.ReloadView method), 39
- get() (shoop.addons.admin_module.views.ReloadView method), 40
- get() (shoop.admin.modules.media.views.MediaBrowserView method), 60
- get() (shoop.admin.modules.orders.views.detail.OrderSetStatusView method), 61
- get() (shoop.admin.modules.orders.views.OrderSetStatusView method), 69
- get() (shoop.admin.modules.products.views.delete.ProductDeleteView method), 83
- get() (shoop.admin.modules.products.views.ProductDeleteView method), 89
- get() (shoop.admin.modules.system.views.telemetry.TelemetryView method), 107
- get() (shoop.admin.utils.picotable.PicotableViewMixin method), 120
- get() (shoop.admin.views.dashboard.DashboardView method), 123
- get() (shoop.admin.views.search.SearchView method), 123
- get() (shoop.admin.views.select.MultiselectAjaxView method), 124
- get() (shoop.core.cache.impl.VersionedCache method), 181
- get() (shoop.core.cache.VersionedCache method), 182
- get() (shoop.core.order_creator.SourceLine method), 242
- get() (shoop.front.views.order.OrderRequiresVerificationView method), 292
- get() (shoop.front.views.product.ProductDetailView method), 293
- get() (shoop.notify.Context method), 314
- get() (shoop.notify.script.Context method), 310

`get()` (shoop.simple_cms.views.PageView method), 320

`get()` (shoop.utils.setup.Setup method), 349

`get()` (shoop.xtheme.template_ns.XthemeNamespace method), 371

`get_active_customers_block()` (in module shoop.admin.modules.contacts.dashboard), 57

`get_activity()` (in module shoop.admin.dashboard), 44

`get_activity()` (in module shoop.admin.dashboard.utils), 43

`get_activity()` (shoop.admin.base.AdminModule method), 124

`get_activity()` (shoop.admin.modules.demo.DemoModule method), 58

`get_addon_path()` (shoop.addons.admin_module.views.AddonUploadConfirmView method), 40

`get_addon_path()` (shoop.addons.admin_module.views.upload.AddonUploadConfirmView method), 39

`get_addons_from_entry_points()` (in module shoop.addons.manager), 41

`get_address()` (in module shoop.testing.factories), 330

`get_adjustment_success_message()` (in module shoop.simple_supplier.admin_module.views), 322

`get_all_available_combinations()` (shoop.core.models.Product method), 211, 215

`get_all_manufacturers()` (in module shoop.front.template_helpers.general), 288

`get_all_package_children()` (shoop.core.models.Product method), 211, 215

`get_all_package_parents()` (shoop.core.models.Product method), 211, 215

`get_alphabetical_limits()` (shoop.utils.patterns.Pattern method), 347

`get_and_increment()` (shoop.core.models.Counter class method), 194

`get_available_attribute_queryset()` (shoop.core.models.Product method), 211, 215

`get_available_payment_methods()` (shoop.front.basket.objects.BaseBasket method), 280

`get_available_shipping_methods()` (shoop.front.basket.objects.BaseBasket method), 280

`get_available_variation_results()` (shoop.core.models.Product method), 211, 215

`get_avg_purchase_size_block()` (in module shoop.admin.modules.orders.dashboard), 69

`get_babel_locale` (in module shoop.utils.i18n), 342

`get_base_price()` (shoop.core.models.Product method), 211, 215

`get_basket()` (in module shoop.front.basket), 281

`get_basket_command_dispatcher()` (in module shoop.front.basket), 281

`get_basket_order_creator()` (in module shoop.front.basket), 281

`get_basket_view()` (in module shoop.front.basket), 281

`get_best_selling_product_info()` (in module shoop.front.utils.product_statistics), 290

`get_best_selling_products()` (in module shoop.front.template_helpers.general), 288

`get_breadcrumb_parents()` (shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSellEditView method), 85

`get_breadcrumb_parents()` (shoop.admin.modules.products.views.edit_media.ProductMediaEditView method), 86

`get_breadcrumb_parents()` (shoop.admin.modules.products.views.edit_parent.ProductParentEditView method), 87

`get_breadcrumb_parents()` (shoop.admin.modules.products.views.ProductCrossSellEditView method), 89

`get_breadcrumb_parents()` (shoop.admin.modules.products.views.ProductMediaEditView method), 90

`get_breadcrumb_parents()` (shoop.admin.modules.service_providers.views.ServiceProviderEditView method), 93

`get_browse_markup()` (shoop.admin.forms.widgets.BasePopupChoiceWidget method), 45

`get_browse_markup()` (shoop.admin.forms.widgets.ContactChoiceWidget method), 46

`get_cache_duration()` (in module shoop.core.cache.impl), 180

`get_cell()` (shoop.xtheme.layout.Layout method), 364

`get_chart()` (shoop.admin.dashboard.blocks.DashboardChartBlock method), 43

`get_chart()` (shoop.admin.dashboard.DashboardChartBlock method), 44

`get_chart()` (shoop.admin.modules.orders.dashboard.OrderValueChartDashboard method), 69

`get_cheapest_child_price()` (shoop.core.models.Product method), 211, 215

`get_cheapest_child_price_info()` (shoop.core.models.Product method), 211, 215

`get_checkout_phase()` (shoop.front.checkout.BasicServiceCheckoutPhaseProvider method), 285

`get_checkout_phase()` (shoop.front.checkout.ServiceCheckoutPhaseProvider method), 286

`get_child_price_range()` (shoop.core.models.Product method), 211, 215

`get_choices()` (shoop.core.fields.LanguageField method),

- 185
- `get_clear_markup()` (shoop.admin.forms.widgets.BasePopupChoiceWidget method), 45
- `get_cms_navigation_links()` (shoop.themes.classic_gray.ClassicGrayTheme method), 337
- `get_command_handler()` (shoop.front.basket.command_dispatcher.BasketCommandDispatcher method), 277
- `get_company_contact()` (in module shoop.core.models), 197
- `get_completed_order_status()` (in module shoop.testing.factories), 331
- `get_computed_class()` (shoop.admin.toolbar.BaseActionButton method), 128
- `get_config()` (in module shoop.admin.template_helpers.shoop_admin), 116
- `get_config()` (shoop.admin.dashboard.BarChart method), 44
- `get_config()` (shoop.admin.dashboard.charts.BarChart method), 43
- `get_config()` (shoop.admin.dashboard.charts.Chart method), 43
- `get_config()` (shoop.admin.modules.orders.views.edit.OrderEditView method), 62
- `get_config()` (shoop.admin.modules.orders.views.OrderEditView method), 67
- `get_config()` (shoop.xtheme.plugins.forms.PluginForm method), 353
- `get_config_json()` (shoop.admin.dashboard.charts.Chart method), 43
- `get_configuration_form()` (shoop.themes.classic_gray.ClassicGrayTheme method), 337
- `get_configuration_form()` (shoop.xtheme.Theme method), 375
- `get_confirmation_pdf()` (in module shoop.order_printouts.admin_module.views), 316
- `get_contact()` (shoop.admin.modules.contacts.views.ContactBaseDetailView method), 56
- `get_contact()` (shoop.admin.modules.contacts.views.reset.ContactResetPasswordView method), 56
- `get_contacts_in_sales_range()` (in module shoop.campaigns.utils), 178
- `get_context()` (shoop.core.pricing.PricingModule method), 249
- `get_context_data()` (shoop.addons.admin_module.views.AddonListView method), 40
- `get_context_data()` (shoop.addons.admin_module.views.AddonUploadConfirmationView method), 40
- `get_context_data()` (shoop.addons.admin_module.views.AddonUploadView method), 40
- `get_context_data()` (shoop.addons.admin_module.views.list.AddonListView method), 38
- `get_context_data()` (shoop.addons.admin_module.views.upload.AddonUploadView method), 39
- `get_context_data()` (shoop.addons.admin_module.views.upload.AddonUploadView method), 39
- `get_context_data()` (shoop.admin.base.OrderSection static method), 39
- `get_context_data()` (shoop.admin.modules.contact_groups.views.ContactGroupView method), 53
- `get_context_data()` (shoop.admin.modules.contact_groups.views.list.ContactGroupListView method), 52
- `get_context_data()` (shoop.admin.modules.contacts.views.ContactDetailView method), 56
- `get_context_data()` (shoop.admin.modules.contacts.views.detail.ContactDetailView method), 54
- `get_context_data()` (shoop.admin.modules.media.views.MediaBrowserView method), 60
- `get_context_data()` (shoop.admin.modules.orders.sections.ContentsOrderSection static method), 71
- `get_context_data()` (shoop.admin.modules.orders.sections.LogEntriesOrderSection static method), 71
- `get_context_data()` (shoop.admin.modules.orders.sections.PaymentOrderSection static method), 71
- `get_context_data()` (shoop.admin.modules.orders.views.detail.OrderDetailView method), 61
- `get_context_data()` (shoop.admin.modules.orders.views.edit.OrderEditView method), 62
- `get_context_data()` (shoop.admin.modules.orders.views.OrderCreateFullRefundView method), 68
- `get_context_data()` (shoop.admin.modules.orders.views.OrderCreatePaymentView method), 68
- `get_context_data()` (shoop.admin.modules.orders.views.OrderCreateRefundView method), 68
- `get_context_data()` (shoop.admin.modules.orders.views.OrderCreateShipmentView method), 69
- `get_context_data()` (shoop.admin.modules.orders.views.OrderDetailView method), 66
- `get_context_data()` (shoop.admin.modules.orders.views.OrderEditView method), 67
- `get_context_data()` (shoop.admin.modules.orders.views.payment.OrderCreatePaymentView method), 63
- `get_context_data()` (shoop.admin.modules.orders.views.refund.OrderCreateRefundView method), 65
- `get_context_data()` (shoop.admin.modules.orders.views.refund.OrderCreateRefundView method), 64
- `get_context_data()` (shoop.admin.modules.orders.views.shipment.OrderCreateShipmentView method), 66
- `get_context_data()` (shoop.admin.modules.permission_groups.views.list.PermissionGroupListView method), 73
- `get_context_data()` (shoop.admin.modules.permission_groups.views.PermissionGroupView method), 73
- `get_context_data()` (shoop.admin.modules.products.views.edit.ProductEditView method), 84
- `get_context_data()` (shoop.admin.modules.products.views.edit_cross_sell.ProductEditCrossSellView method), 84

method), 85

get_context_data() (shoop.admin.modules.products.views.edit.ProductEditView method), 86

get_context_data() (shoop.admin.modules.products.views.edit.PackageEditView method), 87

get_context_data() (shoop.admin.modules.products.views.edit.ProductParameterEditView method), 87

get_context_data() (shoop.admin.modules.products.views.edit.VariantEditView method), 88

get_context_data() (shoop.admin.modules.products.views.ProductTextSearchView method), 89

get_context_data() (shoop.admin.modules.products.views.ProductDetailView method), 89

get_context_data() (shoop.admin.modules.products.views.ProductMediaEditView method), 90

get_context_data() (shoop.admin.modules.products.views.ProductPackageView method), 90

get_context_data() (shoop.admin.modules.products.views.ProductVariantView method), 90

get_context_data() (shoop.admin.modules.system.views.telegram.TelegramView method), 107

get_context_data() (shoop.admin.modules.users.views.list.UserListView method), 112

get_context_data() (shoop.admin.modules.users.views.password.PasswordChangeView method), 113

get_context_data() (shoop.admin.modules.users.views.permissions.PermissionListView method), 114

get_context_data() (shoop.admin.modules.users.views.UserChangePasswordView method), 115

get_context_data() (shoop.admin.modules.users.views.UserChangePermissionsView method), 115

get_context_data() (shoop.admin.modules.users.views.UserListView method), 114

get_context_data() (shoop.admin.utils.views.CreateOrUpdateView method), 122

get_context_data() (shoop.admin.utils.views.PicotableView method), 123

get_context_data() (shoop.admin.views.dashboard.DashboardView method), 123

get_context_data() (shoop.campaigns.admin_module.views.BasketLinkListView method), 141

get_context_data() (shoop.campaigns.admin_module.views.CatalogCampaignListView method), 140

get_context_data() (shoop.campaigns.admin_module.views.CouponListView method), 141

get_context_data() (shoop.front.admin_module.carts.views.CartListView method), 268

get_context_data() (shoop.front.apps.auth.views.LoginView method), 270

get_context_data() (shoop.front.apps.simple_search.views.SearchView method), 277

get_context_data() (shoop.front.checkout.confirm.ConfirmPhase method), 283

get_context_data() (shoop.front.checkout.single_page.SingleCheckoutView method), 285

get_context_data() (shoop.front.views.basket.DefaultBasketView method), 291

get_context_data() (shoop.front.views.category.CategoryView method), 292

get_context_data() (shoop.front.views.index.IndexView method), 292

get_context_data() (shoop.front.views.order.OrderRequiresVerificationView method), 292

get_context_data() (shoop.front.views.payment.ProcessPaymentView method), 293

get_context_data() (shoop.front.views.product.ProductDetailView method), 293

get_context_data() (shoop.notify.admin_module.views.edit.ScriptEditView method), 299

get_context_data() (shoop.notify.admin_module.views.editor.EditScriptContentView method), 300

get_context_data() (shoop.notify.admin_module.views.EditScriptContentView method), 300

get_context_data() (shoop.notify.admin_module.views.ScriptEditView method), 300

get_context_data() (shoop.simple_cms.plugins.PageLinksPlugin method), 320

get_context_data() (shoop.simple_supplier.admin_module.views.StocksListView method), 322

get_context_data() (shoop.themes.classic_gray.plugins.ProductCrossSellsPlugin method), 337

get_context_data() (shoop.themes.classic_gray.plugins.ProductHighlightPlugin method), 337

get_context_data() (shoop.xtheme.admin_module.views.ThemeConfigDetailView method), 352

get_context_data() (shoop.xtheme.admin_module.views.ThemeConfigView method), 351

get_context_data() (shoop.xtheme.plugins.category_links.CategoryLinksPlugin method), 353

get_context_data() (shoop.xtheme.plugins.image.ImagePlugin method), 354

get_context_data() (shoop.xtheme.plugins.social_media_links.SocialMediaLinksPlugin method), 356

get_context_data() (shoop.xtheme.TemplatedPlugin method), 375

get_context_data() (shoop.xtheme.views.editor.EditorView method), 358

get_context_data() (shoop.core.pricing.PricingModule method), 249

get_context_data() (shoop.core.taxing.TaxModule method), 253

get_context_from_order_source() (shoop.core.taxing.TaxModule method), 253

get_context_from_request() (shoop.core.pricing.PricingModule method), 249

get_context_from_request() (shoop.core.pricing.PricingModule method), 249

(`shoop.core.taxing.TaxModule` method), `get_default_category()` (in module `shoop.testing.factories`), 331

`get_costs()` (`shoop.core.models.FixedCostBehaviorComponent` method), 196

`get_costs()` (`shoop.core.models.RoundingBehaviorComponent` method), 222

`get_costs()` (`shoop.core.models.Service` method), 224

`get_costs()` (`shoop.core.models.ServiceBehaviorComponent` method), 225

`get_costs()` (`shoop.core.models.WaivingCostBehaviorComponent` method), 236

`get_costs()` (`shoop.core.models.WeightBasedPricingBehaviorComponent` method), 237

`get_costs()` (`shoop.testing.models.ExpensiveSwedenBehaviorComponent` method), 327

`get_country_display()` (`shoop.core.models.ImmutableAddress` method), 198

`get_country_display()` (`shoop.core.models.MutableAddress` method), 199

`get_create_or_change_title()` (in module `shoop.admin.utils.views`), 122

`get_currency_name()` (in module `shoop.utils.i18n`), 343

`get_current_babel_locale()` (in module `shoop.utils.i18n`), 342

`get_current_phase()` (`shoop.front.checkout.CheckoutProcess` method), 286

`get_current_stock_value()` (in module `shoop.simple_supplier.utils`), 324

`get_current_theme()` (in module `shoop.xtheme`), 376

`get_custom_carrier()` (in module `shoop.testing.factories`), 331

`get_custom_payment_processor()` (in module `shoop.testing.factories`), 331

`get_customer_data()` (`shoop.admin.modules.orders.views.edit_order_edit_view` method), 62

`get_customer_data()` (`shoop.admin.modules.orders.views.order_edit_view` method), 67

`get_dashboard_blocks()` (`shoop.admin.base.AdminModule` method), 124

`get_dashboard_blocks()` (`shoop.admin.modules.contacts.ContactModule` method), 57

`get_dashboard_blocks()` (`shoop.admin.modules.demo.DemoModule` method), 58

`get_dashboard_blocks()` (`shoop.admin.modules.orders.OrderModule` method), 72

`get_dashboard_blocks()` (`shoop.front.admin_module.CartAdminModule` method), 269

`get_data()` (`shoop.admin.utils.picotable.Picotable` method), 120

`get_data()` (`shoop.admin.views.select.MultiselectAjaxView` method), 124

`get_data_dict()` (in module `shoop.utils.models`), 343

`get_default_attribute_set()` (in module `shoop.testing.factories`), 331

`get_default_company_group()` (`shoop.core.models.CustomerTaxGroup` class method), 195

`get_default_customer_group()` (in module `shoop.testing.factories`), 331

`get_default_edit_toolbar()` (in module `shoop.admin.toolbar`), 130

`get_default_group()` (`shoop.core.models.Contact` class method), 192

`get_default_model_permissions()` (in module `shoop.admin.utils.permissions`), 118

`get_default_payment_method()` (in module `shoop.testing.factories`), 331

`get_default_person_group()` (`shoop.core.models.CustomerTaxGroup` class method), 195

`get_default_product()` (in module `shoop.testing.factories`), 331

`get_default_product_type()` (in module `shoop.testing.factories`), 331

`get_default_sales_unit()` (in module `shoop.testing.factories`), 331

`get_default_settings_module()` (`shoop.apps.AppConfig` method), 135

`get_default_shipping_method()` (in module `shoop.testing.factories`), 331

`get_default_shop()` (in module `shoop.testing.factories`), 331

`get_default_shop_product()` (in module `shoop.testing.factories`), 331

`get_default_supplier()` (in module `shoop.testing.factories`), 331

`get_default_tax()` (in module `shoop.testing.factories`), 331

`get_default_tax_class()` (in module `shoop.testing.factories`), 331

`get_delivery_pdf()` (in module `shoop.order_printouts.admin_module.views`), 146

`get_delivery_time()` (`shoop.core.models.ServiceBehaviorComponent` method), 225

`get_discard_button()` (in module `shoop.admin.toolbar`), 130

`get_discount_lines()` (`shoop.campaigns.models.basket_line_effects.BasketLineEffect` method), 155

`get_discount_lines()` (`shoop.campaigns.models.basket_line_effects.DiscountLineEffect` method), 157

`get_discount_lines()` (`shoop.campaigns.models.basket_line_effects.FreeProductLineEffect` method), 156

`get_discount_lines()` (`shoop.campaigns.models.BasketLineEffect` method), 170

`get_discount_modules()` (in module `shoop.core.pricing`), 244

- `get_display_value()` (shoop.admin.utils.picotable.Column method), 120
- `get_edit_and_list_urls()` (in module shoop.admin.utils.urls), 121
- `get_editor_args()` (shoop.admin.modules.products.forms.variable_variables_data_form.VariationVariablesDataForm method), 79
- `get_editor_args()` (shoop.admin.modules.products.forms.VariationVariablesDataForm method), 82
- `get_editor_form_class()` (shoop.xtheme.Plugin method), 373
- `get_effective_form_data()` (in module shoop.utils.forms), 341
- `get_effective_name()` (shoop.core.models.Service method), 224
- `get_effective_name()` (shoop.core.models.ServiceProvider method), 227
- `get_enabled_addons()` (in module shoop.addons.manager), 41
- `get_enabled_addons()` (shoop.addons.admin_module.views.list.AddonEnabledDisableForm method), 38
- `get_entries()` (shoop.admin.breadcrumbs.Breadcrumbs method), 126
- `get_enum_choices_dict()` (in module shoop.notify.admin_module.utils), 301
- `get_event_identifier_text()` (shoop.notify.admin_module.views.list.ScriptListView method), 300
- `get_event_identifier_text()` (shoop.notify.admin_module.views.ScriptListView method), 301
- `get_extra_fields()` (shoop.admin.modules.orders.views.shipment.ShipmentForm method), 65
- `get_faker()` (in module shoop.testing.factories), 331
- `get_field()` (shoop.notify.typology.Decimal method), 312
- `get_field()` (shoop.notify.typology.Email method), 312
- `get_field()` (shoop.notify.typology.Enum method), 313
- `get_field()` (shoop.notify.typology.Integer method), 312
- `get_field()` (shoop.notify.typology.Model method), 313
- `get_field()` (shoop.notify.typology.Type method), 311
- `get_field()` (shoop.notify.typology.URL method), 313
- `get_fields()` (shoop.core.api.orders.OrderSerializer method), 179
- `get_filter()` (shoop.admin.utils.picotable.PicotableViewMixin method), 120
- `get_final_lines()` (shoop.core.order_creator.OrderSource method), 239
- `get_folder_name()` (in module shoop.admin.modules.media.views), 60
- `get_footer_information()` (in module shoop.order_printouts.admin_module.views), 316
- `get_form()` (shoop.admin.form_part.FormPartsViewMixin method), 127
- `get_form()` (shoop.admin.modules.orders.views.OrderCreatePaymentView method), 68
- `get_form()` (shoop.admin.modules.orders.views.OrderCreateRefundView method), 68
- `get_form()` (shoop.admin.modules.orders.views.OrderCreateShipmentView method), 69
- `get_form()` (shoop.admin.modules.orders.views.payment.OrderCreatePaymentView method), 69
- `get_form()` (shoop.admin.modules.orders.views.refund.OrderCreateRefundView method), 64
- `get_form()` (shoop.admin.modules.orders.views.shipment.OrderCreateShipmentView method), 66
- `get_form()` (shoop.admin.modules.service_providers.views.ServiceProviderView method), 93
- `get_form()` (shoop.admin.modules.suppliers.views.edit.SupplierEditView method), 105
- `get_form()` (shoop.admin.modules.suppliers.views.SupplierEditView method), 106
- `get_form()` (shoop.front.apps.auth.views.LoginView method), 345
- `get_form()` (shoop.front.apps.customer_information.views.CompanyEditView method), 272
- `get_form()` (shoop.front.apps.customer_information.views.CustomerEditView method), 272
- `get_form()` (shoop.front.checkout.addresses.AddressesPhaseView method), 283
- `get_form()` (shoop.front.checkout.single_page.SingleCheckoutPhaseView method), 285
- `get_form()` (shoop.testing.admin_module.mock_view.MockableView method), 326
- `get_form()` (shoop.xtheme.admin_module.views.ThemeConfigDetailView method), 345
- `get_form_class()` (shoop.admin.form_part.FormPartsViewMixin method), 127
- `get_form_class()` (shoop.admin.modules.users.views.detail.UserDetailView method), 112
- `get_form_class()` (shoop.admin.modules.users.views.permissions.UserChangePermissionsView method), 114
- `get_form_class()` (shoop.admin.modules.users.views.UserChangePermissionsView method), 115
- `get_form_class()` (shoop.admin.modules.users.views.UserDetailView method), 114
- `get_form_defs()` (shoop.admin.form_part.FormPart method), 127
- `get_form_defs()` (shoop.admin.modules.categories.views.edit.CategoryBaseForm method), 49
- `get_form_defs()` (shoop.admin.modules.contact_groups.views.forms.ContactGroupForm method), 51
- `get_form_defs()` (shoop.admin.modules.contact_groups.views.forms.ContactGroupForm method), 52
- `get_form_defs()` (shoop.admin.modules.contacts.views.edit.ContactAddressForm method), 55
- `get_form_defs()` (shoop.admin.modules.contacts.views.edit.ContactBaseForm method), 55
- `get_form_defs()` (shoop.admin.modules.products.views.edit.BaseProductForm method), 55

method), 83

get_form_defs() (shoop.admin.modules.products.views.edit_product_attributes), 83

get_form_defs() (shoop.admin.modules.products.views.edit_product_base_form_part), 83

get_form_defs() (shoop.admin.modules.products.views.edit_product_links_form_part), 83

get_form_defs() (shoop.admin.modules.products.views.edit_product_media_form_part), 86

get_form_defs() (shoop.admin.modules.products.views.edit_product_shipping_form_part), 87

get_form_defs() (shoop.admin.modules.products.views.edit_product_variations_child_form_part), 88

get_form_defs() (shoop.admin.modules.products.views.edit_product_variations_shop_base_form_part), 88

get_form_defs() (shoop.admin.modules.services.base_form_part), 96

get_form_defs() (shoop.admin.modules.services.behavior_form_part), 97

get_form_defs() (shoop.admin.modules.services.weight_based_pricing_weight_base_form_part), 96

get_form_defs() (shoop.admin.modules.shops.views.edit_shop_base_form_part), 103

get_form_defs() (shoop.admin.modules.shops.views.edit_shop_base_form_part), 102

get_form_defs() (shoop.campaigns.admin_module.form_parts.catalog_campaign_base_form_part), 143

get_form_defs() (shoop.campaigns.admin_module.form_parts.catalog_campaign_base_form_part), 142

get_form_defs() (shoop.campaigns.admin_module.form_parts.sales_ranges_form_part), 142

get_form_defs() (shoop.customer_group_pricing.admin_form_part.customer_group_pricing_form_part), 264

get_form_defs() (shoop.simple_supplier.admin_module.form_parts.simple_supplier_form_part), 321

get_form_kwargs() (shoop.admin.modules.orders.views.order_create_refund_view), 68

get_form_kwargs() (shoop.admin.modules.orders.views.order_create_payment_view), 68

get_form_kwargs() (shoop.admin.modules.orders.views.order_create_refund_view), 68

get_form_kwargs() (shoop.admin.modules.orders.views.order_create_shipping_view), 69

get_form_kwargs() (shoop.admin.modules.orders.views.payment_order_create_payment_view), 63

get_form_kwargs() (shoop.admin.modules.orders.views.refund_order_create_refund_view), 65

get_form_kwargs() (shoop.admin.modules.orders.views.refund_order_create_refund_view), 64

get_form_kwargs() (shoop.admin.modules.orders.views.shipment_order_create_shipment_view), 66

get_form_kwargs() (shoop.admin.modules.products.views.edit_cross_sell_form_part), 85

get_form_kwargs() (shoop.admin.modules.products.views.edit_media_form_part), 86

get_form_kwargs() (shoop.admin.modules.products.views.edit_product_attributes), 89

get_form_kwargs() (shoop.admin.modules.products.views.edit_product_base_form_part), 90

get_form_kwargs() (shoop.admin.modules.products.views.edit_product_links_form_part), 113

get_form_kwargs() (shoop.admin.modules.products.views.edit_product_media_form_part), 114

get_form_kwargs() (shoop.admin.modules.products.views.edit_product_shipping_form_part), 115

get_form_kwargs() (shoop.admin.modules.products.views.edit_product_variations_child_form_part), 115

get_form_kwargs() (shoop.admin.modules.products.views.edit_product_variations_shop_base_form_part), 122

get_form_kwargs() (shoop.front.apps.auth.views.recover_password_confirm), 271

get_form_kwargs() (shoop.front.checkout.methods.methods_phase), 284

get_form_kwargs() (shoop.front.checkout.methods.methods_phase), 127

get_form_kwargs() (shoop.admin.modules.products.views.edit_parent), 87

get_form_kwargs() (shoop.admin.form_part.form_parts_view_mixin), 127

get_form_kwargs() (shoop.campaigns.admin_module.views.catalog_campaign_base_form_part), 140

get_form_kwargs() (shoop.campaigns.admin_module.views.catalog_campaign_base_form_part), 140

get_form_kwargs() (shoop.core.models.person_contact), 318

get_form_kwargs() (shoop.simple_cms.models.page), 372

get_form_kwargs() (shoop.core.utils.model_caching_descriptor.model_caching_descriptor), 372

get_id() (shoop.core.utils.model_caching_descriptor.model_caching_descriptor), 372

get_id_display() (shoop.core.models.counter), 132

get_identifier_to_object_map() (in module shoop.admin.template_helpers.shoop_admin), 132

get_identifier_to_spec_map() (in module shoop.admin.template_helpers.shoop_admin), 132

get_initial() (shoop.admin.modules.products.views.edit_product_base_form_part), 83

get_initial() (shoop.admin.modules.users.views.detail.user_detail_view), 132

get_initial() (shoop.admin.modules.users.views.detail.user_detail_view), 132

get_initial() (shoop.admin.modules.users.views.user_detail_view), 132

get_initial() (shoop.front.checkout.addresses.addresses_phase), 132

get_initial() (shoop.front.checkout.methods.methods_phase), 132

get_initial() (shoop.notify.admin_module.forms.script_item_edit_form), 132

get_initial() (shoop.notify.admin_module.forms.script_item_edit_form), 132

`get_initial()` (shoop.testing.simple_checkout_phase.TestMethodCheckoutPhase method), 335

`get_initial_order_data()` (shoop.admin.modules.orders.views.OrderEditView method), 62

`get_initial_order_data()` (shoop.admin.modules.orders.views.OrderEditView method), 67

`get_initial_order_status()` (in module shoop.testing.factories), 331

`get_installation_key()` (in module shoop.core.telemetry), 263

`get_kind_display()` (shoop.core.models.OrderLogEntry method), 204

`get_kind_display()` (shoop.core.models.ProductMedia method), 219

`get_kind_display()` (shoop.utils.analog.BaseLogEntry method), 338

`get_known_additional_data()` (shoop.core.models.Order method), 201

`get_known_settings()` (in module shoop.apps), 135

`get_known_settings()` (in module shoop.apps.settings), 133

`get_known_settings_documentation()` (in module shoop.utils.settings_doc), 349

`get_known_settings_with_comments()` (in module shoop.utils.settings_doc), 349

`get_label()` (shoop.admin.utils.bs3_renderers.AdminFieldRenderer method), 117

`get_language_choices()` (in module shoop.core.templatetags.shoop_common), 256

`get_language_code_display()` (shoop.campaigns.models.campaigns.BasketCampaignTranslation method), 160

`get_language_code_display()` (shoop.campaigns.models.campaigns.CatalogCampaignTranslation method), 161

`get_language_code_display()` (shoop.simple_cms.models.PageTranslation method), 319

`get_language_display()` (shoop.core.models.Contact method), 192

`get_language_display()` (shoop.core.models.Order method), 201

`get_language_name()` (in module shoop.utils.i18n), 342

`get_last_submission_data()` (in module shoop.core.telemetry), 263

`get_last_submission_time()` (in module shoop.core.telemetry), 263

`get_layout_data()` (shoop.xtheme.models.SavedViewConfig method), 366

`get_lifetime_sales_block()` (in module shoop.admin.modules.orders.dashboard), 69

`get_line_data_for_edit()` (in module shoop.admin.modules.orders.views.edit), 62

`get_line_data_for_edit()` (shoop.core.models.Service method), 224

`get_lines()` (shoop.core.order_creator.OrderSource method), 240

`get_lines()` (shoop.front.basket.objects.BaseBasket method), 279

`get_links()` (shoop.xtheme.plugins.social_media_links.SocialMediaLinksPlugin method), 356

`get_locally_formatted_datetime()` (in module shoop.utils.i18n), 342

`get_log()` (shoop.addons.installer.PackageInstaller method), 41

`get_manufacturers()` (in module shoop.front.template_helpers.category), 288

`get_matching()` (shoop.campaigns.models.BasketCampaign class method), 171

`get_matching()` (shoop.campaigns.models.campaigns.BasketCampaign class method), 158

`get_matching()` (shoop.campaigns.models.campaigns.CatalogCampaign class method), 158

`get_matching()` (shoop.campaigns.models.CatalogCampaign class method), 174

`get_matching_types()` (shoop.notify.base.Variable method), 306

`get_matching_types()` (shoop.notify.Variable method), 315

`get_menu_category_icons()` (shoop.addons.admin_module.AddonModule method), 40

`get_menu_category_icons()` (shoop.admin.base.AdminModule method), 124

`get_menu_category_icons()` (shoop.admin.modules.attributes.AttributeModule method), 48

`get_menu_category_icons()` (shoop.admin.modules.contacts.ContactModule method), 57

`get_menu_category_icons()` (shoop.admin.modules.media.MediaModule method), 61

`get_menu_category_icons()` (shoop.admin.modules.orders.OrderModule method), 71

`get_menu_category_icons()` (shoop.admin.modules.permission_groups.PermissionGroupModule method), 73

`get_menu_category_icons()` (shoop.admin.modules.products.ProductModule method), 91

`get_menu_category_icons()` (shoop.admin.modules.service_providers.ServiceProviderModule method), 94

[get_menu_category_icons\(\)](#) (shoop.admin.modules.system.SystemModule method), 107
[get_menu_category_icons\(\)](#) (shoop.admin.modules.taxes.TaxModule method), 111
[get_menu_category_icons\(\)](#) (shoop.admin.modules.users.UserModule method), 116
[get_menu_category_icons\(\)](#) (shoop.campaigns.admin_module.CampaignAdminModule method), 145
[get_menu_category_icons\(\)](#) (shoop.front.admin_module.CartAdminModule method), 269
[get_menu_category_icons\(\)](#) (shoop.notify.admin_module.NotifyAdminModule method), 302
[get_menu_category_icons\(\)](#) (shoop.simple cms.admin_module.SimpleCMSAdminModule method), 318
[get_menu_category_icons\(\)](#) (shoop.simple_supplier.admin_module.StocksAdminModule method), 322
[get_menu_category_icons\(\)](#) (shoop.xtheme.admin_module.XthemeAdminModule method), 352
[get_menu_entries\(\)](#) (shoop.addons.admin_module.AddonModule method), 40
[get_menu_entries\(\)](#) (shoop.admin.base.AdminModule method), 124
[get_menu_entries\(\)](#) (shoop.admin.modules.attributes.AttributeModule method), 48
[get_menu_entries\(\)](#) (shoop.admin.modules.categories.CategoryModule method), 50
[get_menu_entries\(\)](#) (shoop.admin.modules.contact_groups.ContactGroupModule method), 53
[get_menu_entries\(\)](#) (shoop.admin.modules.contacts.ContactModule method), 57
[get_menu_entries\(\)](#) (shoop.admin.modules.demo.DemoModule method), 58
[get_menu_entries\(\)](#) (shoop.admin.modules.manufacturers.ManufacturerModule method), 60
[get_menu_entries\(\)](#) (shoop.admin.modules.media.MediaModule method), 61
[get_menu_entries\(\)](#) (shoop.admin.modules.orders.OrderModule method), 71
[get_menu_entries\(\)](#) (shoop.admin.modules.permission_groups.PermissionGroupModule method), 73
[get_menu_entries\(\)](#) (shoop.admin.modules.product_types.ProductTypeModule method), 75
[get_menu_entries\(\)](#) (shoop.admin.modules.products.ProductModule method), 91
[get_menu_entries\(\)](#) (shoop.admin.modules.sales_units.SalesUnitModule method), 92
[get_menu_entries\(\)](#) (shoop.admin.modules.service_providers.ServiceProviderModule method), 94
[get_menu_entries\(\)](#) (shoop.admin.modules.services.ServiceModule method), 101
[get_menu_entries\(\)](#) (shoop.admin.modules.shops.ShopModule method), 104
[get_menu_entries\(\)](#) (shoop.admin.modules.suppliers.SupplierModule method), 106
[get_menu_entries\(\)](#) (shoop.admin.modules.system.SystemModule method), 107
[get_menu_entries\(\)](#) (shoop.admin.modules.taxes.TaxModule method), 111
[get_menu_entries\(\)](#) (shoop.admin.modules.users.UserModule method), 116
[get_menu_entries\(\)](#) (shoop.campaigns.admin_module.CampaignAdminModule method), 145
[get_menu_entries\(\)](#) (shoop.default_tax.admin_module.TaxRulesAdminModule method), 267
[get_menu_entries\(\)](#) (shoop.front.admin_module.CartAdminModule method), 269
[get_menu_entries\(\)](#) (shoop.notify.admin_module.NotifyAdminModule method), 302
[get_menu_entries\(\)](#) (shoop.simple cms.admin_module.SimpleCMSAdminModule method), 318
[get_menu_entries\(\)](#) (shoop.simple_supplier.admin_module.StocksAdminModule method), 322
[get_menu_entries\(\)](#) (shoop.xtheme.admin_module.XthemeAdminModule method), 352
[get_menu_entry_categories\(\)](#) (in module shoop.admin.menu), 127
[get_menu_entry_categories\(\)](#) (in module shoop.admin.template_helpers.shoop_admin), 127
[get_menu_items\(\)](#) (shoop.order_printouts.admin_module.toolbar.SimplePrintoutModule method), 316
[get_method\(\)](#) (shoop.front.checkout.methods.PaymentMethodPhase method), 284
[get_method\(\)](#) (shoop.front.checkout.methods.ShippingMethodPhase method), 284
[get_missing_permissions\(\)](#) (in module shoop.admin.utils.permissions), 118
[get_mockers\(\)](#) (shoop.testing.admin_module.mock_view.MockableView method), 326
[get_mode_display\(\)](#) (shoop.core.models.Product model), 222
[get_mode_display\(\)](#) (shoop.core.models.RoundingBehaviorComponent model), 222
[get_model\(\)](#) (shoop.admin.modules.users.views.list.UserListView method), 112
[get_model\(\)](#) (shoop.admin.modules.users.views.UserListView method), 114

[get_model\(\) \(shoop.notify.typology.Model method\), 313](#)
[get_model_front_url\(\) \(in module shoop.admin.utils.urls\), 122](#)
[get_model_url\(\) \(in module shoop.admin.utils.urls\), 121](#)
[get_model_url\(\) \(shoop.admin.base.AdminModule method\), 125](#)
[get_model_url\(\) \(shoop.admin.modules.attributes.AttributeModule method\), 48](#)
[get_model_url\(\) \(shoop.admin.modules.categories.CategoryModule method\), 50](#)
[get_model_url\(\) \(shoop.admin.modules.contact_groups.ContactGroupModule method\), 53](#)
[get_model_url\(\) \(shoop.admin.modules.contacts.ContactModule method\), 57](#)
[get_model_url\(\) \(shoop.admin.modules.manufacturers.ManufacturerModule method\), 60](#)
[get_model_url\(\) \(shoop.admin.modules.orders.OrderModule method\), 72](#)
[get_model_url\(\) \(shoop.admin.modules.permission_groups.PermissionGroupModule method\), 73](#)
[get_model_url\(\) \(shoop.admin.modules.product_types.ProductTypeModule method\), 75](#)
[get_model_url\(\) \(shoop.admin.modules.products.ProductModule method\), 91](#)
[get_model_url\(\) \(shoop.admin.modules.sales_units.SalesUnitModule method\), 93](#)
[get_model_url\(\) \(shoop.admin.modules.service_providers.ServiceProviderModule method\), 94](#)
[get_model_url\(\) \(shoop.admin.modules.services.ServiceModule method\), 101](#)
[get_model_url\(\) \(shoop.admin.modules.shops.ShopModule method\), 104](#)
[get_model_url\(\) \(shoop.admin.modules.suppliers.SupplierModule method\), 106](#)
[get_model_url\(\) \(shoop.admin.modules.taxes.TaxModule method\), 111](#)
[get_model_url\(\) \(shoop.admin.modules.users.UserModule method\), 116](#)
[get_model_url\(\) \(shoop.campaigns.admin_module.CampaignAdminModule method\), 145](#)
[get_model_url\(\) \(shoop.default_tax.admin_module.TaxRulesAdminModule method\), 267](#)
[get_model_url\(\) \(shoop.notify.admin_module.NotifyAdminModule method\), 302](#)
[get_model_url\(\) \(shoop.simple_cms.admin_module.SimpleCMSAdminModule method\), 318](#)
[get_module_choices\(\) \(shoop.core.modules.interface.ModuleInterface class method\), 238](#)
[get_module_choices\(\) \(shoop.core.modules.ModuleInterface class method\), 238](#)
[get_module_display\(\) \(shoop.admin.modules.suppliers.views.list.SupplierListView method\), 105](#)
[get_module_display\(\) \(shoop.admin.modules.suppliers.views.SupplierListView method\), 106](#)
[get_module_implementation_map\(\) \(shoop.core.modules.interface.ModuleInterface class method\), 238](#)
[get_module_implementation_map\(\) \(shoop.core.modules.ModuleInterface class method\), 238](#)
[get_module_urls\(\) \(in module shoop.admin.module_registry\), 127](#)
[get_modules\(\) \(in module shoop.admin.module_registry\), 127](#)
[get_name\(\) \(shoop.admin.modules.services.behavior_form_part.BehaviorFormPart method\), 96](#)
[get_name\(\) \(shoop.campaigns.admin_module.form_sets.BaseFormset method\), 144](#)
[get_name\(\) \(in module shoop.notify.admin_module.utils\), 301](#)
[get_new_lines\(\) \(shoop.campaigns.modules.BasketCampaignModule method\), 178](#)
[get_permissions\(\) \(shoop.core.order_creator.OrderSourceModifierModule method\), 241](#)
[get_type\(\) \(shoop.admin.utils.views.CreateOrUpdateView method\), 122](#)
[get_latest_products\(\) \(in module shoop.front.template_helpers.general\), 288](#)
[get_module_by_created_on\(\) \(shoop.campaigns.models.BasketCampaign method\), 171](#)
[get_next_by_created_on\(\) \(shoop.campaigns.models.Campaign method\), 173](#)
[get_next_by_created_on\(\) \(shoop.campaigns.models.campaigns.BasketCampaign method\), 159](#)
[get_next_by_created_on\(\) \(shoop.campaigns.models.campaigns.Campaign method\), 157](#)
[get_next_by_created_on\(\) \(shoop.campaigns.models.campaigns.CatalogCampaign method\), 158](#)
[get_next_by_created_on\(\) \(shoop.campaigns.models.campaigns.Coupon method\), 160](#)
[get_next_by_created_on\(\) \(shoop.campaigns.models.campaigns.CouponUsage method\), 159](#)
[get_next_by_created_on\(\) \(shoop.campaigns.models.CatalogCampaign method\), 174](#)
[get_next_by_created_on\(\) \(shoop.campaigns.models.Coupon method\), 177](#)
[get_next_by_created_on\(\) \(shoop.core.models.Contact method\), 192](#)
[get_next_by_created_on\(\)](#)

(shoop.core.models.Manufacturer method), 198

get_next_by_created_on() (shoop.core.models.Order method), 201

get_next_by_created_on() (shoop.core.models.OrderLogEntry method), 205

get_next_by_created_on() (shoop.core.models.Payment method), 206

get_next_by_created_on() (shoop.core.models.Product method), 212, 215

get_next_by_created_on() (shoop.core.models.Shipment method), 228

get_next_by_created_on() (shoop.front.models.stored_basket.StoredBasket method), 287

get_next_by_created_on() (shoop.front.models.StoredBasket method), 288

get_next_by_created_on() (shoop.notify.models.Notification method), 305

get_next_by_created_on() (shoop.notify.models.notification.Notification method), 305

get_next_by_created_on() (shoop.notify.models.Script method), 306

get_next_by_created_on() (shoop.notify.models.script.Script method), 305

get_next_by_created_on() (shoop.simple cms.models.Page method), 319

get_next_by_created_on() (shoop.simple_supplier.models.StockAdjustment method), 323

get_next_by_created_on() (shoop.utils.analog.BaseLogEntry method), 338

get_next_by_created_on() (shoop.xtheme.models.SavedViewConfig method), 366

get_next_by_modified_on() (shoop.campaigns.models.BasketCampaign method), 171

get_next_by_modified_on() (shoop.campaigns.models.Campaign method), 173

get_next_by_modified_on() (shoop.campaigns.models.campaigns.BasketCampaign method), 159

get_next_by_modified_on() (shoop.campaigns.models.campaigns.Campaign method), 157

get_next_by_modified_on() (shoop.campaigns.models.campaigns.CatalogCampaign method), 158

get_next_by_modified_on() (shoop.campaigns.models.campaigns.Coupon method), 160

get_next_by_modified_on() (shoop.campaigns.models.campaigns.CouponUsage method), 159

get_next_by_modified_on() (shoop.campaigns.models.CatalogCampaign method), 174

get_next_by_modified_on() (shoop.campaigns.models.Coupon method), 177

get_next_by_modified_on() (shoop.core.models.Order method), 201

get_next_by_modified_on() (shoop.core.models.Product method), 212, 216

get_next_by_modified_on() (shoop.simple cms.models.Page method), 319

get_next_by_order_date() (shoop.core.models.Order method), 201

get_next_by_time() (shoop.core.models.PersistentCacheEntry method), 208

get_next_by_updated_on() (shoop.front.models.stored_basket.StoredBasket method), 287

get_next_by_updated_on() (shoop.front.models.StoredBasket method), 288

get_next_phase() (shoop.front.checkout.CheckoutProcess method), 286

get_notifications() (shoop.admin.base.AdminModule method), 124

get_notifications() (shoop.admin.modules.demo.DemoModule method), 58

get_notifications() (shoop.admin.modules.orders.OrderModule method), 72

get_notifications() (shoop.admin.modules.system.SystemModule method), 107

get_notifications() (shoop.front.apps.simple_order_notification.admin_mod method), 275

get_notifications() (shoop.notify.admin_module.NotifyAdminModule method), 302

get_notifications() (shoop.xtheme.admin_module.XthemeAdminModule method), 352

get_object() (shoop.admin.forms.widgets.BasePopupChoiceWidget method), 45

get_object() (shoop.admin.forms.widgets.ContactChoiceWidget method), 46

get_object() (shoop.admin.forms.widgets.MediaChoiceWidget method), 45

`get_object()` (shoop.admin.forms.widgets.ProductChoiceWidget method), 46

`get_object()` (shoop.admin.modules.contacts.views.ContactResetPasswordView method), 56

`get_object()` (shoop.admin.modules.contacts.views.reset.ContactResetPasswordView method), 56

`get_object()` (shoop.admin.modules.shops.views.edit.ShopEditView method), 103

`get_object()` (shoop.admin.modules.shops.views.ShopEditView method), 104

`get_object()` (shoop.admin.modules.suppliers.views.edit.SupplierEditView method), 105

`get_object()` (shoop.admin.modules.suppliers.views.SupplierEditView method), 106

`get_object()` (shoop.admin.utils.views.CreateOrUpdateView method), 122

`get_object()` (shoop.core.utils.model_caching_descriptor.ModelCachingDescriptor method), 257

`get_object()` (shoop.front.views.order.OrderCompleteView method), 292

`get_object()` (shoop.front.views.order.OrderRequiresVerificationView method), 292

`get_object()` (shoop.front.views.payment.ProcessPaymentView method), 293

`get_object()` (shoop.xtheme.admin_module.views.ThemeConfigDetailView method), 352

`get_object()` (shoop.xtheme.plugins.image.ImagePluginChoiceWidget method), 354

`get_object_abstract()` (shoop.admin.modules.categories.views.CategoryListView method), 50

`get_object_abstract()` (shoop.admin.modules.categories.views.list.CategoryListView method), 49

`get_object_abstract()` (shoop.admin.modules.contacts.views.ContactListView method), 56

`get_object_abstract()` (shoop.admin.modules.contacts.views.list.ContactListView method), 56

`get_object_abstract()` (shoop.admin.modules.orders.views.list.OrderListView method), 63

`get_object_abstract()` (shoop.admin.modules.orders.views.OrderListView method), 67

`get_object_abstract()` (shoop.admin.modules.products.views.list.ProductListView method), 88

`get_object_abstract()` (shoop.admin.modules.products.views.ProductListView method), 89

`get_object_abstract()` (shoop.admin.modules.service_providers.views.ShopServiceProviderListView method), 93

`get_object_abstract()` (shoop.admin.modules.users.views.list.UserListView method), 112

`get_object_abstract()` (shoop.admin.modules.users.views.UserListView method), 114

`get_object_abstract()` (shoop.admin.utils.picotable.PicotableViewMixin method), 120

`get_object_abstract()` (shoop.admin.utils.views.PicotableView method), 123

`get_object_abstract()` (shoop.front.admin_module.carts.views.CartListView method), 268

`get_object_abstract()` (shoop.notify.admin_module.views.list.ScriptListView method), 300

`get_object_abstract()` (shoop.notify.admin_module.views.ScriptListView method), 301

`get_object_abstract()` (shoop.simple_cms.admin_module.views.PageListView method), 318

`get_object_abstract()` (shoop.simple_supplier.admin_module.views.StocksListView method), 322

`get_object_url()` (shoop.admin.utils.picotable.PicotableViewMixin method), 120

`get_object_url()` (shoop.notify.admin_module.views.list.ScriptListView method), 300

`get_object_url()` (shoop.notify.admin_module.views.ScriptListView method), 301

`get_ordering_descriptor()` (in module shoop.admin.modules.orders.dashboard), 69

`get_or_select_template()` (shoop.xtheme.engine.XthemeEnvironment method), 360

`get_order_source_modifier_modules()` (in module shoop.core.order_creator), 238

`get_order_value_chart_dashboard_block()` (in module shoop.admin.modules.orders.dashboard), 69

`get_orderability_errors()` (shoop.core.models.ShopProduct method), 231

`get_orderability_errors()` (shoop.core.models.Supplier method), 233

`get_orderability_errors()` (shoop.core.suppliers.base.BaseSupplierModule method), 233

`get_orderability_errors()` (shoop.core.suppliers.BaseSupplierModule method), 251

`get_orders_by_currency()` (in module shoop.admin.modules.orders.dashboard), 69

`get_order_list_view_id_to_quantity_map()` (shoop.core.models.Product method), 212,

`get_page_by_identifier()` (shoop.simple_cms.template_helpers.SimpleCMS method), 220

`get_pagination_variables()` (in module shoop.front.template_helpers.general), 289

`get_payment_method()` (in module shoop.core.models.PaymentMethod method), 331

`get_payment_process_response()` (shoop.core.models.PaymentMethod method), 206

`get_payment_process_response()` (shoop.core.models.PaymentProcessor method), 207

`get_payment_process_response()` (shoop.testing.models.PseudoPaymentProcessor method), 329

- get_payment_status_display() (shoop.core.models.Order method), 201
- get_payment_urls() (in module shoop.front.views.payment), 293
- get_permission_object_from_string() (in module shoop.admin.utils.permissions), 118
- get_permissions_from_urls() (in module shoop.admin.utils.permissions), 118
- get_person_contact() (in module shoop.core.models), 197
- get_pip_path() (in module shoop.addons.installer), 41
- get_placeholder_layout() (shoop.xtheme.view_config.ViewConfig method), 372
- get_plugin_choices() (shoop.xtheme.Plugin class method), 374
- get_prefix_to_method_map() (shoop.front.basket.update_methods.BasketUpdateMethods method), 281
- get_prep_value() (shoop.core.fields.InternalIdentifierField method), 184
- get_previous_by_created_on() (shoop.campaigns.models.BasketCampaign method), 171
- get_previous_by_created_on() (shoop.campaigns.models.Campaign method), 173
- get_previous_by_created_on() (shoop.campaigns.models.campaigns.BasketCampaign method), 159
- get_previous_by_created_on() (shoop.campaigns.models.campaigns.Campaign method), 157
- get_previous_by_created_on() (shoop.campaigns.models.campaigns.CatalogCampaign method), 158
- get_previous_by_created_on() (shoop.campaigns.models.campaigns.Coupon method), 160
- get_previous_by_created_on() (shoop.campaigns.models.campaigns.CouponUsage method), 159
- get_previous_by_created_on() (shoop.campaigns.models.CatalogCampaign method), 174
- get_previous_by_created_on() (shoop.campaigns.models.Coupon method), 177
- get_previous_by_created_on() (shoop.core.models.Contact method), 193
- get_previous_by_created_on() (shoop.core.models.Manufacturer method), 198
- get_previous_by_created_on() (shoop.core.models.Order method), 201
- get_previous_by_created_on() (shoop.core.models.OrderLogEntry method), 205
- get_previous_by_created_on() (shoop.core.models.Payment method), 206
- get_previous_by_created_on() (shoop.core.models.Product method), 212, 216
- get_previous_by_created_on() (shoop.core.models.Shipment method), 228
- get_previous_by_created_on() (shoop.front.models.stored_basket.StoredBasket method), 287
- get_previous_by_created_on() (shoop.front.models.StoredBasket method), 288
- get_previous_by_created_on() (shoop.notify.models.Notification method), 305
- get_previous_by_created_on() (shoop.notify.models.notification.Notification method), 305
- get_previous_by_created_on() (shoop.notify.models.Script method), 306
- get_previous_by_created_on() (shoop.notify.models.script.Script method), 305
- get_previous_by_created_on() (shoop.simple_cms.models.Page method), 319
- get_previous_by_created_on() (shoop.simple_supplier.models.StockAdjustment method), 323
- get_previous_by_created_on() (shoop.utils.analog.BaseLogEntry method), 338
- get_previous_by_created_on() (shoop.xtheme.models.SavedViewConfig method), 366
- get_previous_by_modified_on() (shoop.campaigns.models.BasketCampaign method), 171
- get_previous_by_modified_on() (shoop.campaigns.models.Campaign method), 173
- get_previous_by_modified_on() (shoop.campaigns.models.campaigns.BasketCampaign method), 159
- get_previous_by_modified_on() (shoop.campaigns.models.campaigns.Campaign method), 157
- get_previous_by_modified_on() (shoop.campaigns.models.campaigns.CatalogCampaign method), 158

`get_previous_by_modified_on()`
(`shoop.campaigns.models.campaigns.Coupon`
method), 160

`get_previous_by_modified_on()`
(`shoop.campaigns.models.campaigns.CouponUsage`
method), 159

`get_previous_by_modified_on()`
(`shoop.campaigns.models.CatalogCampaign`
method), 174

`get_previous_by_modified_on()`
(`shoop.campaigns.models.Coupon` method),
177

`get_previous_by_modified_on()`
(`shoop.core.models.Order` method), 201

`get_previous_by_modified_on()`
(`shoop.core.models.Product` method), 212,
216

`get_previous_by_modified_on()`
(`shoop.simple cms.models.Page` method),
319

`get_previous_by_order_date()` (`shoop.core.models.Order`
method), 201

`get_previous_by_time()` (`shoop.core.models.PersistentCacheEntry`
method), 208

`get_previous_by_updated_on()`
(`shoop.front.models.stored_basket.StoredBasket`
method), 287

`get_previous_by_updated_on()`
(`shoop.front.models.StoredBasket` method),
288

`get_previous_phase()` (`shoop.front.checkout.CheckoutProcess`
method), 286

`get_price()` (`shoop.core.models.Product` method), 212,
216

`get_price_display_options()` (`shoop.core.models.Contact`
method), 193

`get_price_display_options()`
(`shoop.core.models.ContactGroup` method),
193

`get_price_info()` (in module
`shoop.admin.modules.orders.views.edit`),
62

`get_price_info()` (in module `shoop.core.pricing`), 244

`get_price_info()` (`shoop.core.models.Product` method),
212, 216

`get_price_info()` (`shoop.core.pricing.default_pricing.DefaultPricingModule`
method), 243

`get_price_info()` (`shoop.core.pricing.PricingModule`
method), 249

`get_price_info()` (`shoop.customer_group_pricing.module.CustomerGroupPricingModule`
method), 265

`get_price_infos()` (in module `shoop.core.pricing`), 244

`get_price_infos()` (`shoop.core.pricing.PricingModule`
method), 249

`get_priced_children()` (`shoop.core.models.Product`
method), 212, 216

`get_pricing_module()` (in module `shoop.core.pricing`),
244

`get_pricing_steps()` (in module `shoop.core.pricing`), 244

`get_pricing_steps()` (`shoop.core.pricing.DiscountModule`
method), 243

`get_pricing_steps()` (`shoop.core.pricing.PricingModule`
method), 249

`get_pricing_steps_for_products()` (in module
`shoop.core.pricing`), 244

`get_pricing_steps_for_products()`
(`shoop.core.pricing.DiscountModule` method),
244

`get_pricing_steps_for_products()`
(`shoop.core.pricing.PricingModule` method),
249

`get_priority_display()` (`shoop.notify.models.Notification`
method), 305

`get_priority_display()` (`shoop.notify.models.notification.Notification`
method), 305

`get_product_cross_sells()` (in module
`shoop.front.template_helpers.product`), 289

`get_product_ids_and_quantities()`
(`shoop.core.models.Order` method), 201

`get_product_ids_and_quantities()`
(`shoop.front.basket.objects.BaseBasket`
method), 280

`get_product_lines()` (`shoop.core.order_creator.OrderSource`
method), 240

`get_product_summary()` (`shoop.core.models.Order`
method), 201

`get_products_bought_with()` (in module
`shoop.front.template_helpers.product`), 289

`get_products_by_brand()` (in module
`shoop.front.utils.product_statistics`), 290

`get_products_by_same_categories()` (in module
`shoop.front.utils.product_statistics`), 290

`get_products_ordered_with()` (in module
`shoop.front.utils.product_statistics`), 290

`get_provide_objects()` (in module `shoop.apps.provides`),
132

`get_provide_specs_and_objects()` (in module
`shoop.apps.provides`), 132

`get_public_media()` (`shoop.core.models.Product`
method), 212, 216

`get_purchased_attachments()` (`shoop.core.models.Order`
method), 201

`get_queryset()` (`shoop.admin.modules.attributes.views.AttributeListView`
method), 47

`get_queryset()` (`shoop.admin.modules.attributes.views.list.AttributeListView`
method), 47

`get_queryset()` (`shoop.admin.modules.categories.views.CategoryListView`
method), 50

get_queryset()	(shoop.admin.modules.categories.views.list.CategoriesListView)	(shoop.campaigns.admin_module.form_sets.CatalogConditionFormSet)	method), 49	method), 144
get_queryset()	(shoop.admin.modules.contact_groups.views.ContactGroupListView)	(shoop.campaigns.admin_module.form_sets.CatalogEffectsFormSet)	method), 53	method), 145
get_queryset()	(shoop.admin.modules.contact_groups.views.list.ContactGroupListView)	(shoop.campaigns.admin_module.form_sets.CatalogFiltersFormSet)	method), 52	method), 145
get_queryset()	(shoop.admin.modules.contacts.views.ContactListView)	(shoop.core.api.products.ProductViewSet)	method), 56	method), 180
get_queryset()	(shoop.admin.modules.contacts.views.list.ContactListView)	(shoop.core.api.products.ShopProductViewSet)	method), 56	method), 180
get_queryset()	(shoop.admin.modules.orders.views.list.OrderListView)	(shoop.front.admin_module.carts.views.CartListView)	method), 63	method), 268
get_queryset()	(shoop.admin.modules.orders.views.OrderListView)	(shoop.front.apps.personal_order_history.views.OrderViewMixin)	method), 67	method), 273
get_queryset()	(shoop.admin.modules.product_types.views.list.ProductTypeListView)	(shoop.front.apps.simple_search.views.SearchView)	method), 74	method), 277
get_queryset()	(shoop.admin.modules.product_types.views.ProductTypeListView)	(shoop.front.views.category.CategoryView)	method), 75	method), 292
get_queryset()	(shoop.admin.modules.products.forms.base_forms.BaseForm)	(shoop.front.views.product.ProductDetailFormSet)	method), 77	method), 293
get_queryset()	(shoop.admin.modules.products.views.list.ProductListView)	(shoop.simple_cms.views.PageView)	method), 88	method), 320
get_queryset()	(shoop.admin.modules.products.views.ProductListView)	(shoop.simple_supplier.admin_module.views.StocksListView)	method), 89	method), 322
get_queryset()	(shoop.admin.modules.sales_units.views.list.SalesUnitListView)	MediaImage()	(in module shoop.testing.factories), 331	
get_queryset()	(shoop.admin.modules.sales_units.views.SalesUnitListView)	products()	(in module shoop.front.template_helpers.general), 288	
get_queryset()	(shoop.admin.modules.services.behavior_forget_password_display_form.DisplayForm)	get_participant_type_display()	(shoop.notify.models.Notification)	method), 306
get_queryset()	(shoop.admin.modules.users.views.list.UserListView)	get_recipient_type_display()	(shoop.notify.models.notification.Notification)	method), 305
get_queryset()	(shoop.admin.modules.users.views.password.UserChangePasswordView)	get_reset_password_views()	(in module shoop.addons.reloaders), 42	
get_queryset()	(shoop.admin.modules.users.views.permissions.UserChangePermissionsView)	get_user_change_permissions()	(shoop.addons.admin_module.AddonModule)	method), 41
get_queryset()	(shoop.admin.modules.users.views.UserChangePasswordView)	get_required_permissions()	(shoop.admin.base.AdminModule)	method), 124
get_queryset()	(shoop.admin.modules.users.views.UserListView)	get_required_permissions()	(shoop.admin.modules.attributes.AttributeModule)	method), 48
get_queryset()	(shoop.admin.modules.users.views.UserResetPasswordView)	get_required_permissions()	(shoop.admin.modules.categories.CategoryModule)	method), 50
get_queryset()	(shoop.campaigns.admin_module.form_sets.BaseFormSet)	get_required_permissions()	(shoop.admin.modules.contact_groups.ContactGroupModule)	method), 57
get_queryset()	(shoop.campaigns.admin_module.form_sets.BasketConditionalFormSet)	get_required_permissions()	(shoop.admin.modules.contacts.ContactModule)	method), 57

<code>get_required_permissions()</code> (<code>shoop.admin.modules.manufacturers.ManufacturerModule</code> (<code>shoop.simple_cms.admin_module.SimpleCMSAdminModule</code> method), 60	<code>get_required_permissions()</code> method), 318
<code>get_required_permissions()</code> (<code>shoop.admin.modules.media.MediaModule</code> method), 61	<code>get_required_permissions()</code> (<code>shoop.simple_supplier.admin_module.StocksAdminModule</code> method), 322
<code>get_required_permissions()</code> (<code>shoop.admin.modules.orders.OrderModule</code> method), 71	<code>get_required_permissions()</code> (<code>shoop.testing.admin_module.TestingAdminModule</code> method), 326
<code>get_required_permissions()</code> (<code>shoop.admin.modules.permission_groups.PermissionGroupModule</code> method), 73	<code>get_required_permissions()</code> (<code>shoop.xtheme.admin_module.XthemeAdminModule</code> method), 352
<code>get_required_permissions()</code> (<code>shoop.admin.modules.product_types.ProductTypeModule</code> method), 75	<code>get_resource_container()</code> (in module <code>shoop.xtheme.resources</code>), 370
<code>get_required_permissions()</code> (<code>shoop.admin.modules.products.ProductModule</code> method), 91	<code>get_return_url()</code> (<code>shoop.admin.utils.views.CreateOrUpdateView</code> method), 122
<code>get_required_permissions()</code> (<code>shoop.admin.modules.sales_units.SalesUnitModule</code> method), 92	<code>get_role_display()</code> (<code>shoop.core.models.OrderStatus</code> method), 205
<code>get_required_permissions()</code> (<code>shoop.admin.modules.service_providers.ServiceProviderModule</code> method), 94	<code>get_role_display()</code> (<code>shoop.core.models.SavedAddress</code> method), 223
<code>get_required_permissions()</code> (<code>shoop.admin.modules.services.ServiceModule</code> method), 101	<code>get_root_categories()</code> (in module <code>shoop.front.template_helpers.general</code>), 289
<code>get_required_permissions()</code> (<code>shoop.admin.modules.shops.ShopModule</code> method), 104	<code>get_the_day_block()</code> (in module <code>shoop.admin.modules.orders.dashboard</code>), 69
<code>get_required_permissions()</code> (<code>shoop.admin.modules.suppliers.SupplierModule</code> method), 106	<code>get_save_form_id()</code> (<code>shoop.admin.utils.views.CreateOrUpdateView</code> method), 122
<code>get_required_permissions()</code> (<code>shoop.admin.modules.system.SystemModule</code> method), 107	<code>get_search_form()</code> (<code>shoop.front.apps.simple_search.template_helpers.Temp</code> method), 276
<code>get_required_permissions()</code> (<code>shoop.admin.modules.taxes.TaxModule</code> method), 111	<code>get_search_product_ids()</code> (in module <code>shoop.front.apps.simple_search.views</code>), 276
<code>get_required_permissions()</code> (<code>shoop.admin.modules.users.UserModule</code> method), 116	<code>get_search_query_texts()</code> (<code>shoop.admin.base.MenuEntry</code> method), 125
<code>get_required_permissions()</code> (<code>shoop.campaigns.admin_module.CampaignAdminModule</code> method), 145	<code>get_search_results()</code> (in module <code>shoop.admin.views.search</code>), 123
<code>get_required_permissions()</code> (<code>shoop.default_tax.admin_module.TaxRulesAdminModule</code> method), 267	<code>get_search_results()</code> (<code>shoop.admin.base.AdminModule</code> method), 124
<code>get_required_permissions()</code> (<code>shoop.front.admin_module.CartAdminModule</code> method), 269	<code>get_search_results()</code> (<code>shoop.admin.modules.categories.CategoryModule</code> method), 50
<code>get_required_permissions()</code> (<code>shoop.notify.admin_module.NotifyAdminModule</code> method), 302	<code>get_search_results()</code> (<code>shoop.admin.modules.contacts.ContactModule</code> method), 57
	<code>get_search_results()</code> (<code>shoop.admin.modules.demo.DemoModule</code> method), 58
	<code>get_search_results()</code> (<code>shoop.admin.modules.orders.OrderModule</code> method), 71
	<code>get_search_results()</code> (<code>shoop.admin.modules.products.ProductModule</code> method), 91
	<code>get_search_results()</code> (<code>shoop.admin.modules.users.UserModule</code> method), 116
	<code>get_search_results()</code> (<code>shoop.guide.admin_module.GuideAdminModule</code> method), 296
	<code>get_selected_and_removed()</code> (<code>shoop.admin.modules.contact_groups.views.forms.ContactGroup</code> method), 52

- get_selected_and_removed() (shoop.admin.modules.products.forms.package_form.PackageChildFormSet method), 78
- get_selected_and_removed() (shoop.admin.modules.products.forms.PackageChildFormSet method), 80
- get_selected_and_unlinked() (shoop.admin.modules.products.forms.simple_variation_form.SimpleVariationChildFormSet method), 79
- get_selected_and_unlinked() (shoop.admin.modules.products.forms.SimpleVariationChildFormSet method), 82
- get_selected_reload_method() (shoop.addons.admin_module.views.reload.ReloadMethodForm method), 38
- get_serialized_steps() (shoop.notify.models.Script method), 306
- get_serialized_steps() (shoop.notify.models.script.Script method), 305
- get_service_choices() (shoop.core.models.CustomCarrier method), 195
- get_service_choices() (shoop.core.models.CustomPaymentProcessor method), 196
- get_service_choices() (shoop.core.models.ServiceProvider method), 227
- get_service_choices() (shoop.testing.models.PseudoPaymentProcessor method), 329
- get_service_provider() (shoop.admin.modules.services.forms.BaseMethodForm method), 97
- get_setting() (shoop.xtheme.models.ThemeSettings method), 366
- get_setting() (shoop.xtheme.Theme method), 375
- get_settings() (shoop.xtheme.models.ThemeSettings method), 366
- get_settings() (shoop.xtheme.Theme method), 376
- get_shipping_method() (in module shoop.testing.factories), 331
- get_shipping_mode_display() (shoop.core.models.Product method), 212, 216
- get_shipping_status_display() (shoop.core.models.Order method), 201
- get_shipping_time() (shoop.core.models.ShippingMethod method), 228
- get_shop() (in module shoop.testing.factories), 331
- get_shop_group_field() (shoop.customer_group_pricing.admin_form_group_pricing.SimpleSupplierModule method), 264
- get_shop_instance() (shoop.admin.modules.products.views.get_shop_instance method), 83
- get_shop_instance() (shoop.core.models.Product method), 212, 216
- get_status_display() (shoop.core.models.Category method), 190
- get_status_display() (shoop.core.models.Order method), 201
- get_status_display() (shoop.core.models.Product method), 221
- get_status_display() (shoop.core.models.SavedAddress method), 223
- get_status_display() (shoop.core.models.Shipment method), 228
- get_status_display() (shoop.core.models.SimpleVariationChildFormSet method), 230
- get_status_display() (shoop.xtheme.models.SavedViewConfig method), 366
- get_steps() (shoop.notify.models.Script method), 306
- get_steps() (shoop.notify.models.script.Script method), 306
- get_stock_adjustment_div() (in module shoop.simple_supplier.utils), 324
- get_stock_adjustment_form() (shoop.simple_supplier.admin_module.forms.SimpleSupplierForm method), 321
- get_stock_adjustment_form() (shoop.simple_supplier.admin_module.views.StocksListView method), 322
- get_stock_behavior_display() (shoop.core.models.Product method), 212, 216
- get_stock_information() (shoop.simple_supplier.admin_module.forms.SimpleSupplierForm method), 321
- get_stock_information() (shoop.simple_supplier.admin_module.views.StocksListView method), 322
- get_stock_information_div_id() (in module shoop.simple_supplier.utils), 324
- get_stock_information_html() (in module shoop.simple_supplier.utils), 324
- get_stock_status() (shoop.core.models.Supplier method), 233
- get_stock_status() (shoop.core.suppliers.base.BaseSupplierModule method), 251
- get_stock_status() (shoop.core.suppliers.BaseSupplierModule method), 251
- get_stock_statuses() (shoop.core.models.Supplier method), 233
- get_stock_statuses() (shoop.core.suppliers.base.BaseSupplierModule method), 251
- get_stock_statuses() (shoop.core.suppliers.BaseSupplierModule method), 252
- get_stock_statuses() (shoop.core.suppliers.SimpleSupplierModule method), 324
- get_stock_status_display() (shoop.core.models.Product method), 212, 216
- get_string_sort_order() (in module shoop.utils.numbers), 344
- get_subtitle() (in module shoop.admin.modules.orders.dashboard), 69
- get_success_url() (shoop.admin.modules.contacts.views.ContactResetPassword method), 264

method), 57
get_success_url() (shoop.admin.modules.contacts.views.reset_password.ContactResetPasswordView method), 56
get_success_url() (shoop.admin.modules.products.views.edit_product.ProductEditView method), 87
get_success_url() (shoop.admin.modules.service_providers.views.supplier.ServiceProviderEditView method), 93
get_success_url() (shoop.admin.modules.users.views.detail.UserDetailView method), 112
get_success_url() (shoop.admin.modules.users.views.password.UserChangePasswordView method), 113
get_success_url() (shoop.admin.modules.users.views.password.UserResetPasswordView method), 113
get_success_url() (shoop.admin.modules.users.views.permissions.UserChangePermissionsView method), 114
get_success_url() (shoop.admin.modules.users.views.UserChangePasswordView method), 115
get_success_url() (shoop.admin.modules.users.views.UserChangePermissionsView method), 115
get_success_url() (shoop.admin.modules.users.views.UserDetailView method), 114
get_success_url() (shoop.admin.modules.users.views.UserResetPasswordView method), 115
get_success_url() (shoop.admin.utils.views.CreateOrUpdateView method), 122
get_success_url() (shoop.front.apps.auth.views.LoginView method), 270
get_success_url() (shoop.front.apps.registration.views.ActiveView method), 274
get_success_url() (shoop.front.apps.registration.views.RegistrationView method), 274
get_success_url() (shoop.front.checkout.CheckoutPhaseView method), 285
get_success_url() (shoop.xtheme.admin_module.views.ThemeConfigDetailView method), 352
get_suppliable_products() (shoop.core.models.Supplier method), 234
get_suppliers() (shoop.simple_supplier.admin_module.forms.SimpleSupplierForm method), 321
get_target_user() (shoop.front.apps.auth.views.RecoverPasswordView method), 271
get_tax() (in module shoop.testing.factories), 331
get_tax_class_proportions() (in module shoop.core.taxing.utils), 252
get_tax_module() (in module shoop.core.taxing), 255
get_tax_summary() (shoop.core.models.Order method), 202
get_taxed_price() (shoop.core.taxing.TaxModule method), 253
get_taxed_price() (shoop.default_tax.module.DefaultTaxModule method), 267
get_taxed_price_for() (shoop.core.taxing.TaxModule method), 254
get_taxes_of_effective_rules() (in module shoop.default_tax.module), 267
get_contact_reset_password_view() (shoop.core.telemetry), 263
get_template() (shoop.notify.base.Action method), 308
get_template_values() (shoop.notify.Action method), 313
get_test_tax() (in module shoop.testing.factories), 331
get_user_reset_password_view() (shoop.admin_module.views.ThemeConfigDetailView method), 352
get_toolbar() (shoop.admin.modules.contacts.views.ContactGroupDetailView method), 53
get_toolbar() (shoop.admin.modules.contacts.views.edit.ContactGroupDetailView method), 51
get_toolbar() (shoop.admin.modules.contacts.views.ContactEditView method), 57
get_toolbar() (shoop.admin.modules.contacts.views.edit.ContactEditView method), 55
get_toolbar() (shoop.admin.modules.orders.views.detail.OrderDetailView method), 61
get_toolbar() (shoop.admin.modules.orders.views.edit.OrderEditView method), 62
get_toolbar() (shoop.admin.modules.orders.views.OrderDetailView method), 66
get_toolbar() (shoop.admin.modules.orders.views.OrderEditView method), 67
get_toolbar() (shoop.admin.modules.products.views.edit.ProductEditView method), 84
get_toolbar() (shoop.admin.modules.products.views.ProductEditView method), 89
get_toolbar() (shoop.admin.modules.service_providers.views.ServiceProviderEditView method), 93
get_toolbar() (shoop.admin.modules.shops.views.edit.ShopEditView method), 103
get_toolbar() (shoop.admin.modules.shops.views.list.ShopListView method), 104
get_toolbar() (shoop.admin.modules.shops.views.ShopEditView method), 104
get_toolbar() (shoop.admin.modules.shops.views.ShopListView method), 104

method), 104

get_toolbar() (shoop.admin.modules.suppliers.views.list.SupplierListView method), 308

method), 105

get_toolbar() (shoop.admin.modules.suppliers.views.SupplierListView method), 106

get_toolbar() (shoop.admin.modules.users.views.detail.UserDetailView method), 112

get_toolbar() (shoop.admin.modules.users.views.password.UserChangePasswordView method), 113

get_toolbar() (shoop.admin.modules.users.views.permissions.UserChangePermissionsView method), 114

get_toolbar() (shoop.admin.modules.users.views.UserChangePasswordView method), 115

get_toolbar() (shoop.admin.modules.users.views.UserChangePermissionsView method), 115

get_toolbar() (shoop.admin.modules.users.views.UserDetailView method), 114

get_toolbar() (shoop.admin.utils.views.CreateOrUpdateView method), 122

get_toolbar() (shoop.admin.utils.views.PicotableView method), 123

get_toolbar() (shoop.notify.admin_module.views.list.ScriptListView method), 300

get_toolbar() (shoop.notify.admin_module.views.ScriptListView method), 301

get_total_cost() (shoop.core.models.Service method), 224

get_total_paid_amount() (shoop.core.models.Order method), 202

get_total_refunded_amount() (shoop.core.models.Order method), 202

get_total_sales() (in module shoop.campaigns.utils), 178

get_total_unpaid_amount() (shoop.core.models.Order method), 202

get_total_unrefunded_amount() (shoop.core.models.Order method), 202

get_tracking_codes() (shoop.core.models.Order method), 202

get_translated_value() (shoop.xtheme.Plugin method), 374

get_type_display() (shoop.admin.modules.contacts.views.ContactListView method), 56

get_type_display() (shoop.admin.modules.contacts.views.list.ContactListView method), 56

get_type_display() (shoop.admin.modules.service_providers.views.ServiceProviderListView method), 93

get_type_display() (shoop.core.models.Attribute method), 188

get_type_display() (shoop.core.models.OrderLine method), 203

get_type_display() (shoop.core.models.ProductCrossSell method), 218

get_type_display() (shoop.core.models.Supplier method), 234

get_ui_info_map() (shoop.notify.base.ScriptItem class method), 308

get_unavailability_reasons() (shoop.core.models.GroupAvailabilityBehaviorComponent method), 194

get_unavailability_reasons() (shoop.core.models.ServiceAvailabilityBehaviorComponent method), 224

get_unavailability_reasons() (shoop.core.models.ServiceBehaviorComponent method), 235

get_unavailability_reasons() (shoop.core.models.StaffOnlyBehaviorComponent method), 232

get_unavailability_reasons() (shoop.core.models.WeightBasedPricingBehaviorComponent method), 237

get_unavailability_reasons() (shoop.core.models.WeightLimitsBehaviorComponent method), 237

get_unavailability_reasons() (shoop.testing.models.ExpensiveSwedenBehaviorComponent method), 327

get_unfinalized_cart_block() (in module shoop.front.admin_module), 269

get_unshipped_products() (shoop.core.models.Order method), 202

get_url() (in module shoop.front.template_helpers.urls), 289

get_urls() (in module shoop.admin.urls), 130

get_urls() (shoop.addons.admin_module.AddonModule method), 40

get_urls() (shoop.admin.base.AdminModule method), 124

get_urls() (shoop.admin.modules.attributes.AttributeModule method), 48

get_urls() (shoop.admin.modules.categories.CategoryModule method), 50

get_urls() (shoop.admin.modules.contact_groups.ContactGroupModule method), 53

get_urls() (shoop.admin.modules.contacts.ContactModule method), 57

get_urls() (shoop.admin.modules.demo.DemoModule method), 58

get_urls() (shoop.admin.modules.manufacturers.ManufacturerModule method), 60

get_urls() (shoop.admin.modules.media.MediaModule method), 61

get_urls() (shoop.admin.modules.orders.OrderModule method), 71

get_urls() (shoop.admin.modules.permission_groups.PermissionGroupModule method), 73

get_urls() (shoop.admin.modules.product_types.ProductTypeModule method), 75

get_urls() (shoop.admin.modules.products.ProductModule method), 75

method), 91

get_urls() (shoop.admin.modules.sales_units.SalesUnitModule method), 92

get_urls() (shoop.admin.modules.service_providers.ServiceProviderModule method), 94

get_urls() (shoop.admin.modules.services.ServiceModule method), 101

get_urls() (shoop.admin.modules.shops.ShopModule method), 104

get_urls() (shoop.admin.modules.suppliers.SupplierModule method), 106

get_urls() (shoop.admin.modules.system.SystemModule method), 107

get_urls() (shoop.admin.modules.taxes.TaxModule method), 111

get_urls() (shoop.admin.modules.users.UserModule method), 116

get_urls() (shoop.campaigns.admin_module.CampaignAdminModule method), 145

get_urls() (shoop.default_tax.admin_module.TaxRulesAdminModule method), 266

get_urls() (shoop.front.admin_module.CartAdminModule method), 269

get_urls() (shoop.notify.admin_module.NotifyAdminModule method), 302

get_urls() (shoop.order_printouts.admin_module.PrintoutsAdminModule method), 316

get_urls() (shoop.simple cms.admin_module.SimpleCMSAdminModule method), 318

get_urls() (shoop.simple_supplier.admin_module.StocksAdminModule method), 322

get_urls() (shoop.testing.admin_module.TestingAdminModule method), 326

get_urls() (shoop.xtheme.admin_module.XthemeAdminModule method), 352

get_usages() (shoop.campaigns.admin_module.views.CouponsAdminModule method), 141

get_validation_errors() (shoop.core.order_creator.OrderSource method), 240

get_validation_errors() (shoop.front.basket.objects.BaseBasket method), 280

get_value() (shoop.notify.base.Binding method), 307

get_value() (shoop.notify.base.ScriptItem method), 307

get_value() (shoop.notify.base.TemplatedBinding method), 307

get_value() (shoop.notify.Binding method), 314

get_values() (shoop.notify.base.ScriptItem method), 308

get_variable_data() (shoop.admin.modules.products.forms.variables.VariableForm method), 79

get_variable_data() (shoop.admin.modules.products.forms.variables.VariableForm method), 82

get_variables() (shoop.notify.Context method), 314

get_variables() (shoop.notify.script.Context method), 310

get_variation_siblings() (shoop.core.models.Product method), 212, 216

get_vat_prefix_for_country() (in module shoop.core.utils.vat), 260

get_view_name_plural() (shoop.admin.utils.picotable.Picotable method), 120

get_version() (shoop.core.cache.impl.VersionedCache method), 181

get_version() (shoop.core.cache.VersionedCache method), 182

get_viable_reload_methods() (shoop.addons.admin_module.views.reload.ReloadMethodForm method), 38

get_view() (shoop.themes.classic_gray.ClassicGrayTheme method), 337

get_view() (shoop.xtheme.Theme method), 376

get_view_by_name() (in module shoop.xtheme.views.extra), 358

get_view_config() (in module shoop.xtheme.rendering), 369

get_view_name() (shoop.xtheme.template_ns.XthemeNamespace method), 371

get_visibility_display() (shoop.core.models.Category method), 190

get_visibility_errors() (shoop.core.models.ShopProduct method), 231

get_visibility_limit_display() (shoop.core.models.ShopProduct method), 231

get_visibility_mode_display() (shoop.core.models.Attribute method), 188

get_visible_attributes() (in module shoop.front.template_helpers.product), 289

get_visible_pages() (shoop.simple cms.template_helpers.SimpleCMSTemplate method), 320

get_visible_products() (in module shoop.front.template_helpers.general), 288

getlist() (shoop.utils.setup.Setup method), 349

global_placeholders (shoop.xtheme.Theme attribute), 376

group (shoop.campaigns.models.contact_group_sales_ranges.ContactGroup attribute), 164

group (shoop.campaigns.models.ContactGroupSalesRange attribute), 175

group (shoop.customer_group_pricing.models.CgpPrice attribute), 265

group_by_period() (in module shoop.core.utils.query), 259

GroupAvailabilityBehaviorComponent in shoop.core.models, 194

GroupAvailabilityBehaviorComponent in shoop.core.models, 225

GroupAvailabilityBehaviorComponent.DoesNotExist, 194

GroupAvailabilityBehaviorComponent.MultipleObjectsReturned, (shoop.admin.modules.orders.views.OrderEditView
 194 method), 67
 groupavailabilitybehaviorcomponent_set handle_del() (in module shoop.front.basket.commands),
 (shoop.core.models.ContactGroup attribute), 278
 194 handle_finalize() (shoop.admin.modules.orders.views.edit.OrderEditView
 GroupAvailabilityBehaviorComponentForm (class in method), 62
 shoop.admin.modules.services.forms), 99 handle_finalize() (shoop.admin.modules.orders.views.OrderEditView
 GroupAvailabilityBehaviorComponentForm.Meta (class method), 67
 in shoop.admin.modules.services.forms), 99 handle_get_data() (shoop.notify.admin_module.views.editor.ScriptAPI
 groups (shoop.core.models.AnonymousContact at- method), 299
 tribute), 187 handle_get_folder() (shoop.admin.modules.media.views.MediaBrowserView
 groups (shoop.core.models.Contact attribute), 193 method), 60
 groups (shoop.core.models.GroupAvailabilityBehaviorComponent handle_get_folders() (shoop.admin.modules.media.views.MediaBrowserView
 attribute), 194 method), 60
 GuideAdminModule (class in handle_post_delete_file()
 shoop.guide.admin_module), 296 (shoop.admin.modules.media.views.MediaBrowserView
 method), 61
 UnicornReloadMethod (class in shoop.addons.reloader, handle_post_delete_folder()
 42 (shoop.admin.modules.media.views.MediaBrowserView
 method), 60
H handle_post_new_file()
 handle() (shoop.core.management.commands.compute_bought_and_sold_items.Command (shoop.admin.modules.media.views.MediaBrowserView
 method), 185 method), 61
 handle() (shoop.core.management.commands.makemessages.Command handle_post_new_folder()
 method), 185 (shoop.admin.modules.media.views.MediaBrowserView
 method), 60
 handle() (shoop.core.management.commands.shoop_init.Command (shoop.admin.modules.media.views.MediaBrowserView
 method), 186 method), 60
 handle() (shoop.core.management.commands.shoop_makemessages.Command handle_post_rename_file()
 method), 186 (shoop.admin.modules.media.views.MediaBrowserView
 method), 60
 handle() (shoop.core.management.commands.shoop_show_settings.Command handle_post_rename_folder()
 method), 187 (shoop.admin.modules.media.views.MediaBrowserView
 method), 60
 handle() (shoop.front.basket.command_dispatcher.BasketCommandDispatcher handle_post_upload_data()
 method), 277 (shoop.admin.modules.orders.views.edit.OrderEditView
 method), 62
 handle() (shoop.testing.management.commands.shoop_populate_products.Command handle_product_data() (shoop.admin.modules.orders.views.OrderEditView
 method), 326 method), 67
 handle_add() (in module shoop.front.basket.commands), handle_save_data() (shoop.notify.admin_module.views.editor.ScriptAPI
 278 method), 299
 handle_add_campaign_code() (in module handle_source_data() (shoop.admin.modules.orders.views.edit.OrderEditView
 shoop.front.basket.commands), 278 method), 62
 handle_add_var() (in module handle_source_data() (shoop.admin.modules.orders.views.OrderEditView
 shoop.front.basket.commands), 278 method), 67
 handle_clear() (in module handle_update() (in module
 shoop.front.basket.commands), 278 shoop.front.basket.commands), 278
 handle_command() (in module handle_upload() (shoop.admin.modules.media.views.MediaBrowserView
 shoop.xtheme.views.command), 357 method), 60
 handle_custom_payment_return_requests() (in module has_bought_and_sold() (shoop.campaigns.models.campaigns.Coupon
 shoop.admin.modules.orders.receivers), 70 method), 160
 handle_customer_data() (shoop.admin.modules.orders.views.OrderEditView has_bought_and_sold() (shoop.campaigns.models.Coupon
 method), 62 method), 177
 handle_customer_data() (shoop.admin.modules.orders.views.OrderEditView has_changed() (shoop.admin.modules.services.forms.AlwaysChangedModel
 method), 67 method), 97
 handle_customer_details() has_language() (shoop.notify.template.Template
 (shoop.admin.modules.orders.views.edit.OrderEditView method), 311
 method), 62
 handle_customer_details()

- has_refunds() (shoop.core.models.Order method), 202
- has_shippable_lines() (shoop.core.order_creator.OrderSource method), 240
- has_url() (in module shoop.front.template_helpers.urls), 289
- help (shoop.core.management.commands.shoop_makemessages method), 186
- help (shoop.core.management.commands.shoop_show_settings method), 186
- help_text (shoop.core.models.FixedCostBehaviorComponent attribute), 196
- help_text (shoop.core.models.GroupAvailabilityBehaviorComponent attribute), 194
- help_text (shoop.core.models.RoundingBehaviorComponent attribute), 222
- help_text (shoop.core.models.ServiceBehaviorComponent attribute), 225
- help_text (shoop.core.models.StaffOnlyBehaviorComponent attribute), 232
- help_text (shoop.core.models.WaivingCostBehaviorComponent attribute), 236
- help_text (shoop.core.models.WeightBasedPricingBehaviorComponent attribute), 237
- help_text (shoop.core.models.WeightLimitsBehaviorComponent attribute), 237
- help_texts (shoop.admin.modules.services.forms.PaymentMethodForm attribute), 98
- help_texts (shoop.admin.modules.services.forms.ShippingMethodForm attribute), 98
- help_texts (shoop.campaigns.admin_module.form_parts.SalesRangesForm attribute), 142
- help_texts (shoop.default_tax.admin_module.views.TaxRuleForm attribute), 266
- HelpersNamespace (class in shoop.front.templatetags.shoop_front), 290
- HIDDEN (shoop.core.models.AttributeVisibility attribute), 189
- hide_cell_class_template (shoop.xtheme.layout.Layout attribute), 363
- hide_prices (shoop.core.pricing.PriceDisplayOptions attribute), 245
- HIGH (shoop.notify.enums.Priority attribute), 310
- I
- icon (shoop.admin.base.OrderSection attribute), 126
- icon (shoop.admin.modules.orders.sections.ContentsOrderSection attribute), 71
- icon (shoop.admin.modules.orders.sections.LogEntriesOrderSection attribute), 71
- icon (shoop.admin.modules.orders.sections.PaymentOrderSection attribute), 71
- icon_classes (shoop.xtheme.plugins.social_media_links.SocialMediaLinksPlugin attribute), 355
- id (shoop.core.models.AnonymousContact attribute), 187
- id (shoop.core.models.Counter attribute), 194
- id_for_label() (shoop.xtheme.plugins.widgets.TranslatableFieldWidget method), 356
- identifier (shoop.addons.reloader.DevServerReloadMethod attribute), 41
- identifier (shoop.addons.reloader.GunicornReloadMethod attribute), 42
- identifier (shoop.addons.reloader.ModWSGIReloadMethod attribute), 42
- identifier (shoop.addons.reloader.ReloadMethod attribute), 41
- identifier (shoop.addons.reloader.UwsgiReloadMethod attribute), 41
- identifier (shoop.admin.base.OrderSection attribute), 126
- identifier (shoop.admin.modules.orders.sections.ContentsOrderSection attribute), 71
- identifier (shoop.admin.modules.orders.sections.LogEntriesOrderSection attribute), 71
- identifier (shoop.admin.modules.orders.sections.PaymentOrderSection attribute), 71
- identifier (shoop.campaigns.models.basket_conditions.BasketMaxTotalAmount attribute), 149
- identifier (shoop.campaigns.models.basket_conditions.BasketMaxTotalProduct attribute), 148
- identifier (shoop.campaigns.models.basket_conditions.BasketTotalAmount attribute), 147
- identifier (shoop.campaigns.models.basket_conditions.BasketTotalProduct attribute), 146
- identifier (shoop.campaigns.models.basket_conditions.ContactBasketCondition attribute), 152
- identifier (shoop.campaigns.models.basket_conditions.ContactGroupBasketCondition attribute), 151
- identifier (shoop.campaigns.models.basket_conditions.ProductsInBasketCondition attribute), 150
- identifier (shoop.campaigns.models.basket_effects.BasketDiscountAmount attribute), 153
- identifier (shoop.campaigns.models.basket_effects.BasketDiscountEffect attribute), 152
- identifier (shoop.campaigns.models.basket_effects.BasketDiscountPercentage attribute), 154
- identifier (shoop.campaigns.models.basket_line_effects.BasketLineEffect attribute), 155
- identifier (shoop.campaigns.models.basket_line_effects.DiscountFromProduct attribute), 156
- identifier (shoop.campaigns.models.basket_line_effects.FreeProductLine attribute), 155
- identifier (shoop.campaigns.models.BasketDiscountEffect attribute), 172
- identifier (shoop.campaigns.models.BasketLineEffect attribute), 170
- identifier (shoop.campaigns.models.catalog_filters.CatalogFilter attribute), 161
- identifier (shoop.campaigns.models.catalog_filters.CategoryFilter attribute), 163

- identifier (shoop.campaigns.models.catalog_filters.ProductFilter attribute), 285
- attribute), 163
- identifier (shoop.campaigns.models.catalog_filters.ProductTypeFilter attribute), 294
- attribute), 162
- identifier (shoop.campaigns.models.CatalogFilter attribute), 175
- identifier (shoop.campaigns.models.context_conditions.ContextCondition attribute), 166
- identifier (shoop.campaigns.models.context_conditions.ContextCondition attribute), 166
- identifier (shoop.campaigns.models.context_conditions.ContextCondition attribute), 165
- identifier (shoop.campaigns.models.ContextCondition attribute), 176
- identifier (shoop.campaigns.models.product_effects.ProductDiscountEffect attribute), 168
- identifier (shoop.campaigns.models.product_effects.ProductDiscountEffect attribute), 167
- identifier (shoop.campaigns.models.product_effects.ProductDiscountPercentage attribute), 169
- identifier (shoop.campaigns.models.ProductDiscountEffect attribute), 174
- identifier (shoop.campaigns.modules.BasketCampaignModule attribute), 178
- identifier (shoop.campaigns.modules.CatalogCampaignModule attribute), 178
- identifier (shoop.core.pricing.default_pricing.DefaultPricingModule attribute), 242
- identifier (shoop.core.pricing.PricingModule attribute), 250
- identifier (shoop.core.suppliers.base.BaseSupplierModule attribute), 251
- identifier (shoop.core.suppliers.BaseSupplierModule attribute), 252
- identifier (shoop.core.taxing.TaxModule attribute), 254
- identifier (shoop.customer_group_pricing.module.CustomerGroupPricingModule attribute), 265
- identifier (shoop.default_tax.module.DefaultTaxModule attribute), 267
- identifier (shoop.front.checkout.addresses.AddressesPhase attribute), 282
- identifier (shoop.front.checkout.CheckoutPhaseViewMixin attribute), 285
- identifier (shoop.front.checkout.confirm.ConfirmPhase attribute), 283
- identifier (shoop.front.checkout.empty.EmptyPhase attribute), 283
- identifier (shoop.front.checkout.methods.MethodsPhase attribute), 284
- identifier (shoop.front.checkout.methods.PaymentMethodPhase attribute), 284
- identifier (shoop.front.checkout.methods.ShippingMethodPhase attribute), 284
- identifier (shoop.front.checkout.single_page.SingleCheckoutPhase attribute), 285
- identifier (shoop.front.notify_events.OrderReceived attribute), 314
- attribute), 294
- identifier (shoop.notify.Action attribute), 314
- identifier (shoop.notify.actions.AddNotification attribute), 298
- identifier (shoop.notify.actions.AddOrderLogEntry attribute), 298
- identifier (shoop.notify.actions.debug.SetDebugFlag attribute), 297
- identifier (shoop.notify.actions.email.SendEmail attribute), 297
- identifier (shoop.notify.actions.notification.AddNotification attribute), 297
- identifier (shoop.notify.actions.order.AddOrderLogEntry attribute), 298
- identifier (shoop.notify.actions.SendEmail attribute), 298
- identifier (shoop.notify.actions.SetDebugFlag attribute), 298
- identifier (shoop.notify.base.Action attribute), 309
- identifier (shoop.notify.base.Base attribute), 307
- identifier (shoop.notify.base.Condition attribute), 308
- identifier (shoop.notify.base.Event attribute), 307
- identifier (shoop.notify.base.ScriptItem attribute), 308
- identifier (shoop.notify.Condition attribute), 315
- identifier (shoop.notify.conditions.BooleanEqual attribute), 303
- identifier (shoop.notify.conditions.Empty attribute), 303
- identifier (shoop.notify.conditions.IntegerEqual attribute), 303
- identifier (shoop.notify.conditions.LanguageEqual attribute), 303
- identifier (shoop.notify.conditions.NonEmpty attribute), 304
- identifier (shoop.notify.conditions.simple.BaseEqual attribute), 302
- identifier (shoop.notify.conditions.simple.CaseInsensitiveStringEqual attribute), 303
- identifier (shoop.notify.conditions.simple.Empty attribute), 302
- identifier (shoop.notify.conditions.simple.NonEmpty attribute), 302
- identifier (shoop.notify.conditions.TextEqual attribute), 304
- identifier (shoop.notify.Event attribute), 315
- identifier (shoop.notify.typology.Boolean attribute), 312
- identifier (shoop.notify.typology.Decimal attribute), 312
- identifier (shoop.notify.typology.Email attribute), 312
- identifier (shoop.notify.typology.Enum attribute), 313
- identifier (shoop.notify.typology.Integer attribute), 312
- identifier (shoop.notify.typology.Language attribute), 312
- identifier (shoop.notify.typology.Model attribute), 313
- identifier (shoop.notify.typology.Phone attribute), 313
- identifier (shoop.notify.typology.Text attribute), 312

- identifier (shoop.notify.typology.Type attribute), [311](#)
- identifier (shoop.notify.typology.URL attribute), [313](#)
- identifier (shoop.simple_cms.plugins.PageLinksPlugin attribute), [319](#)
- identifier (shoop.simple_supplier.module.SimpleSupplierModel attribute), [324](#)
- identifier (shoop.testing.factories.CategoryFactory attribute), [330](#)
- identifier (shoop.testing.factories.ProductTypeFactory attribute), [329](#)
- identifier (shoop.testing.simple_checkout_phase.TestPaymentCheckoutPhase attribute), [335](#)
- identifier (shoop.testing.simple_checkout_phase.TestShipmentCheckoutPhase attribute), [335](#)
- identifier (shoop.themes.classic_gray.ClassicGrayTheme attribute), [337](#)
- identifier (shoop.themes.classic_gray.plugins.ProductCrossSellsPlugin attribute), [337](#)
- identifier (shoop.themes.classic_gray.plugins.ProductHighlightPlugin attribute), [337](#)
- identifier (shoop.xtheme.parsing.LayoutPartExtension attribute), [368](#)
- identifier (shoop.xtheme.parsing.PlaceholderExtension attribute), [367](#)
- identifier (shoop.xtheme.parsing.PluginExtension attribute), [368](#)
- identifier (shoop.xtheme.Plugin attribute), [374](#)
- identifier (shoop.xtheme.plugins.category_links.CategoryLinksPlugin attribute), [353](#)
- identifier (shoop.xtheme.plugins.image.ImagePlugin attribute), [354](#)
- identifier (shoop.xtheme.plugins.snippets.SnippetsPlugin attribute), [355](#)
- identifier (shoop.xtheme.plugins.social_media_links.SocialMediaLinksPlugin attribute), [355](#)
- identifier (shoop.xtheme.plugins.text.TextPlugin attribute), [356](#)
- identifier (shoop.xtheme.Theme attribute), [376](#)
- identifier_attr (shoop.core.models.ShoopModel attribute), [229](#)
- identifier_attr (shoop.core.models.Tax attribute), [235](#)
- identifier_suffix (shoop.notify.conditions.simple.BaseEqual attribute), [302](#)
- identifier_suffix (shoop.notify.conditions.simple.CaseInsensitiveString attribute), [303](#)
- identifiyfy() (in module shoop.utils.text), [350](#)
- image (shoop.core.models.Category attribute), [190](#)
- IMAGE (shoop.core.models.ProductMediaKind attribute), [220](#)
- ImageChoiceWidget (class in shoop.admin.forms.widgets), [45](#)
- ImageIDField (class in shoop.xtheme.plugins.image), [354](#)
- ImagePlugin (class in shoop.xtheme.plugins.image), [354](#)
- ImagePluginChoiceWidget (class in shoop.xtheme.plugins.image), [354](#)
- images (shoop.core.models.ShopProduct attribute), [231](#)
- ImmutabilityError, [260](#)
- ImmutableAddress (class in shoop.core.models), [197](#)
- ImmutableAddress.DoesNotExist, [197](#)
- ImmutableAddress.MultipleObjectsReturned, [197](#)
- ImpossibleProductModeException, [260](#)
- includes_tax (shoop.core.pricing.Price attribute), [245](#)
- includes_tax (shoop.core.pricing.TaxfulPrice attribute), [245](#)
- includes_tax (shoop.core.pricing.TaxlessPrice attribute), [245](#)
- includes_tax (shoop.simple_supplier.models.StockAdjustment attribute), [323](#)
- includes_tax (shoop.simple_supplier.models.StockCount attribute), [323](#)
- increase_customer_usage_limit_by() (shoop.campaigns.models.campaigns.Coupon method), [160](#)
- increase_customer_usage_limit_by() (shoop.campaigns.models.Coupon method), [177](#)
- increase_usage_limit_by() (shoop.campaigns.models.campaigns.Coupon method), [160](#)
- increase_usage_limit_by() (shoop.campaigns.models.campaigns.Coupon method), [177](#)
- IndexView (class in shoop.front.views.index), [292](#)
- infer() (shoop.admin.breadcrumbs.Breadcrumbs class method), [126](#)
- inherited_variables (shoop.xtheme.TemplatedPlugin attribute), [375](#)
- init_search_fields() (shoop.admin.views.select.MultiselectAjaxView method), [124](#)
- INITIAL (shoop.core.models.OrderStatusRole attribute), [205](#)
- Initializer (class in shoop.core.management.commands.shoop_init), [186](#)
- inject_resources() (in module shoop.xtheme.resources), [370](#)
- InlineMarkupResource (class in shoop.xtheme.resources), [370](#)
- InlineScriptResource (class in shoop.xtheme.resources), [369](#)
- insert_row() (shoop.xtheme.layout.Layout method), [364](#)
- install_error_handlers() (in module shoop.front.error_handling), [293](#)
- install_package() (shoop.addons.installer.PackageInstaller method), [41](#)
- instantiate() (shoop.utils.form_group.FormDef method), [341](#)
- instantiate_forms() (shoop.utils.form_group.FormGroup method), [341](#)

- method), 341
- instantiate_phase_class()
 - (shoop.front.checkout.CheckoutProcess method), 286
- instantiate_plugin() (shoop.xtheme.layout.LayoutCell method), 362
- Integer (class in shoop.notify.typology), 312
- INTEGER (shoop.core.models.AttributeType attribute), 189
- IntegerEqual (class in shoop.notify.conditions), 303
- INTERNAL (shoop.core.models.SupplierType attribute), 234
- InternalIdentifierField (class in shoop.core.fields), 184
- invalid_modes (shoop.admin.modules.products.views.edit_attribute), 86
- invalid_modes (shoop.admin.modules.products.views.edit_attribute), 87
- invalid_modes (shoop.admin.modules.products.views.edit_attribute), 88
- INVISIBLE (shoop.core.models.CategoryStatus attribute), 191
- INVISIBLE (shoop.core.models.ProductVariationLinkStatus attribute), 221
- is_active() (shoop.campaigns.models.contact_group_sales_ranges.ContactGroupSalesRange method), 164
- is_active() (shoop.campaigns.models.ContactGroupSalesRange method), 175
- is_all_seeing (shoop.core.models.Contact attribute), 193
- is_all_seeing (shoop.core.models.PersonContact attribute), 209
- is_anonymous (shoop.core.models.AnonymousContact attribute), 187
- is_anonymous (shoop.core.models.Contact attribute), 193
- is_available() (shoop.campaigns.models.Campaign method), 173
- is_available() (shoop.campaigns.models.campaigns.Campaign method), 157
- is_available_for() (shoop.core.models.Service method), 225
- is_canceled() (shoop.core.models.Order method), 202
- is_code_usable() (in module shoop.core.order_creator), 238
- is_coercible_from() (shoop.notify.typology.Model method), 313
- is_coercible_from() (shoop.notify.typology.Text method), 312
- is_coercible_from() (shoop.notify.typology.Type method), 312
- is_complete() (shoop.core.models.Order method), 202
- is_context_valid() (shoop.xtheme.Plugin method), 374
- is_discounted (shoop.core.pricing.Priceful attribute), 247
- is_edit_mode() (in module shoop.xtheme.editing), 359
- is_edit_mode() (shoop.xtheme.template_ns.XthemeNamespaces method), 371
- is_empty (shoop.core.order_creator.OrderSource attribute), 240
- is_in_grace_period() (in module shoop.core.telemetry), 263
- is_list_visible() (shoop.core.models.ShopProduct method), 231
- is_null_value() (shoop.core.models.Attribute method), 188
- is_numeric (shoop.core.models.Attribute attribute), 188
- is_opt_out() (in module shoop.core.telemetry), 263
- is_orderable() (shoop.core.models.ShopProduct method), 231
- is_package_child() (shoop.core.models.Product method), 212, 216
- is_package_child() (shoop.core.models.ProductChildrenFormPart method), 212, 216
- is_package_parent() (shoop.core.models.Product parent.ProductChildrenBaseFormPart method), 212, 216
- is_paid() (shoop.core.models.Order method), 202
- is_payment_variation_child() (shoop.campaigns.reloader.GunicornReloadMethod method), 42
- is_read (shoop.notify.models.Notification attribute), 306
- is_read (shoop.notify.models.notification.Notification attribute), 304
- is_stocked() (shoop.core.models.Product method), 212, 216
- is_stringy (shoop.core.models.Attribute attribute), 188
- is_telemetry_enabled() (in module shoop.core.telemetry), 263
- is_temporal (shoop.core.models.Attribute attribute), 188
- is_translated (shoop.core.models.Attribute attribute), 188
- is_url_valid() (shoop.simple_cms.admin_module.views.PageForm method), 317
- is_usable() (shoop.campaigns.models.campaigns.Coupon class method), 160
- is_usable() (shoop.campaigns.models.Coupon class method), 177
- is_valid (shoop.admin.modules.orders.json_order_creator.JsonOrderCreator attribute), 70
- is_valid() (shoop.front.checkout.addresses.AddressesPhase method), 283
- is_valid() (shoop.front.checkout.CheckoutPhaseViewMixin method), 286
- is_valid() (shoop.front.checkout.confirm.ConfirmPhase method), 283
- is_valid() (shoop.front.checkout.methods.MethodsPhase method), 284
- is_valid() (shoop.testing.simple_checkout_phase.TestMethodCheckoutPhase method), 335
- is_valid() (shoop.utils.form_group.FormGroup method), 341
- is_valid_key() (shoop.utils.setup.Setup method), 349
- is_variation_child() (shoop.core.models.Product method), 212, 216
- is_variation_parent() (shoop.core.models.Product method), 212, 216

- is_viable() (shoop.addons.reloaders.DevServerReloadMethod label (shoop.front.apps.auth.AuthAppConfig attribute), method), 41 271
- is_viable() (shoop.addons.reloaders.GunicornReloadMethod label (shoop.front.apps.customer_information.AppConfig attribute), method), 42 273
- is_viable() (shoop.addons.reloaders.ModWSGIReloadMethod label (shoop.front.apps.personal_order_history.AppConfig attribute), method), 42 273
- is_viable() (shoop.addons.reloaders.ReloadMethod label (shoop.front.apps.registration.RegistrationAppConfig attribute), method), 41 275
- is_viable() (shoop.addons.reloaders.UwsgiReloadMethod label (shoop.front.apps.simple_order_notification.SimpleOrderNotification attribute), method), 41 276
- is_visible() (in module shoop.front.template_helpers.product), 289 label (shoop.front.apps.simple_search.SimpleSearchAppConfig attribute), 277
- is_visible() (shoop.core.models.Category method), 190 label (shoop.front.ShoopFrontAppConfig attribute), 296
- is_visible() (shoop.core.models.ShopProduct method), 231 label (shoop.order_printouts.AppConfig attribute), 316
- is_visible() (shoop.simple_cms.models.Page method), 318 label (shoop.simple_cms.AppConfig attribute), 321
- is_visible() (shoop.simple_supplier.ShoopSimpleSupplierAppConfig attribute), 325
- isoformat() (in module shoop.core.fields.tagged_json), 183 label (shoop.testing.ShoopTestingAppConfig attribute), 336
- J**
- javascript_catalog_all() (in module shoop.utils.i18n), 343 label (shoop.themes.classic_gray.ClassicGrayThemeAppConfig attribute), 337
- JavaScriptActionButton (class in shoop.admin.toolbar), 128 label (shoop.utils.ShoopUtilsAppConfig attribute), 351
- label_from_instance() (shoop.xtheme.plugins.widgets.XThemeModelChoiceField method), 357
- labels (shoop.admin.modules.services.forms.FixedCostBehaviorComponent attribute), 98
- labels (shoop.admin.modules.services.forms.WaivingCostBehaviorComponent attribute), 99
- join_css_classes() (in module shoop.xtheme.utils), 372
- json() (in module shoop.core.templatetags.shoop_common), 256 labels (shoop.campaigns.admin_module.form_parts.SalesRangesForm.Meta attribute), 142
- JsonOrderCreator (class in shoop.admin.modules.orders.json_order_creator), 70 Language (class in shoop.notify.typology), 312
- LANGUAGE_CODES (shoop.core.fields.LanguageField attribute), 185
- K**
- LanguageEqual (class in shoop.notify.conditions), 303
- LanguageField (class in shoop.core.fields), 185
- last_name (shoop.core.utils.name_mixin.NameMixin attribute), 257
- last_name (shoop.testing.factories.UserFactory attribute), 329
- Layout (class in shoop.xtheme.layout), 363
- LayoutCell (class in shoop.xtheme.layout), 361
- LayoutCellFormGroup (class in shoop.xtheme.views.forms), 359
- LayoutCellGeneralInfoForm (class in shoop.xtheme.views.forms), 358
- LayoutPartExtension (class in shoop.xtheme.parsing), 368
- LayoutRow (class in shoop.xtheme.layout), 362
- leave_locale_alone (shoop.core.management.commands.shoop_init.Command attribute), 186
- limit_method_fields() (shoop.front.checkout.methods.MethodsForm method), 284
- limit_method_fields() (shoop.front.checkout.single_page.OrderForm method), 285
- L**
- label (shoop.admin.ShoopAdminAppConfig attribute), 130
- label (shoop.api.ShoopApiAppConfig attribute), 132
- label (shoop.campaigns.apps.CampaignAppConfig attribute), 177
- label (shoop.core.ShoopCoreAppConfig attribute), 264
- label (shoop.customer_group_pricing.CustomerGroupPricingAppConfig attribute), 265
- label (shoop.default_tax.AppConfig attribute), 268

- line_effects (shoop.campaigns.models.BasketCampaign attribute), 171
- line_effects (shoop.campaigns.models.campaigns.BasketCampaign attribute), 159
- line_type (shoop.core.models.PaymentMethod attribute), 206
- line_type (shoop.core.models.ShippingMethod attribute), 229
- lines (shoop.core.models.Order attribute), 202
- LineTax (class in shoop.core.taxing), 253
- link_to_parent() (shoop.core.models.Product method), 213, 216
- listed (shoop.testing.factories.ShopProductFactory attribute), 330
- load() (in module shoop.utils.importing), 343
- load() (shoop.front.basket.storage.BasketStorage method), 280
- load() (shoop.xtheme.Plugin class method), 374
- load_module() (in module shoop.apps.provides), 132
- load_module_instances() (in module shoop.apps.provides), 133
- load_modules() (in module shoop.apps.provides), 133
- load_variables() (shoop.notify.base.Event method), 307
- load_variables() (shoop.notify.Event method), 315
- log() (shoop.notify.Context method), 314
- log() (shoop.notify.script.Context method), 310
- log_entries (shoop.core.models.Category attribute), 190
- log_entries (shoop.core.models.Order attribute), 202
- log_entries (shoop.core.models.Product attribute), 213, 217
- log_entry_queryset (shoop.notify.Context attribute), 315
- log_entry_queryset (shoop.notify.script.Context attribute), 311
- log_target (shoop.notify.base.Event attribute), 307
- log_target (shoop.notify.Event attribute), 315
- log_target_variable (shoop.notify.base.Event attribute), 307
- log_target_variable (shoop.notify.Event attribute), 315
- LogEntriesOrderSection (class in shoop.admin.modules.orders.sections), 71
- LogEntryKind (class in shoop.utils.analog), 338
- logged_model (shoop.core.models.OrderLogEntry attribute), 205
- login() (in module shoop.admin.urls), 130
- LoginView (class in shoop.front.apps.auth.views), 270
- logo (shoop.core.models.PaymentMethod attribute), 206
- logo (shoop.core.models.Service attribute), 225
- logo (shoop.core.models.ServiceProvider attribute), 227
- logo (shoop.core.models.ShippingMethod attribute), 229
- logo (shoop.core.models.Shop attribute), 230
- LogoutView (class in shoop.front.apps.auth.views), 270
- LOW (shoop.notify.enums.Priority attribute), 310
- M**
- maintenance_message (shoop.core.models.Shop attribute), 230
- maintenance_mode_exempt() (in module shoop.core.utils.maintenance), 257
- make_error_view() (in module shoop.front.error_handling), 293
- make_package() (shoop.core.models.Product method), 213, 217
- MALE (shoop.core.models.Gender attribute), 197
- manipulate_query_string() (in module shoop.admin.utils.urls), 122
- Manufacturer (class in shoop.core.models), 198
- manufacturer (shoop.core.models.Product attribute), 213, 217
- Manufacturer.DoesNotExist, 198
- Manufacturer.MultipleObjectsReturned, 198
- ManufacturerEditView (class in shoop.admin.modules.manufacturers.views), 59
- ManufacturerEditView (class in shoop.admin.modules.manufacturers.views.edit), 59
- ManufacturerForm (class in shoop.admin.modules.manufacturers.views.edit), 58
- ManufacturerForm.Meta (class in shoop.admin.modules.manufacturers.views.edit), 58
- ManufacturerListView (class in shoop.admin.modules.manufacturers.views), 59
- ManufacturerListView (class in shoop.admin.modules.manufacturers.views.list), 59
- ManufacturerModule (class in shoop.admin.modules.manufacturers), 60
- map_relation_type() (in module shoop.front.template_helpers.product), 289
- mark_notification_read_view() (shoop.notify.admin_module.NotifyAdminModule method), 302
- mark_read() (shoop.notify.models.Notification method), 306
- mark_read() (shoop.notify.models.notification.Notification method), 304
- markdown() (in module shoop.front.templatetags.shoop_front), 290
- marked_read_by (shoop.notify.models.Notification attribute), 306
- marked_read_by (shoop.notify.models.notification.Notification attribute), 304
- master (shoop.campaigns.models.campaigns.BasketCampaignTranslation attribute), 161

master (shoop.campaigns.models.campaigns.CatalogCampaignTranslation (in module shoop.admin.utils.picotable), attribute), 161
 master (shoop.simple_cms.models.PageTranslation attribute), 319
 matches() (shoop.campaigns.models.basket_conditions.BasketConditionField (class in shoop.core.fields), 184 method), 145
 matches() (shoop.campaigns.models.basket_conditions.BasketMaxTotalAttributeCondition media (shoop.addons.admin_module.views.list.AddonEnableDisableForm attribute), 149
 matches() (shoop.campaigns.models.basket_conditions.BasketMaxTotalProductAmountCondition media (shoop.addons.admin_module.views.reload.ReloadMethodForm attribute), 148
 matches() (shoop.campaigns.models.basket_conditions.BasketTotalAmountCondition media (shoop.addons.admin_module.views.upload.AddonUploadForm attribute), 147
 matches() (shoop.campaigns.models.basket_conditions.BasketTotalProductAmountCondition media (shoop.admin.forms.ShoopAdminForm attribute), 146
 matches() (shoop.campaigns.models.basket_conditions.ContactBasketAttributeCondition media (shoop.admin.forms.widgets.BasePopupChoiceWidget attribute), 152
 matches() (shoop.campaigns.models.basket_conditions.ContactGroupBasketCondition media (shoop.admin.forms.widgets.ContactChoiceWidget attribute), 151
 matches() (shoop.campaigns.models.basket_conditions.ProductsInBasketCondition media (shoop.admin.forms.widgets.ImageChoiceWidget attribute), 150
 matches() (shoop.campaigns.models.BasketCondition attribute), 45
 matches() (shoop.campaigns.models.context_conditions.ContactCondition attribute), 46
 matches() (shoop.campaigns.models.context_conditions.ContactGroupCondition attribute), 46
 matches() (shoop.campaigns.models.context_conditions.ContextCondition attribute), 46
 matches() (shoop.campaigns.models.ContextCondition attribute), 47
 matches() (shoop.default_tax.models.TaxRule method), 267
 matches() (shoop.utils.patterns.Pattern method), 347
 matches_to_value() (shoop.core.models.WeightBasedPriceRange media (shoop.admin.modules.contact_groups.views.forms.ContactGroupBaseForm attribute), 52
 MAX_DECIMAL_PLACES_FOR_STEP (shoop.core.fields.FormattedDecimalFormField attribute), 184
 max_num (shoop.admin.modules.products.forms.base_forms.BaseProductMediaFormSet media (shoop.admin.modules.orders.views.refund.FullRefundConfirmationForm attribute), 76
 max_num (shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSellFormSet media (shoop.admin.modules.orders.views.refund.RefundForm attribute), 84
 max_num (shoop.admin.modules.products.views.edit_media.ProductMediaFormSet media (shoop.admin.modules.orders.views.shipment.ShipmentForm attribute), 85
 max_num (shoop.admin.modules.services.behavior_form_part.BehaviorFormSet media (shoop.admin.modules.permission_groups.views.edit.PermissionGroupForm attribute), 96
 max_num (shoop.campaigns.admin_module.form_sets.BaseFormset attribute), 72
 max_refundable_amount (shoop.core.models.OrderLine attribute), 203
 may_inject() (in module shoop.xtheme.editing), 359
 may_match_postal_code() (shoop.default_tax.models.TaxRuleQuerySet attribute), 76
 method), 267
 media (shoop.admin.modules.products.forms.base_forms.ProductBaseForm

Index 453

- attribute), 138
- media (shoop.campaigns.admin_module.forms.ContactGroupConditionalForm attribute), 317
- attribute), 138
- media (shoop.campaigns.admin_module.forms.CouponForm attribute), 319
- attribute), 138
- media (shoop.campaigns.admin_module.forms.DiscountFromProductForm attribute), 321
- attribute), 138
- media (shoop.campaigns.admin_module.forms.FreeProductLineForm attribute), 322
- attribute), 139
- media (shoop.campaigns.admin_module.forms.ProductDiscountAmountForm attribute), 325
- attribute), 139
- media (shoop.campaigns.admin_module.forms.ProductDiscountPercentageForm attribute), 334
- attribute), 139
- media (shoop.campaigns.admin_module.forms.ProductFilterForm attribute), 333
- attribute), 139
- media (shoop.campaigns.admin_module.forms.ProductsInBasketConditionalForm attribute), 333
- attribute), 140
- media (shoop.campaigns.admin_module.forms.ProductTypeFilterForm attribute), 334
- attribute), 140
- media (shoop.core.models.Product attribute), 213, 217
- media (shoop.customer_group_pricing.admin_form_part.CustomerGroupPricingForm attribute), 344
- attribute), 264
- media (shoop.default_tax.admin_module.views.TaxRuleForm attribute), 351
- attribute), 266
- media (shoop.front.apps.auth.forms.EmailAuthenticationForm attribute), 361
- attribute), 269
- media (shoop.front.apps.auth.forms.RecoverPasswordForm attribute), 353
- attribute), 270
- media (shoop.front.apps.customer_information.forms.AddressForm attribute), 354
- attribute), 272
- media (shoop.front.apps.customer_information.forms.CompanyContactForm attribute), 354
- attribute), 272
- media (shoop.front.apps.customer_information.forms.PersonalDataForm attribute), 354
- attribute), 272
- media (shoop.front.apps.simple_search.views.SearchForm attribute), 355
- attribute), 276
- media (shoop.front.checkout.addresses.AddressForm attribute), 356
- attribute), 282
- media (shoop.front.checkout.addresses.CompanyForm attribute), 356
- attribute), 282
- media (shoop.front.checkout.confirm.ConfirmForm attribute), 359
- attribute), 283
- media (shoop.front.checkout.methods.MethodsForm attribute), 60
- attribute), 284
- MediaBrowserView (class in shoop.admin.modules.media.views), 60
- media (shoop.front.checkout.methods.MethodWidget attribute), 284
- MediaChoiceWidget (class in shoop.admin.forms.widgets), 45
- media (shoop.front.checkout.single_page.OrderForm attribute), 61
- attribute), 285
- MediaModule (class in shoop.admin.modules.media), 61
- media (shoop.front.views.category.ProductListForm attribute), 191
- attribute), 291
- members (shoop.core.models.CompanyContact attribute), 191
- media (shoop.notify.admin_module.forms.ScriptForm attribute), 301
- members (shoop.core.models.ContactGroup attribute), 194
- attribute), 301
- menu_entry_url (shoop.admin.modules.services.PaymentMethodModule attribute), 102
- media (shoop.notify.admin_module.forms.ScriptItemEditForm attribute), 301
- menu_entry_url (shoop.admin.modules.services.ServiceModule attribute), 102
- media (shoop.simple_cms.admin_module.views.PageForm attribute), 317
- media (shoop.simple_cms.plugins.PageLinksConfigForm attribute), 319
- media (shoop.simple_supplier.admin_module.forms.SimpleSupplierForm attribute), 321
- media (shoop.simple_supplier.forms.StockAdjustmentForm attribute), 322
- media (shoop.testing.admin_module.mock_view.MockerForm attribute), 325
- media (shoop.testing.service_forms.CarrierWithCheckoutPhaseForm attribute), 334
- media (shoop.testing.service_forms.PaymentWithCheckoutPhaseForm attribute), 333
- media (shoop.testing.service_forms.PseudoPaymentProcessorForm attribute), 333
- media (shoop.testing.simple_checkout_phase.TestPaymentCheckoutPhaseForm attribute), 334
- media (shoop.testing.simple_checkout_phase.TestShipmentCheckoutPhaseForm attribute), 334
- media (shoop.xtheme.admin_module.views.ActivationForm attribute), 351
- media (shoop.xtheme.forms.GenericThemeForm attribute), 361
- media (shoop.xtheme.plugins.category_links.CategoryLinksConfigForm attribute), 353
- media (shoop.xtheme.plugins.forms.GenericPluginForm attribute), 354
- media (shoop.xtheme.plugins.forms.PluginForm attribute), 354
- media (shoop.xtheme.plugins.image.ImagePluginChoiceWidget attribute), 354
- media (shoop.xtheme.plugins.social_media_links.SocialMediaLinksPluginForm attribute), 355
- media (shoop.xtheme.plugins.widgets.TranslatableFieldWidget attribute), 356
- media (shoop.xtheme.plugins.widgets.XThemeModelChoiceWidget attribute), 356
- media (shoop.xtheme.views.forms.LayoutCellGeneralInfoForm attribute), 359

- attribute), 101
- menu_entry_url (shoop.admin.modules.services.ShippingMethodModel attribute), 101
- MenuEntry (class in shoop.admin.base), 125
- MenuView (class in shoop.admin.views.menu), 123
- merged_initial_and_data() (in module shoop.utils.forms), 341
- message (shoop.utils.excs.Problem attribute), 339
- MethodChoiceIterator (class in shoop.front.checkout.methods), 284
- MethodsForm (class in shoop.front.checkout.methods), 284
- MethodsPhase (class in shoop.front.checkout.methods), 284
- MethodWidget (class in shoop.front.checkout.methods), 284
- min_num (shoop.admin.modules.products.forms.base_forms.BaseProductFormSet attribute), 76
- min_num (shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSellFormSet attribute), 84
- min_num (shoop.admin.modules.products.views.edit_media.ProductMediaFormSet attribute), 85
- min_num (shoop.admin.modules.services.behavior_form_part.BehaviorFormSet attribute), 96
- min_num (shoop.campaigns.admin_module.form_sets.BaseFormSet attribute), 144
- minimum_price (shoop.core.models.ShopProduct attribute), 231
- mock_company() (shoop.testing.admin_module.mock_view.MockView method), 325
- mock_order() (shoop.testing.admin_module.mock_view.MockView method), 325
- mock_person() (shoop.testing.admin_module.mock_view.MockView method), 325
- MockerForm (class in shoop.testing.admin_module.mock_view), 325
- Mockers (class in shoop.testing.admin_module.mock_view), 325
- mockers (shoop.testing.admin_module.mock_view.MockView attribute), 326
- MockerView (class in shoop.testing.admin_module.mock_view), 325
- mode (shoop.core.models.Product attribute), 213, 217
- mode (shoop.core.models.RoundingBehaviorComponent attribute), 222
- Model (class in shoop.notify.typology), 313
- model (shoop.admin.modules.attributes.views.AttributeEditView attribute), 47
- model (shoop.admin.modules.attributes.views.AttributeListView attribute), 48
- model (shoop.admin.modules.attributes.views.edit.AttributeEditView attribute), 47
- model (shoop.admin.modules.attributes.views.list.AttributeListView attribute), 47
- model (shoop.admin.modules.categories.views.CategoryEditView attribute), 49
- model (shoop.admin.modules.categories.views.CategoryListView attribute), 50
- model (shoop.admin.modules.categories.views.edit.CategoryBaseForm.Meta attribute), 48
- model (shoop.admin.modules.categories.views.edit.CategoryEditView attribute), 49
- model (shoop.admin.modules.categories.views.list.CategoryListView attribute), 49
- model (shoop.admin.modules.contact_groups.views.ContactGroupDeleteView attribute), 52
- model (shoop.admin.modules.contact_groups.views.ContactGroupEditView attribute), 53
- model (shoop.admin.modules.contact_groups.views.ContactGroupListView attribute), 53
- model (shoop.admin.modules.contact_groups.views.delete.ContactGroupDeleteView attribute), 50
- model (shoop.admin.modules.contact_groups.views.edit.ContactGroupEditView attribute), 51
- model (shoop.admin.modules.contact_groups.views.forms.ContactGroupBaseForm attribute), 51
- model (shoop.admin.modules.contact_groups.views.list.ContactGroupListView attribute), 52
- model (shoop.admin.modules.contacts.views.ContactDetailView attribute), 56
- model (shoop.admin.modules.contacts.views.ContactEditView attribute), 57
- model (shoop.admin.modules.contacts.views.ContactListView attribute), 56
- model (shoop.admin.modules.contacts.views.detail.ContactDetailView attribute), 54
- model (shoop.admin.modules.contacts.views.edit.AddressForm.Meta attribute), 55
- model (shoop.admin.modules.contacts.views.edit.ContactEditView attribute), 55
- model (shoop.admin.modules.contacts.views.list.ContactListView attribute), 55
- model (shoop.admin.modules.manufacturers.views.edit.ManufacturerEditView attribute), 59
- model (shoop.admin.modules.manufacturers.views.edit.ManufacturerForm.Meta attribute), 58
- model (shoop.admin.modules.manufacturers.views.list.ManufacturerListView attribute), 59
- model (shoop.admin.modules.manufacturers.views.ManufacturerEditView attribute), 59
- model (shoop.admin.modules.manufacturers.views.ManufacturerListView attribute), 59
- model (shoop.admin.modules.orders.views.detail.OrderDetailView attribute), 61

model (shoop.admin.modules.orders.views.detail.OrderSetStatusView attribute), 61

model (shoop.admin.modules.orders.views.edit.OrderEditView attribute), 62

model (shoop.admin.modules.orders.views.list.OrderListView attribute), 63

model (shoop.admin.modules.orders.views.OrderCreateFullRefundView attribute), 68

model (shoop.admin.modules.orders.views.OrderCreatePaymentView attribute), 68

model (shoop.admin.modules.orders.views.OrderCreateRefundView attribute), 68

model (shoop.admin.modules.orders.views.OrderCreateShipmentView attribute), 69

model (shoop.admin.modules.orders.views.OrderDetailView attribute), 66

model (shoop.admin.modules.orders.views.OrderEditView attribute), 67

model (shoop.admin.modules.orders.views.OrderListView attribute), 67

model (shoop.admin.modules.orders.views.OrderSetStatusView attribute), 69

model (shoop.admin.modules.orders.views.payment.OrderCreatePaymentView attribute), 63

model (shoop.admin.modules.orders.views.refund.OrderCreateFullRefundView attribute), 65

model (shoop.admin.modules.orders.views.refund.OrderCreateRefundView attribute), 64

model (shoop.admin.modules.orders.views.shipment.OrderCreateShipmentView attribute), 66

model (shoop.admin.modules.permission_groups.views.edit.PermissionGroupEditView attribute), 72

model (shoop.admin.modules.permission_groups.views.edit.PermissionGroupFormMeta attribute), 72

model (shoop.admin.modules.permission_groups.views.list.PermissionGroupListView attribute), 73

model (shoop.admin.modules.permission_groups.views.PermissionGroupEditView attribute), 73

model (shoop.admin.modules.permission_groups.views.PermissionGroupListView attribute), 73

model (shoop.admin.modules.product_types.views.edit.ProductTypeEditView attribute), 74

model (shoop.admin.modules.product_types.views.edit.ProductTypeFormMeta attribute), 74

model (shoop.admin.modules.product_types.views.list.ProductTypeListView attribute), 74

model (shoop.admin.modules.product_types.views.ProductTypeEditView attribute), 75

model (shoop.admin.modules.product_types.views.ProductTypeListView attribute), 75

model (shoop.admin.modules.products.forms.base_forms.BaseProductFormMeta attribute), 76

model (shoop.admin.modules.products.forms.base_forms.BaseProductForm attribute), 77

model (shoop.admin.modules.products.forms.base_forms.ProductBaseForm attribute), 75

model (shoop.admin.modules.products.forms.base_forms.ShopProductForm attribute), 76

model (shoop.admin.modules.products.forms.BaseProductMediaForm.Meta attribute), 79

model (shoop.admin.modules.products.forms.ProductBaseForm.Meta attribute), 80

model (shoop.admin.modules.products.forms.ShopProductForm.Meta attribute), 81

model (shoop.admin.modules.products.views.delete.ProductDeleteView attribute), 83

model (shoop.admin.modules.products.views.edit.ProductEditView attribute), 84

model (shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSell attribute), 85

model (shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSell attribute), 84

model (shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSell attribute), 85

model (shoop.admin.modules.products.views.edit_media.ProductMediaEditView attribute), 86

model (shoop.admin.modules.products.views.edit_media.ProductMediaForm attribute), 85

model (shoop.admin.modules.products.views.edit_media.ProductMediaForm attribute), 86

model (shoop.admin.modules.products.views.edit_parent.ProductParentBase attribute), 87

model (shoop.admin.modules.products.views.list.ProductListView attribute), 88

model (shoop.admin.modules.products.views.ProductCrossSellEditView attribute), 89

model (shoop.admin.modules.products.views.ProductDeleteView attribute), 89

model (shoop.admin.modules.products.views.ProductEditView attribute), 89

model (shoop.admin.modules.products.views.ProductListView attribute), 90

model (shoop.admin.modules.products.views.ProductMediaEditView attribute), 90

model (shoop.admin.modules.sales_units.views.edit.SalesUnitEditView attribute), 91

model (shoop.admin.modules.sales_units.views.edit.SalesUnitForm.Meta attribute), 91

model (shoop.admin.modules.sales_units.views.list.SalesUnitListView attribute), 92

model (shoop.admin.modules.sales_units.views.SalesUnitEditView attribute), 92

model (shoop.admin.modules.sales_units.views.SalesUnitListView attribute), 92

model (shoop.admin.modules.service_providers.forms.CustomCarrierForm attribute), 94

model (shoop.admin.modules.service_providers.forms.CustomPaymentProcess attribute), 94

model (shoop.admin.modules.service_providers.views.ServiceProviderDetailView attribute), 93
 model (shoop.admin.modules.service_providers.views.ServiceProviderEditView attribute), 93
 model (shoop.admin.modules.service_providers.views.ServiceProviderListView attribute), 93
 model (shoop.admin.modules.services.behavior_form_part.BaseFormAdmin attribute), 96
 model (shoop.admin.modules.services.forms.BaseMethodForm.Meta attribute), 97
 model (shoop.admin.modules.services.forms.FixedCostBehaviorForm.Meta attribute), 98
 model (shoop.admin.modules.services.forms.GroupAvailabilityBehaviorForm.Meta attribute), 100
 model (shoop.admin.modules.services.forms.PaymentMethodEditView attribute), 98
 model (shoop.admin.modules.services.forms.RoundingBehaviorForm.Meta attribute), 101
 model (shoop.admin.modules.services.forms.ShippingMethodEditView attribute), 98
 model (shoop.admin.modules.services.forms.StaffOnlyBehaviorForm.Meta attribute), 100
 model (shoop.admin.modules.services.forms.WaivingCostBehaviorForm.Meta attribute), 98
 model (shoop.admin.modules.services.forms.WeightLimitsBehaviorForm.Meta attribute), 99
 model (shoop.admin.modules.services.PaymentMethodModule attribute), 101
 model (shoop.admin.modules.services.ServiceModule attribute), 101
 model (shoop.admin.modules.services.ShippingMethodModule attribute), 101
 model (shoop.admin.modules.services.views.PaymentMethodDetailView attribute), 95
 model (shoop.admin.modules.services.views.PaymentMethodEditView attribute), 95
 model (shoop.admin.modules.services.views.PaymentMethodListView attribute), 95
 model (shoop.admin.modules.services.views.ShippingMethodDetailView attribute), 95
 model (shoop.admin.modules.services.views.ShippingMethodEditView attribute), 95
 model (shoop.admin.modules.services.views.ShippingMethodListView attribute), 95
 model (shoop.admin.modules.shops.views.edit.ContactAddressForm.Meta attribute), 103
 model (shoop.admin.modules.shops.views.edit.ShopBaseForm.Meta attribute), 102
 model (shoop.admin.modules.shops.views.edit.ShopEditView attribute), 103
 model (shoop.admin.modules.shops.views.list.ShopListView attribute), 103
 model (shoop.admin.modules.shops.views.ShopEditView attribute), 104
 model (shoop.admin.modules.shops.views.ShopDetailView attribute), 104
 model (shoop.admin.modules.suppliers.views.edit.SupplierEditView attribute), 105
 model (shoop.admin.modules.suppliers.views.edit.SupplierForm.Meta attribute), 105
 model (shoop.admin.modules.suppliers.views.list.SupplierListView attribute), 105
 model (shoop.admin.modules.suppliers.views.SupplierEditView attribute), 106
 model (shoop.admin.modules.suppliers.views.SupplierListView attribute), 106
 model (shoop.admin.modules.taxes.views.CustomerTaxGroupEditView attribute), 110
 model (shoop.admin.modules.taxes.views.CustomerTaxGroupListView attribute), 111
 model (shoop.admin.modules.taxes.views.edit.CustomerTaxGroupEditView attribute), 108
 model (shoop.admin.modules.taxes.views.edit.CustomerTaxGroupForm.Meta attribute), 108
 model (shoop.admin.modules.taxes.views.edit.TaxClassEditView attribute), 109
 model (shoop.admin.modules.taxes.views.edit.TaxClassForm.Meta attribute), 108
 model (shoop.admin.modules.taxes.views.edit.TaxEditView attribute), 108
 model (shoop.admin.modules.taxes.views.edit.TaxForm.Meta attribute), 107
 model (shoop.admin.modules.taxes.views.list.CustomerTaxGroupListView attribute), 109
 model (shoop.admin.modules.taxes.views.list.TaxClassListView attribute), 109
 model (shoop.admin.modules.taxes.views.list.TaxListView attribute), 109
 model (shoop.admin.modules.taxes.views.TaxClassEditView attribute), 110
 model (shoop.admin.modules.taxes.views.TaxClassListView attribute), 110
 model (shoop.admin.modules.taxes.views.TaxEditView attribute), 110
 model (shoop.admin.modules.taxes.views.TaxListView attribute), 110
 model (shoop.admin.modules.users.views.list.UserListView attribute), 112
 model (shoop.admin.modules.users.views.password.UserChangePasswordView attribute), 113
 model (shoop.admin.modules.users.views.password.UserResetPasswordView attribute), 113
 model (shoop.admin.modules.users.views.permissions.UserChangePermissionsView attribute), 114
 model (shoop.admin.modules.users.views.UserChangePasswordView attribute), 115
 model (shoop.admin.modules.users.views.UserChangePermissionsView attribute), 115

model (shoop.admin.modules.users.views.UserListView attribute), 114
 model (shoop.admin.modules.users.views.UserResetPassword attribute), 115
 model (shoop.admin.views.select.MultiselectAjaxView attribute), 124
 model (shoop.campaigns.admin_module.form_parts.SalesRangeForm attribute), 142
 model (shoop.campaigns.admin_module.form_sets.BaseForm attribute), 144
 model (shoop.campaigns.admin_module.form_sets.BasketConditionForm attribute), 144
 model (shoop.campaigns.admin_module.form_sets.BasketDiscountForm attribute), 144
 model (shoop.campaigns.admin_module.form_sets.BasketLineEffectForm attribute), 144
 model (shoop.campaigns.admin_module.form_sets.CatalogConditionForm attribute), 144
 model (shoop.campaigns.admin_module.form_sets.CatalogEffectForm attribute), 145
 model (shoop.campaigns.admin_module.form_sets.CatalogFilterForm attribute), 145
 model (shoop.campaigns.admin_module.forms.BasketCampaignForm attribute), 135
 model (shoop.campaigns.admin_module.forms.BasketDiscountForm attribute), 135
 model (shoop.campaigns.admin_module.forms.BasketDiscountForm attribute), 136
 model (shoop.campaigns.admin_module.forms.BasketMaxTotalAmountForm attribute), 136
 model (shoop.campaigns.admin_module.forms.BasketMaxTotalProductConditionForm attribute), 136
 model (shoop.campaigns.admin_module.forms.BasketTotalAmountForm attribute), 136
 model (shoop.campaigns.admin_module.forms.BasketTotalProductConditionForm attribute), 137
 model (shoop.campaigns.admin_module.forms.CatalogCampaignForm attribute), 137
 model (shoop.campaigns.admin_module.forms.CategoryFilterForm attribute), 137
 model (shoop.campaigns.admin_module.forms.ContactBasketConditionForm attribute), 137
 model (shoop.campaigns.admin_module.forms.ContactConditionForm attribute), 137
 model (shoop.campaigns.admin_module.forms.ContactGroupBasketConditionForm attribute), 138
 model (shoop.campaigns.admin_module.forms.ContactGroupConditionForm attribute), 138
 model (shoop.campaigns.admin_module.forms.CouponForm attribute), 138
 model (shoop.campaigns.admin_module.forms.DiscountFromProductForm attribute), 138
 model (shoop.campaigns.admin_module.forms.FreeProductLineForm attribute), 139
 model (shoop.campaigns.admin_module.forms.ProductDiscountAmountForm attribute), 139
 model (shoop.campaigns.admin_module.forms.ProductDiscountPercentageForm attribute), 139
 model (shoop.campaigns.admin_module.forms.ProductFilterForm.Meta attribute), 139
 model (shoop.campaigns.admin_module.forms.ProductsInBasketConditionForm attribute), 140
 model (shoop.campaigns.admin_module.forms.ProductTypeFilterForm.Meta attribute), 140
 model (shoop.campaigns.admin_module.views.BasketCampaignEditView attribute), 141
 model (shoop.campaigns.admin_module.views.BasketCampaignListView attribute), 141
 model (shoop.campaigns.admin_module.views.CatalogCampaignEditView attribute), 140
 model (shoop.campaigns.admin_module.views.CatalogCampaignListView attribute), 140
 model (shoop.campaigns.admin_module.views.CouponEditView attribute), 141
 model (shoop.campaigns.admin_module.views.CouponListView attribute), 141
 model (shoop.campaigns.models.basket_conditions.BasketCondition attribute), 145
 model (shoop.campaigns.models.basket_conditions.ContactBasketCondition attribute), 151
 model (shoop.campaigns.models.basket_conditions.ContactGroupBasketCondition attribute), 151
 model (shoop.campaigns.models.basket_conditions.ProductsInBasketCondition attribute), 150
 model (shoop.campaigns.models.basket_line_effects.BasketLineEffect attribute), 152
 model (shoop.campaigns.models.basket_line_effects.BasketLineEffect attribute), 155
 model (shoop.campaigns.models.basket_line_effects.DiscountFromProductLine attribute), 156
 model (shoop.campaigns.models.basket_line_effects.FreeProductLine attribute), 155
 model (shoop.campaigns.models.BasketCondition attribute), 172
 model (shoop.campaigns.models.BasketDiscountEffect attribute), 172
 model (shoop.campaigns.models.BasketLineEffect attribute), 170
 model (shoop.campaigns.models.catalog_filters.CatalogFilter attribute), 161
 model (shoop.campaigns.models.catalog_filters.CategoryFilter attribute), 163
 model (shoop.campaigns.models.catalog_filters.ProductFilter attribute), 163
 model (shoop.campaigns.models.catalog_filters.ProductTypeFilter attribute), 162
 model (shoop.campaigns.models.CatalogFilter attribute), 175

- model (shoop.campaigns.models.context_conditions.ContactCondition attribute), 166
- model (shoop.campaigns.models.context_conditions.ContactCondition attribute), 166
- model (shoop.campaigns.models.context_conditions.ContextCondition attribute), 165
- model (shoop.campaigns.models.ContextCondition attribute), 176
- model (shoop.campaigns.models.product_effects.ProductDiscountEffect attribute), 167
- model (shoop.campaigns.models.ProductDiscountEffect attribute), 174
- model (shoop.core.api.orders.AddressSerializer.Meta attribute), 179
- model (shoop.core.api.orders.OrderLineSerializer.Meta attribute), 179
- model (shoop.core.api.orders.OrderSerializer.Meta attribute), 179
- model (shoop.core.api.orders.PaymentSerializer.Meta attribute), 179
- model (shoop.core.api.products.ProductSerializer.Meta attribute), 180
- model (shoop.core.api.products.ShopProductSerializer.Meta attribute), 180
- model (shoop.default_tax.admin_module.views.TaxRuleEditorView attribute), 266
- model (shoop.default_tax.admin_module.views.TaxRuleFormView attribute), 266
- model (shoop.default_tax.admin_module.views.TaxRuleListView attribute), 266
- model (shoop.front.admin_module.carts.views.CartListView attribute), 268
- model (shoop.front.apps.customer_information.forms.AddressForm.Meta attribute), 272
- model (shoop.front.apps.customer_information.forms.CompanyForm.Meta attribute), 272
- model (shoop.front.apps.customer_information.forms.PersonForm.Meta attribute), 271
- model (shoop.front.apps.personal_order_history.views.OrderView attribute), 273
- model (shoop.front.apps.simple_search.views.SearchView attribute), 277
- model (shoop.front.checkout.addresses.AddressForm.Meta attribute), 282
- model (shoop.front.checkout.addresses.CompanyForm.Meta attribute), 282
- model (shoop.front.views.category.CategoryView attribute), 291
- model (shoop.front.views.order.OrderCompleteView attribute), 292
- model (shoop.front.views.order.OrderRequiresVerificationView attribute), 292
- model (shoop.front.views.payment.ProcessPaymentView attribute), 293
- model (shoop.front.views.product.ProductDetailView attribute), 293
- model (shoop.notify.admin_module.forms.ScriptForm.Meta attribute), 301
- model (shoop.notify.admin_module.views.edit.ScriptEditView attribute), 299
- model (shoop.notify.admin_module.views.editor.EditScriptContentView attribute), 299
- model (shoop.notify.admin_module.views.EditScriptContentView attribute), 300
- model (shoop.notify.admin_module.views.list.ScriptListView attribute), 300
- model (shoop.notify.admin_module.views.ScriptEditView attribute), 300
- model (shoop.notify.admin_module.views.ScriptListView attribute), 301
- model (shoop.simple_cms.admin_module.views.PageEditView attribute), 317
- model (shoop.simple_cms.admin_module.views.PageForm.Meta attribute), 317
- model (shoop.simple_cms.admin_module.views.PageListView attribute), 317
- model (shoop.simple_cms.views.PageView attribute), 320
- model (shoop.simple_supplier.admin_module.views.StocksListView attribute), 322
- model (shoop.testing.service_forms.CarrierWithCheckoutPhaseForm.Meta attribute), 334
- model (shoop.testing.service_forms.PaymentWithCheckoutPhaseForm.Meta attribute), 333
- model (shoop.testing.service_forms.PseudoPaymentProcessorForm.Meta attribute), 333
- model (shoop.xtheme.admin_module.views.ThemeConfigDetailView attribute), 352
- model (shoop.xtheme.forms.GenericThemeForm.Meta attribute), 361
- model (shoop.notify.typology.Model attribute), 313
- model_url() (in module shoop.admin.template_helpers.shoop_admin), 116
- model_url() (in module shoop.front.template_helpers.urls), 289
- ModelCachingDescriptor (class in shoop.core.utils.model_caching_descriptor), 257
- ModernArtImageGenerator (class in shoop.testing.image_generator), 332
- modified_by (shoop.campaigns.models.BasketCampaign attribute), 171
- modified_by (shoop.campaigns.models.Campaign attribute), 173
- modified_by (shoop.campaigns.models.campaigns.BasketCampaign attribute), 159
- modified_by (shoop.campaigns.models.campaigns.Campaign

- attribute), 157
- modified_by (shoop.campaigns.models.campaigns.CatalogCampaign attribute), 158
- modified_by (shoop.campaigns.models.campaigns.Coupon attribute), 160
- modified_by (shoop.campaigns.models.campaigns.CouponUsage attribute), 159
- modified_by (shoop.campaigns.models.CatalogCampaign attribute), 174
- modified_by (shoop.campaigns.models.Coupon attribute), 177
- modified_by (shoop.core.models.Order attribute), 202
- modified_by (shoop.core.order_creator.OrderSource attribute), 240
- modified_by (shoop.simple_cms.models.Page attribute), 318
- Module, 18
- module (shoop.core.modules.interface.ModuleInterface attribute), 238
- module (shoop.core.modules.ModuleInterface attribute), 238
- module_data (shoop.core.models.Supplier attribute), 234
- module_identifier (shoop.core.modules.interface.ModuleInterface attribute), 238
- module_identifier (shoop.core.modules.ModuleInterface attribute), 238
- module_options_field (shoop.core.modules.interface.ModuleInterface attribute), 238
- module_options_field (shoop.core.modules.ModuleInterface attribute), 238
- module_provides_key (shoop.core.models.Supplier attribute), 234
- module_provides_key (shoop.core.modules.interface.ModuleInterface attribute), 238
- module_provides_key (shoop.core.modules.ModuleInterface attribute), 238
- ModuleInterface (class in shoop.core.modules), 238
- ModuleInterface (class in shoop.core.modules.interface), 237
- ModuleNotFound, 237
- ModWSGIReloadMethod (class in shoop.addons.reloader), 41
- Money (class in shoop.utils.money), 344
- money() (in module shoop.core.templatetags.shoop_common), 256
- MoneyProperty (class in shoop.utils.properties), 347
- MoneyPropped (class in shoop.utils.properties), 348
- MoneyValueField (class in shoop.core.fields), 184
- MultiFieldTextFilter (class in shoop.admin.utils.picotable), 119
- MultiLanguageModelForm (class in shoop.utils.multilanguage_model_form), 344
- MULTILINGUAL (shoop.notify.enums.TemplateUse attribute), 309
- MULTILINGUAL (shoop.notify.TemplateUse attribute), 315
- MULTIPLIER (shoop.admin.forms.fields.PercentageField attribute), 45
- MultiSelectAjaxView (class in shoop.admin.views.select), 124
- MutableAddress (class in shoop.core.models), 198
- MutableAddress.DoesNotExist, 198
- MutableAddress.MultipleObjectsReturned, 198
- N**
- name (shoop.addons.admin_module.AddonModule attribute), 40
- name (shoop.admin.base.AdminModule attribute), 124
- name (shoop.admin.base.OrderSection attribute), 126
- name (shoop.admin.modules.attributes.AttributeModule attribute), 48
- name (shoop.admin.modules.categories.CategoryModule attribute), 50
- name (shoop.admin.modules.contact_groups.ContactGroupModule attribute), 53
- name (shoop.admin.modules.contacts.ContactModule attribute), 57
- name (shoop.admin.modules.demo.DemoModule attribute), 58
- name (shoop.admin.modules.manufacturers.ManufacturerModule attribute), 60
- name (shoop.admin.modules.media.MediaModule attribute), 61
- name (shoop.admin.modules.orders.OrderModule attribute), 71
- name (shoop.admin.modules.orders.sections.ContentsOrderSection attribute), 71
- name (shoop.admin.modules.orders.sections.LogEntriesOrderSection attribute), 71
- name (shoop.admin.modules.orders.sections.PaymentOrderSection attribute), 71
- name (shoop.admin.modules.permission_groups.PermissionGroupModule attribute), 73
- name (shoop.admin.modules.product_types.ProductTypeModule attribute), 75
- name (shoop.admin.modules.products.ProductModule attribute), 91
- name (shoop.admin.modules.products.views.edit.ProductImageMediaForm attribute), 84
- name (shoop.admin.modules.products.views.edit.ProductMediaFormPart attribute), 83
- name (shoop.admin.modules.sales_units.SalesUnitModule attribute), 92
- name (shoop.admin.modules.service_providers.ServiceProviderModule attribute), 94
- name (shoop.admin.modules.services.PaymentMethodModule attribute), 101

name (shoop.admin.modules.services.ServiceModule attribute), 101
 name (shoop.admin.modules.services.ShippingMethodModule attribute), 101
 name (shoop.admin.modules.services.weight_based_pricing.WeightBasedPricingFormPart attribute), 96
 name (shoop.admin.modules.shops.ShopModule attribute), 104
 name (shoop.admin.modules.suppliers.SupplierModule attribute), 106
 name (shoop.admin.modules.system.SystemModule attribute), 107
 name (shoop.admin.modules.taxes.TaxModule attribute), 111
 name (shoop.admin.modules.users.UserModule attribute), 116
 name (shoop.admin.ShoopAdminAppConfig attribute), 130
 name (shoop.api.ShoopApiAppConfig attribute), 132
 name (shoop.campaigns.admin_module.CampaignAdminModule attribute), 145
 name (shoop.campaigns.admin_module.form_parts.SalesRangesFormPart attribute), 142
 name (shoop.campaigns.apps.CampaignAppConfig attribute), 177
 name (shoop.campaigns.models.basket_conditions.BasketCondition attribute), 145
 name (shoop.campaigns.models.basket_conditions.BasketMaxTotalAmountCondition attribute), 149
 name (shoop.campaigns.models.basket_conditions.BasketMaxTotalProductAmountCondition attribute), 148
 name (shoop.campaigns.models.basket_conditions.BasketTotalAmountCondition attribute), 147
 name (shoop.campaigns.models.basket_conditions.BasketTotalProductAmountCondition attribute), 146
 name (shoop.campaigns.models.basket_conditions.ContactBasketCondition attribute), 152
 name (shoop.campaigns.models.basket_conditions.ContactGroupBasketCondition attribute), 151
 name (shoop.campaigns.models.basket_conditions.ProductsInBasketCondition attribute), 150
 name (shoop.campaigns.models.basket_effects.BasketDiscountAmount attribute), 153
 name (shoop.campaigns.models.basket_effects.BasketDiscountPercentage attribute), 154
 name (shoop.campaigns.models.basket_line_effects.DiscountFromProduct attribute), 156
 name (shoop.campaigns.models.basket_line_effects.FreeProduct attribute), 156
 name (shoop.campaigns.models.BasketCondition attribute), 172
 name (shoop.campaigns.models.catalog_filters.CatalogFilter attribute), 161
 name (shoop.campaigns.models.catalog_filters.CategoryFilter attribute), 163
 name (shoop.campaigns.models.catalog_filters.ProductFilter attribute), 163
 name (shoop.campaigns.models.catalog_filters.ProductTypeFilter attribute), 163
 name (shoop.campaigns.models.CatalogFilter attribute), 175
 name (shoop.campaigns.models.context_conditions.ContactCondition attribute), 166
 name (shoop.campaigns.models.context_conditions.ContactGroupCondition attribute), 166
 name (shoop.campaigns.models.context_conditions.ContextCondition attribute), 165
 name (shoop.campaigns.models.ContextCondition attribute), 176
 name (shoop.campaigns.models.product_effects.ProductDiscountAmount attribute), 168
 name (shoop.campaigns.models.product_effects.ProductDiscountPercentage attribute), 169
 name (shoop.campaigns.modules.BasketCampaignModule attribute), 178
 name (shoop.campaigns.modules.CatalogCampaignModule attribute), 178
 name (shoop.core.models.Attribute attribute), 188
 name (shoop.core.models.Category attribute), 190
 name (shoop.core.models.ContactGroup attribute), 194
 name (shoop.core.models.CustomerTaxGroup attribute), 195
 name (shoop.core.models.FixedCostBehaviorComponent attribute), 194
 name (shoop.core.models.GroupAvailabilityBehaviorComponent attribute), 194
 name (shoop.core.models.OrderStatus attribute), 205
 name (shoop.core.models.PaymentMethod attribute), 206
 name (shoop.core.models.PersonContact attribute), 209
 name (shoop.core.models.Product attribute), 213, 217
 name (shoop.core.models.ProductType attribute), 220
 name (shoop.core.models.ProductVariationVariable attribute), 221
 name (shoop.core.models.RoundingBehaviorComponent attribute), 222
 name (shoop.core.models.SalesUnit attribute), 223
 name (shoop.core.models.Service attribute), 225
 name (shoop.core.models.ServiceBehaviorComponent attribute), 225
 name (shoop.core.models.ServiceProvider attribute), 227
 name (shoop.core.models.ShippingMethod attribute), 229
 name (shoop.core.models.Shop attribute), 230
 name (shoop.core.models.StaffOnlyBehaviorComponent attribute), 232
 name (shoop.core.models.Tax attribute), 235
 name (shoop.core.models.TaxClass attribute), 235
 name (shoop.core.models.WaivingCostBehaviorComponent attribute), 236

name (shoop.core.models.WeightBasedPricingBehaviorComponent attribute), 237
 name (shoop.core.models.WeightLimitsBehaviorComponent attribute), 237
 name (shoop.core.pricing.default_pricing.DefaultPricingModule attribute), 242
 name (shoop.core.pricing.PricingModule attribute), 250
 name (shoop.core.ShoopCoreAppConfig attribute), 264
 name (shoop.core.suppliers.base.BaseSupplierModule attribute), 251
 name (shoop.core.suppliers.BaseSupplierModule attribute), 252
 name (shoop.core.taxing.LineTax attribute), 253
 name (shoop.core.taxing.TaxModule attribute), 254
 name (shoop.customer_group_pricing.CustomerGroupPricingAppConfig attribute), 265
 name (shoop.customer_group_pricing.module.CustomerGroupPricingModule attribute), 265
 name (shoop.default_tax.admin_module.TaxRulesAdminModule attribute), 266
 name (shoop.default_tax.AppConfig attribute), 268
 name (shoop.default_tax.module.DefaultTaxModule attribute), 267
 name (shoop.front.admin_module.CartAdminModule attribute), 269
 name (shoop.front.apps.auth.AuthAppConfig attribute), 271
 name (shoop.front.apps.customer_information.AppConfig attribute), 273
 name (shoop.front.apps.personal_order_history.AppConfig attribute), 273
 name (shoop.front.apps.registration.RegistrationAppConfig attribute), 275
 name (shoop.front.apps.simple_order_notification.admin_module.SimpleOrderNotificationAdminModule attribute), 275
 name (shoop.front.apps.simple_order_notification.SimpleOrderNotificationAppConfig attribute), 275
 name (shoop.front.apps.simple_search.SimpleSearchAppConfig attribute), 277
 name (shoop.front.apps.simple_search.template_helpers.TemplateHelpers attribute), 276
 name (shoop.front.notify_events.OrderReceived attribute), 294
 name (shoop.front.ShoopFrontAppConfig attribute), 296
 name (shoop.guide.apps.AppConfig attribute), 296
 name (shoop.notify.Action attribute), 314
 name (shoop.notify.actions.AddNotification attribute), 298
 name (shoop.notify.actions.AddOrderLogEntry attribute), 298
 name (shoop.notify.actions.debug.SetDebugFlag attribute), 297
 name (shoop.notify.actions.email.SendEmail attribute), 297
 name (shoop.notify.actions.notification.AddNotification attribute), 297
 name (shoop.notify.actions.order.AddOrderLogEntry attribute), 298
 name (shoop.notify.actions.SendEmail attribute), 298
 name (shoop.notify.actions.SetDebugFlag attribute), 299
 name (shoop.notify.admin_module.NotifyAdminModule attribute), 301
 name (shoop.notify.base.Action attribute), 309
 name (shoop.notify.base.Base attribute), 307
 name (shoop.notify.base.Condition attribute), 308
 name (shoop.notify.base.ScriptItem attribute), 308
 name (shoop.notify.Condition attribute), 315
 name (shoop.notify.conditions.BooleanEqual attribute), 303
 name (shoop.notify.conditions.Empty attribute), 303
 name (shoop.notify.conditions.IntegerEqual attribute), 303
 name (shoop.notify.conditions.LanguageEqual attribute), 303
 name (shoop.notify.conditions.NonEmpty attribute), 304
 name (shoop.notify.conditions.simple.BaseEqual attribute), 302
 name (shoop.notify.conditions.simple.CaseInsensitiveStringEqual attribute), 303
 name (shoop.notify.conditions.simple.Empty attribute), 302
 name (shoop.notify.conditions.simple.NonEmpty attribute), 302
 name (shoop.notify.conditions.TextEqual attribute), 304
 name (shoop.notify.typology.Boolean attribute), 312
 name (shoop.notify.typology.Decimal attribute), 312
 name (shoop.notify.typology.Email attribute), 312
 name (shoop.notify.typology.Integer attribute), 312
 name (shoop.notify.typology.Language attribute), 312
 name (shoop.notify.typology.Model attribute), 313
 name (shoop.notify.typology.Phone attribute), 313
 name (shoop.notify.typology.Text attribute), 312
 name (shoop.notify.typology.Type attribute), 311
 name (shoop.notify.typology.URL attribute), 312
 name (shoop.order_printouts.AppConfig attribute), 316
 name (shoop.simple_cms.admin_module.SimpleCMSAdminModule attribute), 318
 name (shoop.simple_cms.AppConfig attribute), 321
 name (shoop.simple_cms.plugins.PageLinksPlugin attribute), 319
 name (shoop.simple_cms.template_helpers.SimpleCMSTemplateHelpers attribute), 320
 name (shoop.simple_supplier.admin_module.forms.SimpleSupplierFormPart attribute), 321
 name (shoop.simple_supplier.admin_module.StocksAdminModule attribute), 322
 name (shoop.simple_supplier.module.SimpleSupplierModule

- attribute), 324
- name (shoop.simple_supplier.ShoopSimpleSupplierAppConfig attribute), 325
- name (shoop.testing.factories.CategoryFactory attribute), 330
- name (shoop.testing.factories.CompanyFactory attribute), 329
- name (shoop.testing.factories.ProductFactory attribute), 330
- name (shoop.testing.factories.ProductTypeFactory attribute), 329
- name (shoop.testing.factories.SalesUnitFactory attribute), 330
- name (shoop.testing.factories.ShopFactory attribute), 329
- name (shoop.testing.models.ExpensiveSwedenBehaviorComponent attribute), 327
- name (shoop.testing.ShoopTestingAppConfig attribute), 336
- name (shoop.themes.classic_gray.ClassicGrayTheme attribute), 337
- name (shoop.themes.classic_gray.ClassicGrayThemeAppConfig attribute), 337
- name (shoop.themes.classic_gray.plugins.ProductCrossSellPlugin attribute), 337
- name (shoop.themes.classic_gray.plugins.ProductHighlightPlugin attribute), 337
- name (shoop.utils.ShoopUtilsAppConfig attribute), 351
- name (shoop.xtheme.admin_module.XthemeAdminModule attribute), 352
- name (shoop.xtheme.Plugin attribute), 374
- name (shoop.xtheme.plugins.category_links.CategoryLinksPlugin attribute), 353
- name (shoop.xtheme.plugins.image.ImagePlugin attribute), 354
- name (shoop.xtheme.plugins.snippets.SnippetsPlugin attribute), 355
- name (shoop.xtheme.plugins.social_media_links.SocialMediaLinksPlugin attribute), 355
- name (shoop.xtheme.plugins.text.TextPlugin attribute), 356
- name (shoop.xtheme.Theme attribute), 376
- name_attr (shoop.core.models.TranslatableShoopModel attribute), 235
- name_template (shoop.admin.modules.services.PaymentMethodModule attribute), 102
- name_template (shoop.admin.modules.services.ServiceModule attribute), 101
- name_template (shoop.admin.modules.services.ShippingMethodModule attribute), 101
- NameMixin (class in shoop.core.utils.name_mixin), 257
- NestingError, 366
- new() (shoop.utils.money.Money method), 344
- new() (shoop.utils.numbers.UnittedDecimal method), 345
- NewActionButton (class in shoop.admin.toolbar), 128
- NewLogEntryView (class in shoop.admin.modules.orders.views), 66
- NewLogEntryView (class in shoop.admin.modules.orders.views.log), 63
- next_phase (shoop.front.checkout.CheckoutPhaseViewMixin attribute), 286
- NoLanguageMatches, 311
- NoModelUrl, 121
- non_reentrant() (in module shoop.utils.decorators), 339
- NonConstant, 366
- NONE (shoop.core.models.OrderStatusRole attribute), 205
- NONE (shoop.notify.enums.StepConditionOperator attribute), 309
- NONE (shoop.notify.enums.TemplateUse attribute), 309
- NONE (shoop.notify.TemplateUse attribute), 315
- none() (in module shoop.notify.script), 310
- NonEmpty (class in shoop.notify.conditions), 303
- NonEmpty (class in shoop.notify.conditions.simple), 302
- noop_node() (in module shoop.xtheme.parsing), 367
- NoPaymentToCreateException, 260
- NoProductsToShipException, 260
- NoRefundToCreateException, 260
- NORMAL (shoop.core.models.ProductMode attribute), 220
- NORMAL (shoop.notify.enums.Priority attribute), 310
- NOT_PAID (shoop.core.models.PaymentStatus attribute), 208
- NOT_SHIPPED (shoop.core.models.ShippingMode attribute), 229
- NOT_SHIPPED (shoop.core.models.ShippingStatus attribute), 229
- NOT_VISIBLE (shoop.core.models.AttributeVisibility attribute), 189
- NotifyPlugin (shoop.utils.analog.LogEntryKind attribute), 338
- Notification (class in shoop.admin.base), 125
- Notification (class in shoop.notify.models), 305
- Notification (class in shoop.notify.models.notification), 304
- Notification.DoesNotExist, 304, 305
- Notification.MultipleObjectsReturned, 304, 305
- NotificationManager (class in shoop.notify.models.notification), 304
- NotifyAdminModule (class in shoop.notify.admin_module), 301
- OrderModule (in module shoop.core.templatetags.shoop_common), 256
- O
- object (shoop.admin.form_part.SaveFormPartsMixin attribute), 127

objects (shoop.campaigns.models.basket_conditions.BasketCondition attribute), 146

objects (shoop.campaigns.models.basket_conditions.BasketDiscountCondition attribute), 149

objects (shoop.campaigns.models.basket_conditions.BasketDiscountEffect attribute), 148

objects (shoop.campaigns.models.basket_conditions.BasketLineEffect attribute), 147

objects (shoop.campaigns.models.basket_conditions.BasketLineEffect attribute), 147

objects (shoop.campaigns.models.basket_conditions.ContactGroupSalesRange attribute), 152

objects (shoop.campaigns.models.basket_conditions.ContactGroupSalesRange attribute), 151

objects (shoop.campaigns.models.basket_conditions.ProductDiscountAmount attribute), 150

objects (shoop.campaigns.models.basket_effects.BasketDiscountEffect attribute), 153

objects (shoop.campaigns.models.basket_effects.BasketDiscountEffect attribute), 153

objects (shoop.campaigns.models.basket_effects.BasketDiscountEffect attribute), 154

objects (shoop.campaigns.models.basket_line_effects.BasketDiscountEffect attribute), 155

objects (shoop.campaigns.models.basket_line_effects.DiscountFromProduct attribute), 157

objects (shoop.campaigns.models.basket_line_effects.FreeProductLine attribute), 156

objects (shoop.campaigns.models.BasketCampaign attribute), 171

objects (shoop.campaigns.models.BasketCondition attribute), 173

objects (shoop.campaigns.models.BasketDiscountEffect attribute), 172

objects (shoop.campaigns.models.BasketLineEffect attribute), 170

objects (shoop.campaigns.models.Campaign attribute), 173

objects (shoop.campaigns.models.campaigns.BasketCampaign attribute), 159

objects (shoop.campaigns.models.campaigns.BasketCampaign attribute), 161

objects (shoop.campaigns.models.campaigns.Campaign attribute), 157

objects (shoop.campaigns.models.campaigns.CatalogCampaign attribute), 158

objects (shoop.campaigns.models.campaigns.CatalogCampaign attribute), 161

objects (shoop.campaigns.models.campaigns.Coupon attribute), 160

objects (shoop.campaigns.models.campaigns.CouponUsage attribute), 159

objects (shoop.campaigns.models.catalog_filters.CatalogFilter attribute), 161

objects (shoop.campaigns.models.catalog_filters.CategoryFilter attribute), 164

objects (shoop.campaigns.models.catalog_filters.ProductFilter attribute), 163

objects (shoop.campaigns.models.catalog_filters.ProductTypeFilter attribute), 162

objects (shoop.campaigns.models.CatalogCampaign attribute), 174

objects (shoop.campaigns.models.CatalogFilter attribute), 175

objects (shoop.campaigns.models.contact_group_sales_ranges.ContactGroupSalesRange attribute), 164

objects (shoop.campaigns.models.ContactGroupSalesRange attribute), 175

objects (shoop.campaigns.models.context_conditions.ContactCondition attribute), 167

objects (shoop.campaigns.models.context_conditions.ContactGroupCondition attribute), 166

objects (shoop.campaigns.models.context_conditions.ContextCondition attribute), 165

objects (shoop.campaigns.models.ContextCondition attribute), 176

objects (shoop.campaigns.models.Coupon attribute), 177

objects (shoop.campaigns.models.product_effects.ProductDiscountAmount attribute), 168

objects (shoop.campaigns.models.product_effects.ProductDiscountEffect attribute), 167

objects (shoop.campaigns.models.product_effects.ProductDiscountPercentage attribute), 169

objects (shoop.campaigns.models.ProductDiscountEffect attribute), 174

objects (shoop.core.models.AnonymousContact attribute), 187

objects (shoop.core.models.Attribute attribute), 188

objects (shoop.core.models.Carrier attribute), 189

objects (shoop.core.models.Category attribute), 190

objects (shoop.core.models.CompanyContact attribute), 191

objects (shoop.core.models.ConfigurationItem attribute), 192

objects (shoop.core.models.Contact attribute), 193

objects (shoop.core.models.ContactGroup attribute), 194

objects (shoop.core.models.Counter attribute), 194

objects (shoop.core.models.CustomCarrier attribute), 195

objects (shoop.core.models.CustomerTaxGroup attribute), 195

objects (shoop.core.models.CustomPaymentProcessor attribute), 196

objects (shoop.core.models.FixedCostBehaviorComponent attribute), 196

objects (shoop.core.models.GroupAvailabilityBehaviorComponent attribute), 194

objects (shoop.core.models.ImmutableAddress attribute), 198

- objects (shoop.core.models.Manufacturer attribute), 198
- objects (shoop.core.models.MutableAddress attribute), 199
- objects (shoop.core.models.Order attribute), 202
- objects (shoop.core.models.OrderLine attribute), 203
- objects (shoop.core.models.OrderLineTax attribute), 204
- objects (shoop.core.models.OrderLogEntry attribute), 205
- objects (shoop.core.models.OrderStatus attribute), 205
- objects (shoop.core.models.Payment attribute), 206
- objects (shoop.core.models.PaymentMethod attribute), 206
- objects (shoop.core.models.PaymentProcessor attribute), 207
- objects (shoop.core.models.PersistentCacheEntry attribute), 208
- objects (shoop.core.models.PersonContact attribute), 209
- objects (shoop.core.models.PolymorphicShoopModel attribute), 210
- objects (shoop.core.models.PolymorphicTranslatableShoopModel attribute), 210
- objects (shoop.core.models.Product attribute), 213, 217
- objects (shoop.core.models.ProductAttribute attribute), 218
- objects (shoop.core.models.ProductCrossSell attribute), 218
- objects (shoop.core.models.ProductMedia attribute), 219
- objects (shoop.core.models.ProductPackageLink attribute), 220
- objects (shoop.core.models.ProductType attribute), 220
- objects (shoop.core.models.ProductVariationResult attribute), 221
- objects (shoop.core.models.ProductVariationVariable attribute), 221
- objects (shoop.core.models.ProductVariationVariableValue attribute), 221
- objects (shoop.core.models.RoundingBehaviorComponent attribute), 222
- objects (shoop.core.models.SalesUnit attribute), 223
- objects (shoop.core.models.SavedAddress attribute), 223
- objects (shoop.core.models.Service attribute), 225
- objects (shoop.core.models.ServiceBehaviorComponent attribute), 225
- objects (shoop.core.models.ServiceProvider attribute), 227
- objects (shoop.core.models.Shipment attribute), 228
- objects (shoop.core.models.ShipmentProduct attribute), 228
- objects (shoop.core.models.ShippingMethod attribute), 229
- objects (shoop.core.models.Shop attribute), 230
- objects (shoop.core.models.ShopProduct attribute), 231
- objects (shoop.core.models.StaffOnlyBehaviorComponent attribute), 233
- objects (shoop.core.models.SuppliedProduct attribute), 233
- objects (shoop.core.models.Supplier attribute), 234
- objects (shoop.core.models.Tax attribute), 235
- objects (shoop.core.models.TaxClass attribute), 235
- objects (shoop.core.models.TranslatableShoopModel attribute), 236
- objects (shoop.core.models.WaivingCostBehaviorComponent attribute), 236
- objects (shoop.core.models.WeightBasedPriceRange attribute), 236
- objects (shoop.core.models.WeightBasedPricingBehaviorComponent attribute), 237
- objects (shoop.core.models.WeightLimitsBehaviorComponent attribute), 237
- objects (shoop.customer_group_pricing.models.CgpPrice attribute), 265
- objects (shoop.default_tax.models.TaxRule attribute), 267
- objects (shoop.front.models.stored_basket.StoredBasket attribute), 287
- objects (shoop.front.models.StoredBasket attribute), 288
- objects (shoop.notify.models.Notification attribute), 306
- objects (shoop.notify.models.notification.Notification attribute), 304
- objects (shoop.notify.models.Script attribute), 306
- objects (shoop.notify.models.script.Script attribute), 305
- objects (shoop.simple_cms.models.Page attribute), 318
- objects (shoop.simple_cms.models.PageTranslation attribute), 319
- objects (shoop.simple_supplier.models.StockAdjustment attribute), 323
- objects (shoop.simple_supplier.models.StockCount attribute), 324
- objects (shoop.testing.models.CarrierWithCheckoutPhase attribute), 327
- objects (shoop.testing.models.ExpensiveSwedenBehaviorComponent attribute), 327
- objects (shoop.testing.models.PaymentWithCheckoutPhase attribute), 328
- objects (shoop.testing.models.PseudoPaymentProcessor attribute), 329
- objects (shoop.xtheme.models.SavedViewConfig attribute), 366
- objects (shoop.xtheme.models.ThemeSettings attribute), 366
- old_module_data (shoop.core.models.PaymentMethod attribute), 206
- old_module_data (shoop.core.models.Service attribute), 225
- old_module_data (shoop.core.models.ShippingMethod attribute), 229
- OLD_VERSION (shoop.xtheme.models.SavedViewConfigStatus attribute), 365

- option_list (shoop.core.management.commands.shoop_show_order_stats.Command attribute), 186
- option_list (shoop.testing.management.commands.shoop_populate_mobile_commands attribute), 326
- options (shoop.core.models.Shop attribute), 230
- Order (class in shoop.core.models), 199
- order (shoop.admin.base.OrderSection attribute), 126
- order (shoop.admin.modules.orders.sections.ContentsOrderSection attribute), 71
- order (shoop.admin.modules.orders.sections.LogEntriesOrderSection attribute), 71
- order (shoop.admin.modules.orders.sections.PaymentOrderSection attribute), 71
- order (shoop.campaigns.models.campaigns.CouponUsage attribute), 159
- order (shoop.core.models.OrderLine attribute), 203
- order (shoop.core.models.Payment attribute), 206
- order (shoop.core.models.Shipment attribute), 228
- Order.DoesNotExist, 199
- Order.MultipleObjectsReturned, 199
- order_form_class (shoop.front.checkout.single_page.SingleCheckoutBasket attribute), 285
- order_line (shoop.core.models.OrderLineTax attribute), 204
- order_line_taxes (shoop.core.models.Tax attribute), 235
- order_lines (shoop.core.models.Product attribute), 213, 217
- order_lines (shoop.core.models.Supplier attribute), 234
- ORDER_REFERENCE (shoop.core.models.CounterType attribute), 195
- order_set (shoop.core.models.OrderStatus attribute), 205
- order_set (shoop.core.models.Shop attribute), 230
- orderable (shoop.front.basket.objects.BaseBasket attribute), 280
- OrderCompleteView (class in shoop.front.views.order), 292
- OrderCreateFullRefundView (class in shoop.admin.modules.orders.views), 68
- OrderCreateFullRefundView (class in shoop.admin.modules.orders.views.refund), 64
- OrderCreatePaymentView (class in shoop.admin.modules.orders.views), 67
- OrderCreatePaymentView (class in shoop.admin.modules.orders.views.payment), 63
- OrderCreateRefundView (class in shoop.admin.modules.orders.views), 68
- OrderCreateRefundView (class in shoop.admin.modules.orders.views.refund), 64
- OrderCreateShipmentView (class in shoop.admin.modules.orders.views), 68
- OrderCreateShipmentView (class in shoop.admin.modules.orders.views.shipment), 66
- OrderSettingsCommand (class in shoop.core.order_creator), 238
- OrderDetailView (class in shoop.admin.modules.orders.views), 66
- OrderDetailView (class in shoop.admin.modules.orders.views.detail), 61
- OrderDetailView (class in shoop.front.apps.personal_order_history.views), 273
- OrderEditView (class in shoop.admin.modules.orders.views), 67
- OrderEditView (class in shoop.admin.modules.orders.views.edit), 62
- orderer (shoop.core.models.Order attribute), 202
- orderer (shoop.core.order_creator.OrderSource attribute), 240
- orderer (shoop.front.models.stored_basket.StoredBasket attribute), 287
- orderer (shoop.front.models.StoredBasket attribute), 288
- OrderSource (shoop.core.models.PersonContact attribute), 209
- orderer_orders (shoop.core.models.PersonContact attribute), 209
- OrderForm (class in shoop.front.checkout.single_page), 285
- OrderLine (class in shoop.core.models), 203
- OrderLine.DoesNotExist, 203
- OrderLine.MultipleObjectsReturned, 203
- OrderLineSerializer (class in shoop.core.api.orders), 178
- OrderLineSerializer.Meta (class in shoop.core.api.orders), 178
- OrderLineTax (class in shoop.core.models), 203
- OrderLineTax.DoesNotExist, 203
- OrderLineTax.MultipleObjectsReturned, 204
- OrderLineType (class in shoop.core.models), 204
- OrderListView (class in shoop.admin.modules.orders.views), 67
- OrderListView (class in shoop.admin.modules.orders.views.list), 63
- OrderListView (class in shoop.front.apps.personal_order_history.views), 273
- OrderLogEntry (class in shoop.core.models), 204
- OrderLogEntry.DoesNotExist, 204
- OrderLogEntry.MultipleObjectsReturned, 204
- OrderModifier (class in shoop.core.order_creator), 239
- OrderModule (class in shoop.admin.modules.orders), 71
- OrderReceived (class in shoop.front.notify_events), 294
- OrderRequiresVerificationView (class in shoop.front.views.order), 292
- OrderSection (class in shoop.admin.base), 125
- OrderSerializer (class in shoop.core.api.orders), 179

- OrderSerializer.Meta (class in shoop.core.api.orders), 179
- OrderSetStatusView (class in shoop.admin.modules.orders.views), 69
- OrderSetStatusView (class in shoop.admin.modules.orders.views.detail), 61
- OrderSource (class in shoop.core.order_creator), 239
- OrderSourceModifierModule (class in shoop.core.order_creator), 241
- OrderStatus (class in shoop.core.models), 205
- OrderStatus.DoesNotExist, 205
- OrderStatus.MultipleObjectsReturned, 205
- OrderStatusRole (class in shoop.core.models), 205
- OrderValueChartDashboardBlock (class in shoop.admin.modules.orders.dashboard), 69
- OrderViewMixin (class in shoop.front.apps.personal_order_history.views), 273
- OrderViewSet (class in shoop.core.api.orders), 179
- original_url (shoop.admin.base.Resolvable attribute), 125
- OTHER (shoop.core.models.Gender attribute), 197
- OTHER (shoop.core.models.OrderLineType attribute), 204
- OTHER (shoop.utils.analog.LogEntryKind attribute), 338
- override_current_theme_class() (in module shoop.xtheme.testing), 371
- override_provides() (in module shoop.apps.provides), 132
- owner (shoop.core.models.SavedAddress attribute), 223
- owner (shoop.core.models.Shop attribute), 230
- owner (shoop.testing.factories.ShopFactory attribute), 329
- ## P
- PACKAGE_PARENT (shoop.core.models.ProductMode attribute), 220
- PackageChildForm (class in shoop.admin.modules.products.forms), 80
- PackageChildForm (class in shoop.admin.modules.products.forms.package_forms), 78
- PackageChildFormSet (class in shoop.admin.modules.products.forms), 80
- PackageChildFormSet (class in shoop.admin.modules.products.forms.package_forms), 78
- PackageInstaller (class in shoop.addons.installer), 41
- PackageProductChoiceWidget (class in shoop.admin.forms.widgets), 46
- Page (class in shoop.simple_cms.models), 318
- Page.DoesNotExist, 318
- Page.MultipleObjectsReturned, 318
- PageEditView (class in shoop.simple_cms.admin_module.views), 317
- PageForm (class in shoop.simple_cms.admin_module.views), 317
- PageForm.Meta (class in shoop.simple_cms.admin_module.views), 317
- PageLinksConfigForm (class in shoop.simple_cms.plugins), 319
- PageLinksPlugin (class in shoop.simple_cms.plugins), 319
- PageListView (class in shoop.simple_cms.admin_module.views), 317
- PageQuerySet (class in shoop.simple_cms.models), 318
- PageTranslation (class in shoop.simple_cms.models), 319
- PageTranslation.DoesNotExist, 319
- PageTranslation.MultipleObjectsReturned, 319
- PageView (class in shoop.simple_cms.views), 320
- parent (shoop.core.models.Category attribute), 190
- parent (shoop.core.models.ProductPackageLink attribute), 220
- parent (shoop.simple_cms.models.Page attribute), 318
- parent_line (shoop.core.models.OrderLine attribute), 203
- parent_line (shoop.core.order_creator.SourceLine attribute), 242
- parent_name (shoop.admin.modules.taxes.views.CustomerTaxGroupEditView attribute), 111
- parent_name (shoop.admin.modules.taxes.views.edit.CustomerTaxGroupEditView attribute), 109
- parent_name (shoop.admin.modules.taxes.views.edit.TaxClassEditView attribute), 109
- parent_name (shoop.admin.modules.taxes.views.edit.TaxEditView attribute), 108
- parent_name (shoop.admin.modules.taxes.views.TaxClassEditView attribute), 110
- parent_name (shoop.admin.modules.taxes.views.TaxEditView attribute), 110
- parent_name (shoop.campaigns.admin_module.views.BasketCampaignEditView attribute), 141
- parent_name (shoop.campaigns.admin_module.views.CatalogCampaignEditView attribute), 140
- parent_name (shoop.campaigns.admin_module.views.CouponEditView attribute), 141
- parent_url (shoop.admin.modules.taxes.views.CustomerTaxGroupEditView attribute), 111
- parent_url (shoop.admin.modules.taxes.views.edit.CustomerTaxGroupEditView attribute), 109
- parent_url (shoop.admin.modules.taxes.views.edit.TaxClassEditView attribute), 109
- parent_url (shoop.admin.modules.taxes.views.edit.TaxEditView attribute), 108
- parent_url (shoop.admin.modules.taxes.views.TaxClassEditView attribute), 108

attribute), 110
parent_url (shoop.admin.modules.taxes.views.TaxEditView attribute), 110
parent_url (shoop.campaigns.admin_module.views.BasketCampaignEditView attribute), 141
parent_url (shoop.campaigns.admin_module.views.CatalogCampaignEditView attribute), 140
parent_url (shoop.campaigns.admin_module.views.CouponEditView attribute), 141
parse() (shoop.xtheme.parsing.LayoutPartExtension method), 368
parse() (shoop.xtheme.parsing.PlaceholderExtension method), 367
parse() (shoop.xtheme.parsing.PluginExtension method), 368
parse_constantlike() (in module shoop.xtheme.parsing), 367
parse_date() (in module shoop.utils.dates), 338
parse_decimal_string() (in module shoop.utils.numbers), 344
parse_time() (in module shoop.utils.dates), 339
PARTIALLY_PAID (shoop.core.models.PaymentStatus attribute), 208
PARTIALLY_SHIPPED (shoop.core.models.ShippingStatus attribute), 229
password (shoop.testing.factories.UserFactory attribute), 329
PasswordChangeForm (class in shoop.admin.modules.users.views.password), 112
Pattern (class in shoop.utils.patterns), 346
pattern_matches() (in module shoop.utils.patterns), 347
Payment (class in shoop.core.models), 205
PAYMENT (shoop.core.models.OrderLineType attribute), 204
Payment.DoesNotExist, 205
Payment.MultipleObjectsReturned, 205
payment_data (shoop.core.models.Order attribute), 202
payment_method (shoop.core.models.Order attribute), 202
payment_method (shoop.core.order_creator.OrderSource attribute), 240
payment_methods (shoop.core.models.ShopProduct attribute), 231
payment_orders (shoop.core.models.PaymentMethod attribute), 207
payment_processor (shoop.core.models.PaymentMethod attribute), 207
payment_products (shoop.core.models.PaymentMethod attribute), 207
payment_status (shoop.core.models.Order attribute), 202
PaymentMethod (class in shoop.core.models), 206
PaymentMethod.DoesNotExist, 206
PaymentMethod.MultipleObjectsReturned, 206
paymentmethod_set (shoop.core.models.PaymentProcessor attribute), 208
paymentmethod_set (shoop.core.models.ServiceBehaviorComponent attribute), 225
paymentmethod_set (shoop.core.models.Shop attribute), 208
paymentmethod_set (shoop.core.models.TaxClass attribute), 235
PaymentMethodBaseFormPart (class in shoop.admin.modules.services.base_form_part), 96
PaymentMethodDeleteView (class in shoop.admin.modules.services.views), 95
PaymentMethodEditView (class in shoop.admin.modules.services.views), 95
PaymentMethodForm (class in shoop.admin.modules.services.forms), 98
PaymentMethodForm.Meta (class in shoop.admin.modules.services.forms), 98
PaymentMethodListView (class in shoop.admin.modules.services.views), 95
PaymentMethodModule (class in shoop.admin.modules.services), 101
PaymentMethodPhase (class in shoop.front.checkout.methods), 284
PaymentOrderSection (class in shoop.admin.modules.orders.sections), 71
PaymentPhaseProvider (class in shoop.testing.simple_checkout_phase), 335
PaymentProcessor (class in shoop.core.models), 207
paymentprocessor (shoop.core.models.ServiceProvider attribute), 227
PaymentProcessor.DoesNotExist, 207
PaymentProcessor.MultipleObjectsReturned, 207
paymentprocessor_ptr (shoop.core.models.CustomPaymentProcessor attribute), 196
paymentprocessor_ptr (shoop.testing.models.PseudoPaymentProcessor attribute), 329
payments (shoop.core.models.Order attribute), 202
PaymentSerializer (class in shoop.core.api.orders), 179
PaymentSerializer.Meta (class in shoop.core.api.orders), 179
PaymentStatus (class in shoop.core.models), 208
PaymentUrls (class in shoop.core.models), 208
PaymentWithCheckoutPhase (class in shoop.testing.models), 327
paymentwithcheckoutphase (shoop.core.models.CustomPaymentProcessor attribute), 196
PaymentWithCheckoutPhase.DoesNotExist, 328
PaymentWithCheckoutPhase.MultipleObjectsReturned, 328
PaymentWithCheckoutPhaseForm (class in shoop.testing.service_forms), 333

[PaymentWithCheckoutPhaseForm.Meta](#) (class in [phases](#) ([shoop.front.checkout.CheckoutProcess](#) attribute), [shoop.testing.service_forms](#)), 333
[percent\(\)](#) (in module [shoop.core.templatetags.shoop_common](#)), 256
[PercentageField](#) (class in [shoop.admin.forms.fields](#)), 45
[PermissionChangeFormBase](#) (class in [shoop.admin.modules.users.views.permissions](#)), 113
[PermissionGroupEditView](#) (class in [shoop.admin.modules.permission_groups.views](#)), 73
[PermissionGroupEditView](#) (class in [shoop.admin.modules.permission_groups.views.edit](#)), 72
[PermissionGroupForm](#) (class in [shoop.admin.modules.permission_groups.views.edit](#)), 72
[PermissionGroupForm.Meta](#) (class in [shoop.admin.modules.permission_groups.views.edit](#)), 72
[PermissionGroupListView](#) (class in [shoop.admin.modules.permission_groups.views](#)), 73
[PermissionGroupListView](#) (class in [shoop.admin.modules.permission_groups.views.list](#)), 73
[PermissionGroupModule](#) (class in [shoop.admin.modules.permission_groups](#)), 73
[PersistentCacheEntry](#) (class in [shoop.core.models](#)), 208
[PersistentCacheEntry.DoesNotExist](#), 208
[PersistentCacheEntry.MultipleObjectsReturned](#), 208
[PersonContact](#) (class in [shoop.core.models](#)), 208
[personcontact](#) ([shoop.core.models.Contact](#) attribute), 193
[PersonContact.DoesNotExist](#), 209
[PersonContact.MultipleObjectsReturned](#), 209
[PersonContactChoiceWidget](#) (class in [shoop.admin.forms.widgets](#)), 46
[PersonContactForm](#) (class in [shoop.front.apps.customer_information.forms](#)), 271
[PersonContactForm.Meta](#) (class in [shoop.front.apps.customer_information.forms](#)), 271
[phase_class](#) ([shoop.front.checkout.BasicServiceCheckoutPhaseProvider](#) attribute), 285
[phase_class](#) ([shoop.testing.simple_checkout_phase.PaymentPhaseProvider](#) attribute), 335
[phase_class](#) ([shoop.testing.simple_checkout_phase.ShipmentPhaseProvider](#) attribute), 335
[phase_specs](#) ([shoop.front.views.checkout.BaseCheckoutView](#) attribute), 292
[phases](#) ([shoop.front.checkout.CheckoutPhaseViewMixin](#) attribute), 286
[Phone](#) (class in [shoop.notify.typology](#)), 313
[Picotable](#) (class in [shoop.admin.utils.picotable](#)), 120
[picotable_class](#) ([shoop.admin.utils.picotable.PicotableViewMixin](#) attribute), 120
[PicotableView](#) (class in [shoop.admin.utils.views](#)), 123
[PicotableViewMixin](#) (class in [shoop.admin.utils.picotable](#)), 120
[pk](#) ([shoop.core.models.AnonymousContact](#) attribute), 187
[PlaceholderExtension](#) (class in [shoop.xtheme.parsing](#)), 367
[PlaceholderRenderer](#) (class in [shoop.xtheme.rendering](#)), 369
[Plugin](#) (class in [shoop.xtheme](#)), 373
[plugin_class](#) ([shoop.xtheme.layout.LayoutCell](#) attribute), 361
[plugin_name](#) ([shoop.xtheme.layout.LayoutCell](#) attribute), 362
[PluginExtension](#) (class in [shoop.xtheme.parsing](#)), 368
[PluginForm](#) (class in [shoop.xtheme.plugins.forms](#)), 353
[polymorphic_ctype](#) ([shoop.campaigns.models.basket_conditions.BasketCondition](#) attribute), 146
[polymorphic_ctype](#) ([shoop.campaigns.models.basket_effects.BasketDiscountEffect](#) attribute), 153
[polymorphic_ctype](#) ([shoop.campaigns.models.basket_line_effects.BasketLineEffect](#) attribute), 155
[polymorphic_ctype](#) ([shoop.campaigns.models.BasketCondition](#) attribute), 173
[polymorphic_ctype](#) ([shoop.campaigns.models.BasketDiscountEffect](#) attribute), 172
[polymorphic_ctype](#) ([shoop.campaigns.models.BasketLineEffect](#) attribute), 170
[polymorphic_ctype](#) ([shoop.campaigns.models.catalog_filters.CatalogFilter](#) attribute), 162
[polymorphic_ctype](#) ([shoop.campaigns.models.CatalogFilter](#) attribute), 175
[polymorphic_ctype](#) ([shoop.campaigns.models.context_conditions.ContextCondition](#) attribute), 165
[polymorphic_ctype](#) ([shoop.campaigns.models.ContextCondition](#) attribute), 176
[polymorphic_ctype](#) ([shoop.campaigns.models.product_effects.ProductDiscountEffect](#) attribute), 167
[polymorphic_ctype](#) ([shoop.campaigns.models.ProductDiscountEffect](#) attribute), 174
[polymorphic_ctype](#) ([shoop.core.models.Contact](#) attribute), 193
[polymorphic_ctype](#) ([shoop.core.models.PolymorphicShoopModel](#) attribute), 210
[polymorphic_ctype](#) ([shoop.core.models.PolymorphicTranslatableShoopModel](#) attribute), 210
[polymorphic_ctype](#) ([shoop.core.models.ServiceBehaviorComponent](#) attribute), 225

polymorphic_ctype (shoop.core.models.ServiceProvider attribute), 227	(shoop.campaigns.models.catalog_filters.CatalogFilter attribute), 162
polymorphic_primary_key_name (shoop.campaigns.models.basket_conditions.BasketCondition attribute), 146	polymorphic_primary_key_name (shoop.campaigns.models.catalog_filters.CategoryFilter attribute), 164
polymorphic_primary_key_name (shoop.campaigns.models.basket_conditions.BasketMaxTotalAmountCondition attribute), 149	polymorphic_primary_key_name (shoop.campaigns.models.catalog_filters.ProductFilter attribute), 163
polymorphic_primary_key_name (shoop.campaigns.models.basket_conditions.BasketMaxTotalProductAmountCondition attribute), 148	polymorphic_primary_key_name (shoop.campaigns.models.catalog_filters.ProductTypeFilter attribute), 162
polymorphic_primary_key_name (shoop.campaigns.models.basket_conditions.BasketTotalAmountCondition attribute), 147	polymorphic_primary_key_name (shoop.campaigns.models.CatalogFilter attribute), 175
polymorphic_primary_key_name (shoop.campaigns.models.basket_conditions.BasketTotalProductAmountCondition attribute), 147	polymorphic_primary_key_name (shoop.campaigns.models.context_conditions.ContactCondition attribute), 167
polymorphic_primary_key_name (shoop.campaigns.models.basket_conditions.ContactBasketCondition attribute), 152	polymorphic_primary_key_name (shoop.campaigns.models.context_conditions.ContactGroupCondition attribute), 166
polymorphic_primary_key_name (shoop.campaigns.models.basket_conditions.ContactGroupBasketCondition attribute), 151	polymorphic_primary_key_name (shoop.campaigns.models.context_conditions.ContextCondition attribute), 165
polymorphic_primary_key_name (shoop.campaigns.models.basket_conditions.ProductsInBasketCondition attribute), 150	polymorphic_primary_key_name (shoop.campaigns.models.ContextCondition attribute), 176
polymorphic_primary_key_name (shoop.campaigns.models.basket_effects.BasketDiscountAmountEffect attribute), 153	polymorphic_primary_key_name (shoop.campaigns.models.product_effects.ProductDiscountAmountEffect attribute), 168
polymorphic_primary_key_name (shoop.campaigns.models.basket_effects.BasketDiscountEffect attribute), 153	polymorphic_primary_key_name (shoop.campaigns.models.product_effects.ProductDiscountEffect attribute), 167
polymorphic_primary_key_name (shoop.campaigns.models.basket_effects.BasketDiscountPercentageEffect attribute), 154	polymorphic_primary_key_name (shoop.campaigns.models.product_effects.ProductDiscountPercentageEffect attribute), 169
polymorphic_primary_key_name (shoop.campaigns.models.basket_line_effects.BasketLineEffect attribute), 155	polymorphic_primary_key_name (shoop.campaigns.models.ProductDiscountEffect attribute), 174
polymorphic_primary_key_name (shoop.campaigns.models.basket_line_effects.DiscountFromProduct attribute), 157	polymorphic_primary_key_name (shoop.core.models.AnonymousContact attribute), 187
polymorphic_primary_key_name (shoop.campaigns.models.basket_line_effects.FreeProductLine attribute), 156	polymorphic_primary_key_name (shoop.core.models.Carrier attribute), 189
polymorphic_primary_key_name (shoop.campaigns.models.BasketCondition attribute), 173	polymorphic_primary_key_name (shoop.core.models.CompanyContact attribute), 191
polymorphic_primary_key_name (shoop.campaigns.models.BasketDiscountEffect attribute), 172	polymorphic_primary_key_name (shoop.core.models.Contact attribute), 193
polymorphic_primary_key_name (shoop.campaigns.models.BasketLineEffect attribute), 170	polymorphic_primary_key_name (shoop.core.models.CustomCarrier attribute), 195
polymorphic_primary_key_name	polymorphic_primary_key_name (shoop.core.models.CustomPaymentProcessor attribute), 196

polymorphic_primary_key_name (shoop.core.models.FixedCostBehaviorComponent attribute), 196	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.basket_conditions.BasketTotalAmount attribute), 147
polymorphic_primary_key_name (shoop.core.models.GroupAvailabilityBehaviorComponent attribute), 194	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.basket_conditions.BasketTotalProduct attribute), 147
polymorphic_primary_key_name (shoop.core.models.PaymentProcessor attribute), 208	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.basket_conditions.ContactBasketCondi attribute), 152
polymorphic_primary_key_name (shoop.core.models.PersonContact attribute), 209	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.basket_conditions.ContactGroupBaske attribute), 151
polymorphic_primary_key_name (shoop.core.models.RoundingBehaviorComponent attribute), 222	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.basket_conditions.ProductsInBasketCo attribute), 150
polymorphic_primary_key_name (shoop.core.models.ServiceBehaviorComponent attribute), 226	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.basket_effects.BasketDiscountAmount attribute), 153
polymorphic_primary_key_name (shoop.core.models.ServiceProvider attribute), 227	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.basket_effects.BasketDiscountEffect attribute), 153
polymorphic_primary_key_name (shoop.core.models.StaffOnlyBehaviorComponent attribute), 233	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.basket_effects.BasketDiscountPercenta attribute), 154
polymorphic_primary_key_name (shoop.core.models.WaivingCostBehaviorComponent attribute), 236	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.basket_line_effects.BasketLineEffect attribute), 155
polymorphic_primary_key_name (shoop.core.models.WeightBasedPricingBehaviorComponent attribute), 237	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.basket_line_effects.DiscountFromProd attribute), 157
polymorphic_primary_key_name (shoop.core.models.WeightLimitsBehaviorComponent attribute), 237	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.basket_line_effects.FreeProductLine attribute), 156
polymorphic_primary_key_name (shoop.testing.models.CarrierWithCheckoutPhase attribute), 327	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.BasketCondition attribute), 173
polymorphic_primary_key_name (shoop.testing.models.ExpensiveSwedenBehaviorComponent attribute), 327	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.BasketDiscountEffect attribute), 172
polymorphic_primary_key_name (shoop.testing.models.PaymentWithCheckoutPhase attribute), 328	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.BasketLineEffect attribute), 170
polymorphic_primary_key_name (shoop.testing.models.PseudoPaymentProcessor attribute), 329	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.catalog_filters.CatalogFilter attribute), 162
polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.basket_conditions.BasketCondi attribute), 146	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.catalog_filters.CategoryFilter attribute), 164
polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.basket_conditions.BasketMaxTot attribute), 149	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.catalog_filters.ProductFilter attribute), 163
polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.basket_conditions.BasketMaxTot attribute), 148	polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.catalog_filters.ProductTypeFilter attribute), 162

- polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.CatalogFilter attribute), 175
- polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.context_conditions.ContextCondition attribute), 167
- polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.context_conditions.ContextCondition attribute), 166
- polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.context_conditions.ContextCondition attribute), 165
- polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.ContextCondition attribute), 176
- polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.product_effects.ProductDiscountEffect attribute), 168
- polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.product_effects.ProductDiscountEffect attribute), 167
- polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.product_effects.ProductDiscountEffect attribute), 169
- polymorphic_super_sub_accessors_replaced (shoop.campaigns.models.ProductDiscountEffect attribute), 174
- polymorphic_super_sub_accessors_replaced (shoop.core.models.AnonymousContact attribute), 187
- polymorphic_super_sub_accessors_replaced (shoop.core.models.Carrier attribute), 190
- polymorphic_super_sub_accessors_replaced (shoop.core.models.CompanyContact attribute), 191
- polymorphic_super_sub_accessors_replaced (shoop.core.models.Contact attribute), 193
- polymorphic_super_sub_accessors_replaced (shoop.core.models.CustomCarrier attribute), 195
- polymorphic_super_sub_accessors_replaced (shoop.core.models.CustomPaymentProcessor attribute), 196
- polymorphic_super_sub_accessors_replaced (shoop.core.models.FixedCostBehaviorComponent attribute), 197
- polymorphic_super_sub_accessors_replaced (shoop.core.models.GroupAvailabilityBehaviorComponent attribute), 194
- polymorphic_super_sub_accessors_replaced (shoop.core.models.PaymentProcessor attribute), 208
- polymorphic_super_sub_accessors_replaced (shoop.core.models.PersonContact attribute), 209
- polymorphic_super_sub_accessors_replaced (shoop.core.models.PolymorphicShoopModel attribute), 210
- polymorphic_super_sub_accessors_replaced (shoop.core.models.PolymorphicTranslatableShoopModel attribute), 210
- polymorphic_super_sub_accessors_replaced (shoop.core.models.RoundingBehaviorComponent attribute), 222
- polymorphic_super_sub_accessors_replaced (shoop.core.models.ServiceBehaviorComponent attribute), 226
- polymorphic_super_sub_accessors_replaced (shoop.core.models.ServiceProvider attribute), 227
- polymorphic_super_sub_accessors_replaced (shoop.core.models.StaffOnlyBehaviorComponent attribute), 233
- polymorphic_super_sub_accessors_replaced (shoop.core.models.WaivingCostBehaviorComponent attribute), 236
- polymorphic_super_sub_accessors_replaced (shoop.core.models.WeightBasedPricingBehaviorComponent attribute), 237
- polymorphic_super_sub_accessors_replaced (shoop.core.models.WeightLimitsBehaviorComponent attribute), 237
- polymorphic_super_sub_accessors_replaced (shoop.testing.models.CarrierWithCheckoutPhase attribute), 327
- polymorphic_super_sub_accessors_replaced (shoop.testing.models.ExpensiveSwedenBehaviorComponent attribute), 327
- polymorphic_super_sub_accessors_replaced (shoop.testing.models.PaymentWithCheckoutPhase attribute), 328
- polymorphic_super_sub_accessors_replaced (shoop.testing.models.PseudoPaymentProcessor attribute), 329
- PolymorphicShoopModel (class in shoop.core.models), 209
- PolymorphicShoopModel.Meta (class in shoop.core.models), 209
- PolymorphicTranslatableShoopModel (class in shoop.core.models), 210
- PolymorphicTranslatableShoopModel.Meta (class in shoop.core.models), 210
- populate() (shoop.api.urls.AutoRouter method), 131
- populate() (shoop.simple_cms.plugins.PageLinksConfigForm method), 319
- populate() (shoop.testing.mock_population.Populator method), 333
- populate() (shoop.xtheme.plugins.category_links.CategoryLinksConfigForm method), 333

- method), 353
- populate() (shoop.xtheme.plugins.forms.GenericPluginFormpost() (shoop.notify.admin_module.views.EditScriptContentView method), 354
- populate() (shoop.xtheme.plugins.forms.PluginForm post() (shoop.xtheme.views.editor.EditorView method), 358
- populate() (shoop.xtheme.plugins.social_media_links.SocialMediaLinkPluginForm in shoop.admin.toolbar), 128
- populate() (shoop.xtheme.views.forms.LayoutCellGeneralInfoForm postprocess_response() (shoop.front.basket.command_dispatcher.BasketCommandDispatcher), 278
- populate_core_api() (in module shoop.core.api), 180
- populate_customer_group_pricing_api() (in module shoop.customer_group_pricing.api), 264
- populate_form() (shoop.notify.admin_module.forms.ScriptItemEditForm save() (shoop.admin.modules.products.views.edit_media.ProductMediaForm), 85
- populate_if_required() (in module shoop.testing.mock_population), 333
- populate_if_required() (shoop.testing.mock_population.PopulationForm pre_master_save() (shoop.admin.modules.products.forms.BaseProductMediaForm), 79
- Populator (class in shoop.testing.mock_population), 333
- post (shoop.testing.factories.CategoryFactory attribute), 330
- post (shoop.testing.factories.ProductFactory attribute), 330
- post() (shoop.admin.modules.contacts.views.ContactDetailView pre_master_save() (shoop.admin.modules.products.forms.BaseProductMediaForm), 79
- post() (shoop.admin.modules.contacts.views.detail.ContactDetailForm pre_master_save() (shoop.admin.modules.products.forms.BaseProductMediaForm), 79
- post() (shoop.admin.modules.media.views.MediaBrowserView prepare_value() (shoop.admin.forms.fields.PercentageField method), 45
- post() (shoop.admin.modules.orders.views.detail.OrderSetStatusView preprocess_kwargs() (shoop.front.basket.command_dispatcher.BasketCommandDispatcher), 277
- post() (shoop.admin.modules.orders.views.log.NewLogEntryView previous_phase (shoop.front.checkout.CheckoutPhaseViewMixin attribute), 286
- post() (shoop.admin.modules.orders.views.NewLogEntryView Price (class in shoop.core.pricing), 245
- post() (shoop.admin.modules.orders.views.OrderSetStatusView PriceDisplay (class in shoop.core.pricing), 245
- post() (shoop.admin.modules.products.views.delete.ProductDeleteView PriceDisplayOptions (class in shoop.core.pricing), 245
- post() (shoop.admin.modules.products.views.edit_parent.ProductPropertyBaseView Priceful (class in shoop.core.pricing), 245
- post() (shoop.admin.modules.products.views.ProductDeleteView PriceInfo (class in shoop.core.pricing), 248
- post() (shoop.admin.modules.system.views.telemetry.TelemetryView PricePercentPropertyFilter (class in shoop.core.pricing), 248
- post() (shoop.admin.modules.users.views.detail.UserDetailView PriceProperty (class in shoop.core.pricing), 248
- post() (shoop.admin.modules.users.views.password.UserResetPasswordView PricePropertyBaseView (class in shoop.core.pricing), 248
- post() (shoop.admin.modules.users.views.UserDetailView PriceRangeDisplayFilter (class in shoop.core.pricing), 248
- post() (shoop.admin.modules.users.views.UserResetPasswordView PriceView (class in shoop.core.pricing), 248
- post() (shoop.notify.admin_module.views.editor.EditScriptContentView pricing_context_class (shoop.core.pricing.PricingModule attribute), 250
- primary_category (shoop.core.models.ShopProduct attribute), 231
- primary_image (shoop.core.models.Product attribute), 213, 217
- primary_image (shoop.core.models.ShopProduct attribute), 231

- tribute), 231
- primary_image_for_products (shoop.core.models.ProductMedia attribute), 219
- primary_image_for_shop_products (shoop.core.models.ProductMedia attribute), 219
- primary_products (shoop.core.models.Category attribute), 190
- primary_shop_products (shoop.core.models.Category attribute), 190
- PrintoutsAdminModule (class in shoop.order_printouts.admin_module), 316
- Priority (class in shoop.notify.enums), 309
- priority (shoop.admin.form_part.FormPart attribute), 127
- priority (shoop.admin.modules.categories.views.edit.CategoryBaseFormPart attribute), 49
- priority (shoop.admin.modules.contact_groups.views.forms.ContactGroupBaseFormPart attribute), 51
- priority (shoop.admin.modules.contact_groups.views.forms.ContactGroupFormPart attribute), 52
- priority (shoop.admin.modules.contacts.views.edit.ContactAddressFormPart attribute), 55
- priority (shoop.admin.modules.contacts.views.edit.ContactBaseFormPart attribute), 54
- priority (shoop.admin.modules.products.views.edit.ProductAttributeFormPart attribute), 83
- priority (shoop.admin.modules.products.views.edit.ProductBaseFormPart attribute), 83
- priority (shoop.admin.modules.products.views.edit.ProductImageMediaFormPart attribute), 84
- priority (shoop.admin.modules.products.views.edit.ProductMediaFormPart attribute), 83
- priority (shoop.admin.modules.products.views.edit.ShopProductFormPart attribute), 83
- priority (shoop.admin.modules.products.views.edit_package.ProductChildFormPart attribute), 86
- priority (shoop.admin.modules.products.views.edit_parent.ProductChildFormPart attribute), 87
- priority (shoop.admin.modules.products.views.edit_variation.VariationChildFormPart attribute), 88
- priority (shoop.admin.modules.products.views.edit_variation.VariationFormPart attribute), 88
- priority (shoop.admin.modules.services.base_form_part.ServiceBaseFormPart attribute), 96
- priority (shoop.admin.modules.shops.views.edit.ContactAddressFormPart attribute), 103
- priority (shoop.admin.modules.shops.views.edit.ShopBaseFormPart attribute), 102
- priority (shoop.campaigns.admin_module.form_parts.CampaignBaseFormPart attribute), 142
- priority (shoop.campaigns.admin_module.form_parts.SalesRangesFormPart attribute), 142
- priority (shoop.customer_group_pricing.admin_form_part.CustomerGroupSimpleFormPart attribute), 264
- priority (shoop.notify.models.Notification attribute), 306
- priority (shoop.notify.models.notification.Notification attribute), 304
- priority (shoop.simple_supplier.admin_module.forms.SimpleSupplierForm attribute), 321
- Problem, 339
- ProblemMiddleware (in module shoop.front.middleware), 293
- process() (shoop.front.checkout.addresses.AddressesPhase method), 283
- process() (shoop.front.checkout.CheckoutPhaseViewMixin method), 286
- process() (shoop.front.checkout.confirm.ConfirmPhase method), 283
- process() (shoop.front.checkout.empty.EmptyPhase method), 283
- process() (shoop.front.checkout.methods.MethodsPhase method), 284
- process() (shoop.front.checkout.single_page.SingleCheckoutPhase method), 285
- process() (shoop.testing.simple_checkout_phase.TestMethodCheckoutPhase method), 335
- process_exception() (shoop.core.middleware.ExceptionMiddleware method), 260
- process_payment_return_request() (shoop.admin.utils.picotable.Picotable method), 120
- process_payment_return_request() (shoop.core.models.CustomPaymentProcessor method), 196
- process_payment_return_request() (shoop.core.models.PaymentMethod method), 207
- process_payment_return_request() (shoop.core.models.PaymentProcessor method), 208
- process_payment_return_request() (shoop.testing.models.PseudoPaymentProcessor method), 329
- process_picotable() (shoop.admin.utils.picotable.PicotableViewMixin method), 120
- process_queryset() (shoop.admin.utils.picotable.Picotable method), 120
- process_request() (shoop.front.middleware.ShoopFrontMiddleware method), 294
- process_response() (shoop.front.middleware.ShoopFrontMiddleware method), 294
- process_schema() (shoop.core.management.commands.shoop_init.Initialize method), 186
- process_stock_adjustment() (in module shoop.simple_supplier.admin_module.views), 300

- 322
- process_thumbnailer_options() (in module shoop.front.templatetags.thumbnails), 290
- process_user() (shoop.admin.modules.users.views.password_reset_view), 113
- process_user() (shoop.admin.modules.users.views.UserResetPasswordView), 115
- process_user() (shoop.front.apps.auth.forms.RecoverPasswordForm method), 270
- process_val_datum() (shoop.admin.modules.products.forms.variable_variation_forms.VariationVariablesDataForm method), 79
- process_val_datum() (shoop.admin.modules.products.forms.VariationVariablesDataForm method), 82
- process_var_datum() (shoop.admin.modules.products.forms.variable_variation_forms.VariationVariablesDataForm method), 79
- process_var_datum() (shoop.admin.modules.products.forms.VariationVariablesDataForm method), 82
- process_view() (shoop.front.middleware.ShoopFrontMiddleware method), 294
- ProcessPaymentView (class in shoop.front.views.payment), 293
- Product (class in shoop.core.models), 210, 214
- product (shoop.core.models.OrderLine attribute), 203
- PRODUCT (shoop.core.models.OrderLineType attribute), 204
- product (shoop.core.models.ProductAttribute attribute), 218
- product (shoop.core.models.ProductMedia attribute), 219
- product (shoop.core.models.ProductVariationResult attribute), 221
- product (shoop.core.models.ProductVariationVariable attribute), 221
- product (shoop.core.models.ShipmentProduct attribute), 228
- product (shoop.core.models.ShopProduct attribute), 232
- product (shoop.core.models.SuppliedProduct attribute), 233
- product (shoop.core.utils.product_caching_object.ProductCachingObject attribute), 259
- product (shoop.customer_group_pricing.models.CgpPrice attribute), 265
- product (shoop.simple_supplier.models.StockAdjustment attribute), 323
- product (shoop.simple_supplier.models.StockCount attribute), 323
- Product.DoesNotExist, 210, 214
- Product.MultipleObjectsReturned, 210, 214
- product1 (shoop.core.models.ProductCrossSell attribute), 218
- product2 (shoop.core.models.ProductCrossSell attribute), 218
- product_count (shoop.core.order_creator.OrderSource attribute), 240
- product_id (shoop.core.utils.product_caching_object.ProductCachingObject attribute), 259
- product_ids (shoop.core.order_creator.OrderSource attribute), 240
- product_preview() (in module shoop.front.views), 336
- product_price() (in module shoop.themes.classic_gray.views), 336
- product_set (shoop.core.models.Manufacturer attribute), 223
- product_set (shoop.core.models.SalesUnit attribute), 223
- product_set (shoop.core.models.TaxClass attribute), 235
- product_types (shoop.campaigns.models.catalog_filters.ProductTypeFilter attribute), 218
- product_types (shoop.core.models.Attribute attribute), 218
- ProductAttribute (class in shoop.core.models), 218
- ProductAttribute.DoesNotExist, 218
- ProductAttribute.MultipleObjectsReturned, 218
- productattribute_set (shoop.core.models.Attribute attribute), 188
- ProductAttributeFormPart (class in shoop.admin.modules.products.views.edit), 83
- ProductAttributesForm (class in shoop.admin.modules.products.forms), 80
- ProductAttributesForm (class in shoop.admin.modules.products.forms.base_forms), 76
- ProductBaseForm (class in shoop.admin.modules.products.forms), 80
- ProductBaseForm (class in shoop.admin.modules.products.forms.base_forms), 75
- ProductBaseForm.Meta (class in shoop.admin.modules.products.forms), 80
- ProductBaseForm.Meta (class in shoop.admin.modules.products.forms.base_forms), 75
- ProductBaseFormPart (class in shoop.admin.modules.products.views.edit), 83
- ProductCachingObject (class in shoop.core.utils.product_caching_object), 259
- ProductChildBaseFormSet (class in shoop.admin.modules.products.forms.parent_forms), 78
- ProductChildrenBaseFormPart (class in shoop.admin.modules.products.views.edit_parent), 87
- ProductChildrenFormPart (class in shoop.admin.modules.products.views.edit_package), 87

[ProductChoiceWidget](#) (class in [shoop.admin.forms.widgets](#)), 46
[ProductCrossSell](#) (class in [shoop.core.models](#)), 218
[ProductCrossSell.DoesNotExist](#), 218
[ProductCrossSell.MultipleObjectsReturned](#), 218
[ProductCrossSellEditView](#) (class in [shoop.admin.modules.products.views](#)), 89
[ProductCrossSellEditView](#) (class in [shoop.admin.modules.products.views.edit_cross_sell](#)), 85
[ProductCrossSellForm](#) (class in [shoop.admin.modules.products.views.edit_cross_sell](#)), 84
[ProductCrossSellForm.Meta](#) (class in [shoop.admin.modules.products.views.edit_cross_sell](#)), 84
[ProductCrossSellFormSet](#) (class in [shoop.admin.modules.products.views.edit_cross_sell](#)), 84
[ProductCrossSellsPlugin](#) (class in [shoop.themes.classic_gray.plugins](#)), 337
[ProductCrossSellType](#) (class in [shoop.core.models](#)), 219
[ProductDeleteView](#) (class in [shoop.admin.modules.products.views](#)), 89
[ProductDeleteView](#) (class in [shoop.admin.modules.products.views.delete](#)), 83
[ProductDetailView](#) (class in [shoop.front.views.product](#)), 293
[ProductDiscountAmount](#) (class in [shoop.campaigns.models.product_effects](#)), 168
[productdiscountamount](#) ([shoop.campaigns.models.product_effects.ProductDiscountEffect](#) attribute), 168
[productdiscountamount](#) ([shoop.campaigns.models.ProductDiscountEffect](#) attribute), 174
[ProductDiscountAmount.DoesNotExist](#), 168
[ProductDiscountAmount.MultipleObjectsReturned](#), 168
[ProductDiscountAmountForm](#) (class in [shoop.campaigns.admin_module.forms](#)), 139
[ProductDiscountAmountForm.Meta](#) (class in [shoop.campaigns.admin_module.forms](#)), 139
[ProductDiscountEffect](#) (class in [shoop.campaigns.models](#)), 173
[ProductDiscountEffect](#) (class in [shoop.campaigns.models.product_effects](#)), 167
[ProductDiscountEffect.DoesNotExist](#), 167, 173
[ProductDiscountEffect.MultipleObjectsReturned](#), 167, 173
[productdiscounteffect_ptr](#) ([shoop.campaigns.models.product_effects.ProductDiscountEffect](#) attribute), 168
[productdiscounteffect_ptr](#) ([shoop.campaigns.models.product_effects.ProductDiscountEffect](#) attribute), 169
[ProductDiscountPercentage](#) (class in [shoop.campaigns.models.product_effects](#)), 168
[productdiscountpercentage](#) ([shoop.campaigns.models.product_effects.ProductDiscountEffect](#) attribute), 168
[ProductDiscountPercentage](#) (class in [shoop.campaigns.models.ProductDiscountEffect](#) attribute), 174
[ProductDiscountPercentage.DoesNotExist](#), 169
[ProductDiscountPercentage.MultipleObjectsReturned](#), 169
[ProductDiscountPercentageForm](#) (class in [shoop.campaigns.admin_module.forms](#)), 139
[ProductDiscountPercentageForm.Meta](#) (class in [shoop.campaigns.admin_module.forms](#)), 139
[ProductEditView](#) (class in [shoop.admin.modules.products.views](#)), 89
[ProductEditView](#) (class in [shoop.admin.modules.products.views.edit](#)), 84
[ProductFactory](#) (class in [shoop.testing.factories](#)), 330
[ProductFilter](#) (class in [shoop.campaigns.models.catalog_filters](#)), 162
[productfilter](#) ([shoop.campaigns.models.catalog_filters.CatalogFilter](#) attribute), 162
[productfilter](#) ([shoop.campaigns.models.CatalogFilter](#) attribute), 175
[ProductFilter.DoesNotExist](#), 163
[ProductFilter.MultipleObjectsReturned](#), 163
[productfilter_set](#) ([shoop.core.models.Product](#) attribute), 213, 217
[ProductFilterForm](#) (class in [shoop.campaigns.admin_module.forms](#)), 139
[ProductFilterForm.Meta](#) (class in [shoop.campaigns.admin_module.forms](#)), 139
[ProductHighlightPlugin](#) (class in [shoop.themes.classic_gray.plugins](#)), 337
[ProductImageMediaForm](#) (class in [shoop.admin.modules.products.forms](#)), 80
[ProductImageMediaForm](#) (class in [shoop.admin.modules.products.forms.base_forms](#)), 77
[ProductImageMediaFormPart](#) (class in [shoop.admin.modules.products.views.edit](#)), 77

84			ProductPackageView	(class	in	
ProductImageMediaFormSet	(class	in	shoop.admin.modules.products.views.edit_package),			
shoop.admin.modules.products.forms), 81			86			
ProductImageMediaFormSet	(class	in	ProductPackageViewToolbar	(class	in	
shoop.admin.modules.products.forms.base_forms),			shoop.admin.modules.products.views.edit_package),			
77			86			
ProductListView	(class	in	ProductParentBaseToolbar	(class	in	
shoop.admin.modules.products.views), 89			shoop.admin.modules.products.views.edit_parent),			
ProductListView	(class	in	ProductParentBaseView	(class	in	
shoop.admin.modules.products.views.list),			shoop.admin.modules.products.views.edit_parent),			
88			87			
ProductMedia	(class in shoop.core.models), 219		products (shoop.campaigns.models.basket_conditions.ProductsInBasketCon			
ProductMedia.DoesNotExist, 219			attribute), 150			
ProductMedia.MultipleObjectsReturned, 219			products (shoop.campaigns.models.basket_line_effects.DiscountFromProdu			
ProductMediaEditView	(class	in	attribute), 156			
shoop.admin.modules.products.views), 90			products (shoop.campaigns.models.basket_line_effects.FreeProductLine			
ProductMediaEditView	(class	in	attribute), 156			
shoop.admin.modules.products.views.edit_media),			products (shoop.campaigns.models.catalog_filters.ProductFilter			
86			attribute), 163			
ProductMediaForm	(class	in	products (shoop.core.models.ProductType attribute), 220			
shoop.admin.modules.products.forms), 81			products (shoop.core.models.Shipment attribute), 228			
ProductMediaForm	(class	in	products (shoop.front.models.stored_basket.StoredBasket			
shoop.admin.modules.products.forms.base_forms),			attribute), 287			
77			products (shoop.front.models.StoredBasket attribute),			
ProductMediaForm	(class	in	288			
shoop.admin.modules.products.views.edit_media),			products() (in module shoop.themes.classic_gray.views),			
85			336			
ProductMediaForm.Meta	(class	in	ProductSerializer	(class in shoop.core.api.products), 180		
shoop.admin.modules.products.views.edit_media),			ProductSerializer.Meta	(class in shoop.core.api.products),		
85			180			
ProductMediaFormPart	(class	in	ProductsInBasketCondition	(class	in	
shoop.admin.modules.products.views.edit),			shoop.campaigns.models.basket_conditions),			
83			149			
ProductMediaFormSet	(class	in	productsinbasketcondition			
shoop.admin.modules.products.forms), 81			(shoop.campaigns.models.basket_conditions.BasketCondition			
ProductMediaFormSet	(class	in	attribute), 146			
shoop.admin.modules.products.forms.base_forms),			productsinbasketcondition			
77			(shoop.campaigns.models.BasketCondition			
ProductMediaFormSet	(class	in	attribute), 173			
shoop.admin.modules.products.views.edit_media)			ProductsInBasketCondition.DoesNotExist, 150			
85			ProductsInBasketCondition.MultipleObjectsReturned,			
ProductMediaKind	(class in shoop.core.models), 219		150			
ProductMode	(class in shoop.core.models), 220		productsinbasketcondition_set			
ProductModule	(class in shoop.admin.modules.products),		(shoop.core.models.Product attribute), 213,			
91			217			
ProductNotOrderableProblem, 260			ProductsInBasketConditionForm	(class	in	
ProductNotVisibleProblem, 260			shoop.campaigns.admin_module.forms),			
ProductPackageLink	(class in shoop.core.models), 220		139			
ProductPackageLink.DoesNotExist, 220			ProductsInBasketConditionForm.Meta	(class	in	
ProductPackageLink.MultipleObjectsReturned, 220			shoop.campaigns.admin_module.forms),			
ProductPackageView	(class	in	140			
shoop.admin.modules.products.views), 90			ProductStockStatus	(class in shoop.core.stocks), 250		
			ProductType	(class in shoop.core.models), 220		

[ProductType.DoesNotExist](#), 220

[ProductType.MultipleObjectsReturned](#), 220

[ProductTypeEditView](#) (class in [shoop.admin.modules.product_types.views](#)), 74

[ProductTypeEditView](#) (class in [shoop.admin.modules.product_types.views.edit](#)), 74

[ProductTypeFactory](#) (class in [shoop.testing.factories](#)), 329

[ProductTypeFilter](#) (class in [shoop.campaigns.models.catalog_filters](#)), 162

[producttypefilter](#) ([shoop.campaigns.models.catalog_filters.CatalogFilter](#) attribute), 162

[producttypefilter](#) ([shoop.campaigns.models.CatalogFilter](#) attribute), 175

[ProductTypeFilter.DoesNotExist](#), 162

[ProductTypeFilter.MultipleObjectsReturned](#), 162

[producttypefilter_set](#) ([shoop.core.models.ProductType](#) attribute), 220

[ProductTypeFilterForm](#) (class in [shoop.campaigns.admin_module.forms](#)), 140

[ProductTypeFilterForm.Meta](#) (class in [shoop.campaigns.admin_module.forms](#)), 140

[ProductTypeForm](#) (class in [shoop.admin.modules.product_types.views.edit](#)), 74

[ProductTypeForm.Meta](#) (class in [shoop.admin.modules.product_types.views.edit](#)), 74

[ProductTypeListView](#) (class in [shoop.admin.modules.product_types.views](#)), 75

[ProductTypeListView](#) (class in [shoop.admin.modules.product_types.views.list](#)), 74

[ProductTypeModule](#) (class in [shoop.admin.modules.product_types](#)), 75

[ProductVariationLinkStatus](#) (class in [shoop.core.models](#)), 220

[ProductVariationResult](#) (class in [shoop.core.models](#)), 221

[ProductVariationResult.DoesNotExist](#), 221

[ProductVariationResult.MultipleObjectsReturned](#), 221

[ProductVariationVariable](#) (class in [shoop.core.models](#)), 221

[ProductVariationVariable.DoesNotExist](#), 221

[ProductVariationVariable.MultipleObjectsReturned](#), 221

[ProductVariationVariableValue](#) (class in [shoop.core.models](#)), 221

[ProductVariationVariableValue.DoesNotExist](#), 221

[ProductVariationVariableValue.MultipleObjectsReturned](#), 221

[ProductVariationView](#) (class in [shoop.admin.modules.products.views](#)), 90

[ProductVariationView](#) (class in [shoop.admin.modules.products.views.edit_variation](#)), 88

[ProductVariationViewToolbar](#) (class in [shoop.admin.modules.products.views.edit_variation](#)), 88

[ProductViewSet](#) (class in [shoop.core.api.products](#)), 180

[ProductVisibility](#) (class in [shoop.core.models](#)), 222

[profit_center](#) ([shoop.testing.factories.ProductFactory](#) attribute), 330

[ProtectedFields](#) ([shoop.core.models.Shop](#) attribute), 230

[ProtectedFieldsMixin](#) (class in [shoop.core.utils.form_mixins](#)), 257

[provide_category](#) ([shoop.notify.Action](#) attribute), 314

[provide_category](#) ([shoop.notify.base.Action](#) attribute), 308

[provide_category](#) ([shoop.notify.base.Base](#) attribute), 307

[provide_category](#) ([shoop.notify.base.Condition](#) attribute), 308

[provide_category](#) ([shoop.notify.base.Event](#) attribute), 307

[provide_category](#) ([shoop.notify.base.ScriptItem](#) attribute), 307

[provide_category](#) ([shoop.notify.Condition](#) attribute), 315

[provide_category](#) ([shoop.notify.Event](#) attribute), 315

[provider](#) ([shoop.core.models.Service](#) attribute), 225

[provider_attr](#) ([shoop.core.models.PaymentMethod](#) attribute), 207

[provider_attr](#) ([shoop.core.models.ShippingMethod](#) attribute), 229

[provides](#) ([shoop.admin.ShoopAdminAppConfig](#) attribute), 130

[provides](#) ([shoop.apps.AppConfig](#) attribute), 135

[provides](#) ([shoop.campaigns.apps.CampaignAppConfig](#) attribute), 177

[provides](#) ([shoop.core.ShoopCoreAppConfig](#) attribute), 264

[provides](#) ([shoop.customer_group_pricing.CustomerGroupPricingAppConfig](#) attribute), 265

[provides](#) ([shoop.default_tax.AppConfig](#) attribute), 268

[provides](#) ([shoop.front.apps.auth.AuthAppConfig](#) attribute), 271

[provides](#) ([shoop.front.apps.customer_information.AppConfig](#) attribute), 273

[provides](#) ([shoop.front.apps.personal_order_history.AppConfig](#) attribute), 273

[provides](#) ([shoop.front.apps.registration.RegistrationAppConfig](#) attribute), 275

[provides](#) ([shoop.front.apps.simple_order_notification.SimpleOrderNotificationAppConfig](#) attribute), 276

[provides](#) ([shoop.front.apps.simple_search.SimpleSearchAppConfig](#) attribute), 277

- provides (shoop.front.ShoopFrontAppConfig attribute), 296
- provides (shoop.guide.apps.AppConfig attribute), 296
- provides (shoop.order_printouts.AppConfig attribute), 316
- provides (shoop.simple_cms.AppConfig attribute), 321
- provides (shoop.simple_supplier.ShoopSimpleSupplierAppConfig attribute), 325
- provides (shoop.testing.ShoopTestingAppConfig attribute), 336
- provides (shoop.themes.classic_gray.ClassicGrayThemeAppConfig attribute), 337
- PseudoPaymentProcessor (class in shoop.testing.models), 328
- pseudopaymentprocessor (shoop.core.models.PaymentProcessor attribute), 208
- PseudoPaymentProcessor.DoesNotExist, 328
- PseudoPaymentProcessor.MultipleObjectsReturned, 328
- PseudoPaymentProcessorForm (class in shoop.testing.service_forms), 333
- PseudoPaymentProcessorForm.Meta (class in shoop.testing.service_forms), 333
- PUBLIC (shoop.xtheme.models.SavedViewConfigStatus attribute), 365
- public_name (shoop.campaigns.models.BasketCampaign attribute), 171
- public_name (shoop.campaigns.models.campaigns.BasketCampaign attribute), 159
- public_name (shoop.campaigns.models.campaigns.CatalogCampaign attribute), 158
- public_name (shoop.campaigns.models.CatalogCampaign attribute), 174
- public_name (shoop.core.models.Shop attribute), 230
- publish() (shoop.xtheme.models.SavedViewConfig method), 366
- publish() (shoop.xtheme.view_config.ViewConfig method), 373
- purchasable (shoop.testing.factories.ShopProductFactory attribute), 330
- purchase_price (shoop.simple_supplier.models.StockAdjustment attribute), 323
- Q**
- quantity (shoop.core.order_creator.SourceLine attribute), 242
- quantity (shoop.core.pricing.PriceInfo attribute), 248
- quantity_step (shoop.core.models.SalesUnit attribute), 223
- quantity_step (shoop.core.models.ShopProduct attribute), 232
- QuantityField (class in shoop.core.fields), 184
- quantize() (shoop.utils.numbers.UnittedDecimal method), 345
- queryset (shoop.core.api.orders.OrderViewSet attribute), 179
- queryset (shoop.core.api.products.ProductViewSet attribute), 180
- queryset (shoop.core.api.products.ShopProductViewSet attribute), 180
- R**
- raise_if_not_orderable() (shoop.core.models.ShopProduct method), 232
- raise_if_not_visible() (shoop.core.models.ShopProduct method), 232
- random_title() (in module shoop.testing.text_data), 336
- RandomImageGenerator (class in shoop.testing.image_generator), 332
- RangeFilter (class in shoop.admin.utils.picotable), 119
- ranges (shoop.core.models.WeightBasedPricingBehaviorComponent attribute), 237
- rate (shoop.core.taxing.LineTax attribute), 253
- ready() (shoop.admin.ShoopAdminAppConfig method), 131
- ready() (shoop.api.ShoopApiAppConfig method), 132
- ready() (shoop.campaigns.apps.CampaignAppConfig method), 177
- ready() (shoop.front.apps.registration.RegistrationAppConfig method), 275
- ready() (shoop.front.apps.simple_order_notification.SimpleOrderNotification method), 276
- ready() (shoop.front.ShoopFrontAppConfig method), 296
- ready_or_none() (in module shoop.core.utils.users), 259
- recipient (shoop.notify.models.Notification attribute), 306
- recipient (shoop.notify.models.notification.Notification attribute), 304
- recipient_type (shoop.notify.models.Notification attribute), 306
- recipient_type (shoop.notify.models.notification.Notification attribute), 304
- RecipientType (class in shoop.notify.enums), 309
- RECOMMENDED (shoop.core.models.ProductCrossSellType attribute), 219
- RecoverPasswordCompleteView (class in shoop.front.apps.auth.views), 271
- RecoverPasswordConfirmView (class in shoop.front.apps.auth.views), 270
- RecoverPasswordForm (class in shoop.front.apps.auth.forms), 269
- RecoverPasswordSentView (class in shoop.front.apps.auth.views), 271
- RecoverPasswordView (class in shoop.front.apps.auth.views), 270
- refresh_on_logout() (shoop.front.middleware.ShoopFrontMiddleware class method), 294

`refresh_on_user_change()` (shoop.front.middleware.ShoopFrontMiddleware class method), 294

`REFUND` (shoop.core.models.OrderLineType attribute), 204

`RefundExceedsAmountException`, 260

`RefundForm` (class in shoop.admin.modules.orders.views.refund), 63

`register()` (in module shoop.admin.module_registry), 127

`register()` (shoop.core.fields.tagged_json.TagRegistry method), 183

`registration_complete()` (in module shoop.front.apps.registration.views), 274

`RegistrationAppConfig` (class in shoop.front.apps.registration), 275

`RegistrationNoActivationView` (class in shoop.front.apps.registration.views), 274

`RegistrationView` (class in shoop.front.apps.registration.views), 274

`RegistrationViewMixin` (class in shoop.front.apps.registration.views), 274

`RegistrationWithActivationView` (class in shoop.front.apps.registration.views), 274

`registry` (shoop.core.fields.tagged_json.TaggedJSONEncoder attribute), 184

`RELATED` (shoop.core.models.ProductCrossSellType attribute), 219

`RelaxedModelChoiceField` (class in shoop.utils.fields), 340

`reload_apps()` (in module shoop.apps.settings), 133

`ReloadMethod` (class in shoop.addons.reloader), 41

`ReloadMethodForm` (class in shoop.addons.admin_module.views.reload), 38

`ReloadView` (class in shoop.addons.admin_module.views), 40

`ReloadView` (class in shoop.addons.admin_module.views.reload), 38

`remove_code()` (shoop.core.order_creator.OrderSource method), 240

`remove_code()` (shoop.front.basket.objects.BaseBasket method), 279

`render()` (shoop.admin.forms.widgets.BasePopupChoiceWidget method), 45

`render()` (shoop.admin.toolbar.BaseActionButton method), 128

`render()` (shoop.admin.toolbar.ButtonGroup method), 130

`render()` (shoop.admin.toolbar.DropdownActionButton method), 129

`render()` (shoop.admin.toolbar.DropdownDivider method), 129

`render()` (shoop.admin.toolbar.DropdownHeader method), 130

`render()` (shoop.admin.toolbar.DropdownItem method), 129

`render()` (shoop.admin.toolbar.JavaScriptActionButton method), 128

`render()` (shoop.admin.toolbar.PostActionButton method), 129

`render()` (shoop.admin.toolbar.Toolbar method), 130

`render()` (shoop.admin.toolbar.URLActionButton method), 128

`render()` (shoop.front.checkout.methods.MethodWidget method), 284

`render()` (shoop.notify.template.Template method), 311

`render()` (shoop.xtheme.engine.XthemeTemplate method), 360

`render()` (shoop.xtheme.layout.LayoutCell method), 362

`render()` (shoop.xtheme.Plugin method), 374

`render()` (shoop.xtheme.plugins.snippets.SnippetsPlugin method), 355

`render()` (shoop.xtheme.plugins.text.TextPlugin method), 356

`render()` (shoop.xtheme.plugins.widgets.TranslatableFieldWidget method), 356

`render()` (shoop.xtheme.plugins.widgets.XThemeModelChoiceWidget method), 356

`render()` (shoop.xtheme.rendering.PlaceholderRenderer method), 369

`render()` (shoop.xtheme.TemplatedPlugin method), 375

`render_dropdown()` (shoop.admin.toolbar.DropdownActionButton method), 129

`render_first_match()` (shoop.notify.template.Template method), 311

`render_html_to_pdf()` (in module shoop.order_printouts.pdf_export), 316

`render_in_context()` (in module shoop.notify.template), 311

`render_label()` (shoop.admin.toolbar.BaseActionButton method), 128

`render_placeholder()` (in module shoop.xtheme.rendering), 369

`render_price_property()` (in module shoop.core.utils.price_display), 258

`render_resources()` (shoop.xtheme.resources.ResourceContainer method), 370

`render_text()` (shoop.admin.forms.widgets.BasePopupChoiceWidget method), 45

`render_to_response()` (shoop.front.views.order.OrderCompleteView method), 292

`render_to_string()` (shoop.admin.toolbar.Toolbar method), 130

`replace_modules()` (in module shoop.admin.module_registry), 127

`request` (shoop.admin.form_part.FormPartsViewMixin attribute), 127

`request` (shoop.admin.form_part.SaveFormPartsMixin attribute), 127

`request` (shoop.front.checkout.CheckoutPhaseViewMixin

- attribute), 286
 - required_context_variables (shoop.themes.classic_gray.plugins.ProductCrossSellsPlugin attribute), 337
 - required_context_variables (shoop.xtheme.Plugin attribute), 374
 - required_installed_apps (shoop.admin.ShoopAdminAppConfig attribute), 130
 - required_installed_apps (shoop.api.ShoopApiAppConfig attribute), 132
 - required_installed_apps (shoop.apps.AppConfig attribute), 134
 - required_installed_apps (shoop.core.ShoopCoreAppConfig attribute), 264
 - required_installed_apps (shoop.front.apps.registration.RegistrationAppConfig attribute), 275
 - reset() (shoop.front.checkout.CheckoutPhaseViewMixin method), 286
 - reset() (shoop.front.checkout.CheckoutProcess method), 286
 - Resolvable (class in shoop.admin.base), 125
 - resolve() (in module shoop.utils.properties), 348
 - resolve() (shoop.core.models.ProductVariationResult class method), 221
 - ResourceContainer (class in shoop.xtheme.resources), 370
 - result (shoop.core.models.ProductVariationResult attribute), 221
 - result_limit (shoop.admin.views.select.MultiselectAjaxView attribute), 124
 - retry_request() (in module shoop.utils.http), 342
 - revert() (shoop.xtheme.models.SavedViewConfig method), 366
 - revert() (shoop.xtheme.view_config.ViewConfig method), 373
 - RingImageGenerator (class in shoop.testing.image_generator), 332
 - role (shoop.core.models.OrderStatus attribute), 205
 - role (shoop.core.models.SavedAddress attribute), 223
 - round() (shoop.core.models.SalesUnit method), 223
 - ROUND_DOWN (shoop.core.models.RoundingMode attribute), 222
 - ROUND_HALF_DOWN (shoop.core.models.RoundingMode attribute), 222
 - ROUND_HALF_UP (shoop.core.models.RoundingMode attribute), 222
 - ROUND_UP (shoop.core.models.RoundingMode attribute), 222
 - rounded_minimum_purchase_quantity (shoop.core.models.ShopProduct attribute), 232
 - RoundingBehaviorComponent (class in shoop.core.models), 222
 - roundingbehaviorcomponent (shoop.core.models.ServiceBehaviorComponent attribute), 226
 - RoundingBehaviorComponent.DoesNotExist, 222
 - RoundingBehaviorComponent.MultipleObjectsReturned, 222
 - RoundingBehaviorComponentForm (class in shoop.admin.modules.services.forms), 100
 - RoundingBehaviorComponentForm.Meta (class in shoop.admin.modules.services.forms), 100
 - RoundingMode (class in shoop.core.models), 222
 - row_class (shoop.xtheme.layout.Layout attribute), 363
 - rules_form_part_class (shoop.campaigns.admin_module.views.BasketCampaignFormPart class (shoop.campaigns.admin_module.views.CatalogCampaignFormPart attribute), 140
 - rules_match() (shoop.campaigns.models.BasketCampaign method), 171
 - rules_match() (shoop.campaigns.models.campaigns.BasketCampaign method), 158
 - rules_match() (shoop.campaigns.models.campaigns.CatalogCampaign method), 158
 - rules_match() (shoop.campaigns.models.CatalogCampaign method), 174
 - run() (shoop.core.management.commands.shoop_init.Initializer method), 186
 - run() (shoop.notify.base.Event method), 307
 - run() (shoop.notify.Event method), 315
 - run_event() (in module shoop.notify.runner), 310
- ## S
- safe_get_first() (shoop.admin.modules.orders.json_order_creator.JsonOrder static method), 70
 - safe_json() (in module shoop.core.telemetry), 263
 - sales_unit (shoop.core.models.Product attribute), 213, 217
 - sales_unit (shoop.testing.factories.ProductFactory attribute), 330
 - SalesRangeQuerySet (class in shoop.campaigns.models.contact_group_sales_ranges), 164
 - SalesRangesForm (class in shoop.campaigns.admin_module.form_parts), 142
 - SalesRangesForm.Meta (class in shoop.campaigns.admin_module.form_parts), 142
 - SalesRangesFormPart (class in shoop.campaigns.admin_module.form_parts), 142
 - SalesUnit (class in shoop.core.models), 222
 - SalesUnit.DoesNotExist, 222
 - SalesUnit.MultipleObjectsReturned, 222

- [save\(\)](#) (shoop.xtheme.forms.GenericThemeForm method), 361
[save\(\)](#) (shoop.xtheme.views.forms.LayoutCellFormGroup method), 359
[save\(\)](#) (shoop.xtheme.views.forms.LayoutCellGeneralInfoForm method), 358
[save_default_placeholder_layout\(\)](#) (shoop.xtheme.view_config.ViewConfig method), 373
[save_form\(\)](#) (shoop.admin.modules.users.views.detail.UserDetailView method), 112
[save_form\(\)](#) (shoop.admin.modules.users.views.UserDetailView method), 114
[save_form\(\)](#) (shoop.admin.utils.views.CreateOrUpdateView method), 122
[save_form\(\)](#) (shoop.simple_cms.admin_module.views.PageEditView method), 317
[save_form_parts\(\)](#) (shoop.admin.form_part.SaveFormPartsMixin method), 127
[save_layout\(\)](#) (shoop.xtheme.views.editor.EditorView method), 358
[save_placeholder_layout\(\)](#) (shoop.xtheme.view_config.ViewConfig method), 373
[save_telemetry_submission\(\)](#) (in module shoop.core.telemetry), 263
[saved_addresses](#) (shoop.core.models.MutableAddress attribute), 199
[saved_view_config](#) (shoop.xtheme.view_config.ViewConfig attribute), 372
[SavedAddress](#) (class in shoop.core.models), 223
[SavedAddress.DoesNotExist](#), 223
[SavedAddress.MultipleObjectsReturned](#), 223
[savedaddress_set](#) (shoop.core.models.Contact attribute), 193
[SavedAddressRole](#) (class in shoop.core.models), 223
[SavedAddressStatus](#) (class in shoop.core.models), 224
[SavedViewConfig](#) (class in shoop.xtheme.models), 365
[SavedViewConfig.DoesNotExist](#), 366
[SavedViewConfig.MultipleObjectsReturned](#), 366
[SavedViewConfigQuerySet](#) (class in shoop.xtheme.models), 365
[SavedViewConfigStatus](#) (class in shoop.xtheme.models), 365
[SaveFormPartsMixin](#) (class in shoop.admin.form_part), 127
[schema\(\)](#) (in module shoop.core.management.commands.shoop_initialize), 186
[schemata](#) (shoop.core.management.commands.shoop_init.Initializer attribute), 186
[Script](#) (class in shoop.notify.models), 306
[Script](#) (class in shoop.notify.models.script), 305
[Script.DoesNotExist](#), 305, 306
[Script.MultipleObjectsReturned](#), 305, 306
[script_item_editor\(\)](#) (in module shoop.notify.admin_module.views), 300
[script_item_editor\(\)](#) (in module shoop.notify.admin_module.views.editor), 299
[ScriptAPI](#) (class in shoop.notify.admin_module.views.editor), 299
[ScriptEditView](#) (class in shoop.notify.admin_module.views), 300
[ScriptItemEditView](#) (class in shoop.notify.admin_module.views.edit), 299
[ScriptForm](#) (class in shoop.notify.admin_module.forms), 301
[ScriptForm.Meta](#) (class in shoop.notify.admin_module.forms), 301
[ScriptItem](#) (class in shoop.notify.base), 307
[ScriptItemEditForm](#) (class in shoop.notify.admin_module.forms), 301
[ScriptListView](#) (class in shoop.notify.admin_module.views), 300
[ScriptListView](#) (class in shoop.notify.admin_module.views.list), 300
[search_fields](#) (shoop.admin.views.select.MultiselectAjaxView attribute), 124
[searchable](#) (shoop.testing.factories.ShopProductFactory attribute), 330
[SEARCHABLE_FIELD](#) (shoop.core.models.AttributeVisibility attribute), 189
[SearchForm](#) (class in shoop.front.apps.simple_search.views), 276
[SearchResult](#) (class in shoop.admin.base), 125
[SearchView](#) (class in shoop.admin.views.search), 123
[SearchView](#) (class in shoop.front.apps.simple_search.views), 276
[Select2MultipleField](#) (class in shoop.admin.forms.fields), 45
[send_order_received_notification\(\)](#) (in module shoop.front.notify_events), 294
[send_simple_order_notification\(\)](#) (in module shoop.front.apps.simple_order_notification), 275
[SendEmail](#) (class in shoop.notify.actions), 298
[SendEmail](#) (class in shoop.notify.actions.email), 297
[serialize\(\)](#) (shoop.notify.base.ScriptItem method), 308
[serialize\(\)](#) (shoop.notify.script.Step method), 310
[serialize\(\)](#) (shoop.xtheme.layout.Layout method), 363
[serialize\(\)](#) (shoop.xtheme.layout.LayoutCell method), 362
[serialize\(\)](#) (shoop.xtheme.layout.LayoutRow method), 362
[serializer_class](#) (shoop.core.api.orders.OrderViewSet attribute), 179
[serializer_class](#) (shoop.core.api.products.ProductViewSet attribute), 179

attribute), 180

serializer_class (shoop.core.api.products.ShopProductViewSet attribute), 180

serializer_factory() (in module shoop.api.factories), 131

Service (class in shoop.core.models), 224

Service.Meta (class in shoop.core.models), 224

service_model (shoop.core.models.Carrier attribute), 190

service_model (shoop.core.models.PaymentProcessor attribute), 208

service_model (shoop.core.models.ServiceProvider attribute), 227

service_provider (shoop.admin.modules.services.forms.BaseMethodForm attribute), 97

service_provider_attr (shoop.admin.modules.services.forms.PaymentMethodForm attribute), 98

service_provider_attr (shoop.admin.modules.services.forms.ShippingMethodForm attribute), 98

service_provider_class (shoop.front.checkout.BasicServiceCheckoutPhaseEditView attribute), 285

service_provider_class (shoop.testing.simple_checkout_phase.PaymentPhaseProvider attribute), 335

service_provider_class (shoop.testing.simple_checkout_phase.ShipmentPhaseProvider attribute), 335

ServiceBaseFormPart (class in shoop.admin.modules.services.base_form_part), 96

ServiceBehaviorComponent (class in shoop.core.models), 225

ServiceBehaviorComponent.DoesNotExist, 225

ServiceBehaviorComponent.MultipleObjectsReturned, 225

servicebehaviorcomponent_ptr (shoop.core.models.FixedCostBehaviorComponent attribute), 197

servicebehaviorcomponent_ptr (shoop.core.models.GroupAvailabilityBehaviorComponent attribute), 194

servicebehaviorcomponent_ptr (shoop.core.models.RoundingBehaviorComponent attribute), 222

servicebehaviorcomponent_ptr (shoop.core.models.StaffOnlyBehaviorComponent attribute), 233

servicebehaviorcomponent_ptr (shoop.core.models.WaivingCostBehaviorComponent attribute), 236

servicebehaviorcomponent_ptr (shoop.core.models.WeightBasedPricingBehaviorComponent attribute), 237

servicebehaviorcomponent_ptr (shoop.core.models.WeightLimitsBehaviorComponent attribute), 237

servicebehaviorcomponent_ptr (shoop.testing.models.ExpensiveSwedenBehaviorComponent attribute), 279

ServiceCheckoutPhaseProvider (class in shoop.front.checkout), 286

ServiceChoice (class in shoop.core.models), 226

ServiceCost (class in shoop.core.models), 226

ServiceModule (class in shoop.admin.modules.services), 101

ServiceProvider (class in shoop.core.models), 226

ServiceProvider.DoesNotExist, 227

ServiceProvider.MultipleObjectsReturned, 227

serviceprovider_ptr (shoop.core.models.Carrier attribute), 190

serviceprovider_ptr (shoop.core.models.PaymentProcessor attribute), 208

ServiceProviderDeleteView (class in shoop.admin.modules.service_providers.views), 93

ServiceProviderEditView (class in shoop.admin.modules.service_providers.views), 93

ServiceProviderListView (class in shoop.admin.modules.service_providers.views), 93

ServiceProviderModule (class in shoop.admin.modules.service_providers), 94

set() (in module shoop.configuration), 377

set() (shoop.core.cache.impl.VersionedCache method), 181

set() (shoop.core.cache.VersionedCache method), 182

set() (shoop.notify.Context method), 315

set() (shoop.notify.script.Context method), 310

set_canceled() (shoop.core.models.Order method), 202

set_current() (shoop.xtheme.Theme method), 376

set_current_theme() (in module shoop.xtheme), 377

set_enabled_addons() (in module shoop.addons.manager), 41

set_for_request() (shoop.core.pricing.PriceDisplayOptions method), 245

set_fully_paid() (shoop.core.api.orders.OrderViewSet method), 179

set_id() (shoop.core.utils.model_caching_descriptor.ModelCachingDescriptor method), 257

set_layout_data() (shoop.xtheme.models.SavedViewConfig method), 366

set_model_from_cleaned_data() (shoop.admin.modules.contacts.views.edit.ContactBaseForm method), 54

set_object() (shoop.core.utils.model_caching_descriptor.ModelCachingDescriptor method), 257

set_opt_out() (in module shoop.core.telemetry), 263

set_quantity() (shoop.front.basket.objects.BasketLine method), 279

- `set_reverse_url()` (shoop.notify.models.Notification method), 306
- `set_reverse_url()` (shoop.notify.models.notification.Notification method), 305
- `set_serialized_steps()` (shoop.notify.models.Script method), 306
- `set_serialized_steps()` (shoop.notify.models.script.Script method), 305
- `set_setting()` (shoop.xtheme.Theme method), 376
- `set_settings()` (shoop.xtheme.Theme method), 376
- `set_steps()` (shoop.notify.models.Script method), 306
- `set_steps()` (shoop.notify.models.script.Script method), 305
- `SetDebugFlag` (class in shoop.notify.actions), 298
- `SetDebugFlag` (class in shoop.notify.actions.debug), 297
- `Setting` (class in shoop.apps.settings), 133
- `settings_obj` (shoop.xtheme.Theme attribute), 376
- `Setup` (class in shoop.utils.setup), 349
- `Shipment` (class in shoop.core.models), 227
- `shipment` (shoop.core.models.ShipmentProduct attribute), 228
- `Shipment.DoesNotExist`, 227
- `Shipment.MultipleObjectsReturned`, 227
- `ShipmentForm` (class in shoop.admin.modules.orders.views.shipment), 66
- `ShipmentFormModifier` (class in shoop.admin.modules.orders.views.shipment), 65
- `ShipmentPhaseProvider` (class in shoop.testing.simple_checkout_phase), 335
- `ShipmentProduct` (class in shoop.core.models), 228
- `ShipmentProduct.DoesNotExist`, 228
- `ShipmentProduct.MultipleObjectsReturned`, 228
- `shipments` (shoop.core.models.Order attribute), 202
- `shipments` (shoop.core.models.Product attribute), 213, 217
- `shipments` (shoop.core.models.Supplier attribute), 234
- `SHIPPED` (shoop.core.models.ShippingMode attribute), 229
- `SHIPPING` (shoop.core.models.OrderLineType attribute), 204
- `SHIPPING` (shoop.core.models.SavedAddressRole attribute), 224
- `shipping_address` (shoop.core.models.Order attribute), 202
- `shipping_address_form_class` (shoop.front.checkout.single_page.SingleCheckoutForm attribute), 285
- `shipping_data` (shoop.core.models.Order attribute), 202
- `shipping_method` (shoop.core.models.Order attribute), 202
- `shipping_method` (shoop.core.order_creator.OrderSource attribute), 240
- `shipping_methods` (shoop.core.models.ShopProduct attribute), 232
- `shipping_mode` (shoop.core.models.Product attribute), 213, 217
- `shipping_orders` (shoop.core.models.ImmutableAddress attribute), 198
- `shipping_orders` (shoop.core.models.ShippingMethod attribute), 229
- `shipping_products` (shoop.core.models.ShippingMethod attribute), 229
- `shipping_status` (shoop.core.models.Order attribute), 202
- `ShippingMethod` (class in shoop.core.models), 228
- `ShippingMethod.DoesNotExist`, 228
- `ShippingMethod.MultipleObjectsReturned`, 228
- `shippingmethod_set` (shoop.core.models.Carrier attribute), 190
- `shippingmethod_set` (shoop.core.models.ServiceBehaviorComponent attribute), 226
- `shippingmethod_set` (shoop.core.models.Shop attribute), 230
- `shippingmethod_set` (shoop.core.models.TaxClass attribute), 235
- `ShippingMethodBaseFormPart` (class in shoop.admin.modules.services.base_form_part), 96
- `ShippingMethodDeleteView` (class in shoop.admin.modules.services.views), 95
- `ShippingMethodEditView` (class in shoop.admin.modules.services.views), 95
- `ShippingMethodForm` (class in shoop.admin.modules.services.forms), 97
- `ShippingMethodForm.Meta` (class in shoop.admin.modules.services.forms), 98
- `ShippingMethodListView` (class in shoop.admin.modules.services.views), 95
- `ShippingMethodModule` (class in shoop.admin.modules.services), 101
- `ShippingMethodPhase` (class in shoop.front.checkout.methods), 284
- `ShippingMode` (class in shoop.core.models), 229
- `ShippingStatus` (class in shoop.core.models), 229
- `Shoop`, 18
- `shoop` (module), 378
- `Shoop Base`, 18
- `Shoop Core`, 18
- `shoop.addons` (module), 42
- `shoop.addons.admin_module` (module), 40
- `shoop.addons.admin_module.views` (module), 39
- `shoop.addons.admin_module.views.list` (module), 38
- `shoop.addons.admin_module.views.reload` (module), 38
- `shoop.addons.admin_module.views.upload` (module), 39
- `shoop.addons.installer` (module), 41
- `shoop.addons.manager` (module), 41
- `shoop.addons.reloader` (module), 41

- shoop.admin (module), 130
- shoop.admin.base (module), 124
- shoop.admin.breadcrumbs (module), 126
- shoop.admin.currencybound (module), 126
- shoop.admin.dashboard (module), 44
- shoop.admin.dashboard.blocks (module), 42
- shoop.admin.dashboard.charts (module), 43
- shoop.admin.dashboard.utils (module), 43
- shoop.admin.form_part (module), 127
- shoop.admin.forms (module), 46
- shoop.admin.forms.fields (module), 45
- shoop.admin.forms.widgets (module), 45
- shoop.admin.menu (module), 127
- shoop.admin.module_registry (module), 127
- shoop.admin.modules (module), 116
- shoop.admin.modules.attributes (module), 48
- shoop.admin.modules.attributes.views (module), 47
- shoop.admin.modules.attributes.views.edit (module), 47
- shoop.admin.modules.attributes.views.list (module), 47
- shoop.admin.modules.categories (module), 50
- shoop.admin.modules.categories.views (module), 49
- shoop.admin.modules.categories.views.edit (module), 48
- shoop.admin.modules.categories.views.list (module), 49
- shoop.admin.modules.contact_groups (module), 53
- shoop.admin.modules.contact_groups.views (module), 52
- shoop.admin.modules.contact_groups.views.delete (module), 50
- shoop.admin.modules.contact_groups.views.edit (module), 51
- shoop.admin.modules.contact_groups.views.forms (module), 51
- shoop.admin.modules.contact_groups.views.list (module), 52
- shoop.admin.modules.contacts (module), 57
- shoop.admin.modules.contacts.dashboard (module), 57
- shoop.admin.modules.contacts.views (module), 56
- shoop.admin.modules.contacts.views.detail (module), 54
- shoop.admin.modules.contacts.views.edit (module), 54
- shoop.admin.modules.contacts.views.list (module), 55
- shoop.admin.modules.contacts.views.reset (module), 56
- shoop.admin.modules.demo (module), 58
- shoop.admin.modules.manufacturers (module), 60
- shoop.admin.modules.manufacturers.views (module), 59
- shoop.admin.modules.manufacturers.views.edit (module), 58
- shoop.admin.modules.manufacturers.views.list (module), 59
- shoop.admin.modules.media (module), 61
- shoop.admin.modules.media.utils (module), 60
- shoop.admin.modules.media.views (module), 60
- shoop.admin.modules.orders (module), 71
- shoop.admin.modules.orders.dashboard (module), 69
- shoop.admin.modules.orders.json_order_creator (module), 70
- shoop.admin.modules.orders.receivers (module), 70
- shoop.admin.modules.orders.sections (module), 71
- shoop.admin.modules.orders.views (module), 66
- shoop.admin.modules.orders.views.detail (module), 61
- shoop.admin.modules.orders.views.edit (module), 62
- shoop.admin.modules.orders.views.list (module), 63
- shoop.admin.modules.orders.views.log (module), 63
- shoop.admin.modules.orders.views.payment (module), 63
- shoop.admin.modules.orders.views.refund (module), 63
- shoop.admin.modules.orders.views.shipment (module), 65
- shoop.admin.modules.permission_groups (module), 73
- shoop.admin.modules.permission_groups.views (module), 73
- shoop.admin.modules.permission_groups.views.edit (module), 72
- shoop.admin.modules.permission_groups.views.list (module), 73
- shoop.admin.modules.product_types (module), 75
- shoop.admin.modules.product_types.views (module), 74
- shoop.admin.modules.product_types.views.edit (module), 74
- shoop.admin.modules.product_types.views.list (module), 74
- shoop.admin.modules.products (module), 91
- shoop.admin.modules.products.forms (module), 79
- shoop.admin.modules.products.forms.base_forms (module), 75
- shoop.admin.modules.products.forms.package_forms (module), 78
- shoop.admin.modules.products.forms.parent_forms (module), 78
- shoop.admin.modules.products.forms.simple_variation_forms (module), 78
- shoop.admin.modules.products.forms.variable_variation_forms (module), 79
- shoop.admin.modules.products.utils (module), 91
- shoop.admin.modules.products.views (module), 89
- shoop.admin.modules.products.views.delete (module), 83
- shoop.admin.modules.products.views.edit (module), 83
- shoop.admin.modules.products.views.edit_cross_sell (module), 84
- shoop.admin.modules.products.views.edit_media (module), 85
- shoop.admin.modules.products.views.edit_package (module), 86
- shoop.admin.modules.products.views.edit_parent (module), 87
- shoop.admin.modules.products.views.edit_variation (module), 88
- shoop.admin.modules.products.views.list (module), 88

- shoop.admin.modules.products.views.toolbars (module), 88
- shoop.admin.modules.sales_units (module), 92
- shoop.admin.modules.sales_units.views (module), 92
- shoop.admin.modules.sales_units.views.edit (module), 91
- shoop.admin.modules.sales_units.views.list (module), 92
- shoop.admin.modules.service_providers (module), 94
- shoop.admin.modules.service_providers.forms (module), 94
- shoop.admin.modules.service_providers.views (module), 93
- shoop.admin.modules.services (module), 101
- shoop.admin.modules.services.base_form_part (module), 96
- shoop.admin.modules.services.behavior_form_part (module), 96
- shoop.admin.modules.services.forms (module), 97
- shoop.admin.modules.services.views (module), 95
- shoop.admin.modules.services.weight_based_pricing (module), 95
- shoop.admin.modules.shops (module), 104
- shoop.admin.modules.shops.views (module), 104
- shoop.admin.modules.shops.views.edit (module), 102
- shoop.admin.modules.shops.views.list (module), 103
- shoop.admin.modules.suppliers (module), 106
- shoop.admin.modules.suppliers.views (module), 106
- shoop.admin.modules.suppliers.views.edit (module), 105
- shoop.admin.modules.suppliers.views.list (module), 105
- shoop.admin.modules.system (module), 107
- shoop.admin.modules.system.views (module), 107
- shoop.admin.modules.system.views.telemetry (module), 107
- shoop.admin.modules.taxes (module), 111
- shoop.admin.modules.taxes.views (module), 109
- shoop.admin.modules.taxes.views.edit (module), 107
- shoop.admin.modules.taxes.views.list (module), 109
- shoop.admin.modules.users (module), 116
- shoop.admin.modules.users.views (module), 114
- shoop.admin.modules.users.views.detail (module), 111
- shoop.admin.modules.users.views.list (module), 112
- shoop.admin.modules.users.views.password (module), 112
- shoop.admin.modules.users.views.permissions (module), 113
- shoop.admin.template_helpers (module), 116
- shoop.admin.template_helpers.shoop_admin (module), 116
- shoop.admin.templatetags (module), 117
- shoop.admin.templatetags.shoop_admin (module), 117
- shoop.admin.toolbar (module), 128
- shoop.admin.urls (module), 130
- shoop.admin.utils (module), 123
- shoop.admin.utils.bs3_renderers (module), 117
- shoop.admin.utils.forms (module), 117
- shoop.admin.utils.permissions (module), 118
- shoop.admin.utils.picotable (module), 118
- shoop.admin.utils.search (module), 120
- shoop.admin.utils.urls (module), 121
- shoop.admin.utils.views (module), 122
- shoop.admin.views (module), 124
- shoop.admin.views.dashboard (module), 123
- shoop.admin.views.menu (module), 123
- shoop.admin.views.search (module), 123
- shoop.admin.views.select (module), 124
- shoop.api (module), 132
- shoop.api.encoders (module), 131
- shoop.api.factories (module), 131
- shoop.api.urls (module), 131
- shoop.apps (module), 134
- shoop.apps.provides (module), 132
- shoop.apps.settings (module), 133
- shoop.campaigns (module), 178
- shoop.campaigns.admin_module (module), 145
- shoop.campaigns.admin_module.form_parts (module), 142
- shoop.campaigns.admin_module.form_sets (module), 143
- shoop.campaigns.admin_module.forms (module), 135
- shoop.campaigns.admin_module.views (module), 140
- shoop.campaigns.apps (module), 177
- shoop.campaigns.models (module), 169
- shoop.campaigns.models.basket_conditions (module), 145
- shoop.campaigns.models.basket_effects (module), 152
- shoop.campaigns.models.basket_line_effects (module), 154
- shoop.campaigns.models.campaigns (module), 157
- shoop.campaigns.models.catalog_filters (module), 161
- shoop.campaigns.models.contact_group_sales_ranges (module), 164
- shoop.campaigns.models.context_conditions (module), 164
- shoop.campaigns.models.product_effects (module), 167
- shoop.campaigns.modules (module), 178
- shoop.campaigns.signal_handlers (module), 178
- shoop.campaigns.templates (module), 177
- shoop.campaigns.utils (module), 178
- shoop.configuration (module), 377
- shoop.core (module), 264
- shoop.core.api (module), 180
- shoop.core.api.orders (module), 178
- shoop.core.api.products (module), 179
- shoop.core.cache (module), 182
- shoop.core.cache.impl (module), 180
- shoop.core.defaults (module), 183
- shoop.core.defaults.order_statuses (module), 183
- shoop.core.excs (module), 260

- shoop.core.fields (module), 184
- shoop.core.fields.tagged_json (module), 183
- shoop.core.management (module), 187
- shoop.core.management.commands (module), 187
- shoop.core.management.commands.compute_bought_with_relations (module), 185
- shoop.core.management.commands.makemessages (module), 185
- shoop.core.management.commands.shoop_init (module), 186
- shoop.core.management.commands.shoop_makemessages (module), 186
- shoop.core.management.commands.shoop_show_settings (module), 186
- shoop.core.middleware (module), 260
- shoop.core.models (module), 187
- shoop.core.modules (module), 238
- shoop.core.modules.interface (module), 237
- shoop.core.order_creator (module), 238
- shoop.core.order_creator.signals (module), 238
- shoop.core.pricing (module), 243
- shoop.core.pricing.default_pricing (module), 242
- shoop.core.settings (module), 261
- shoop.core.shortcuts (module), 250
- shoop.core.signals (module), 263
- shoop.core.stocks (module), 250
- shoop.core.suppliers (module), 251
- shoop.core.suppliers.base (module), 251
- shoop.core.taxing (module), 253
- shoop.core.taxing.utils (module), 252
- shoop.core.telemetry (module), 263
- shoop.core.templatetags (module), 256
- shoop.core.templatetags.prices (module), 255
- shoop.core.templatetags.shoop_common (module), 256
- shoop.core.utils (module), 260
- shoop.core.utils.form_mixins (module), 257
- shoop.core.utils.maintenance (module), 257
- shoop.core.utils.model_caching_descriptor (module), 257
- shoop.core.utils.name_mixin (module), 257
- shoop.core.utils.price_display (module), 258
- shoop.core.utils.prices (module), 258
- shoop.core.utils.product_bought_with_relations (module), 258
- shoop.core.utils.product_caching_object (module), 259
- shoop.core.utils.query (module), 259
- shoop.core.utils.slugs (module), 259
- shoop.core.utils.tax_numbers (module), 259
- shoop.core.utils.users (module), 259
- shoop.core.utils.vat (module), 259
- shoop.customer_group_pricing (module), 265
- shoop.customer_group_pricing.admin_form_part (module), 264
- shoop.customer_group_pricing.api (module), 264
- shoop.customer_group_pricing.models (module), 265
- shoop.customer_group_pricing.module (module), 265
- shoop.default_tax (module), 268
- shoop.default_tax.admin_module (module), 266
- shoop.default_tax.admin_module.views (module), 266
- shoop.default_tax.models (module), 267
- shoop.default_tax.module (module), 267
- shoop.front (module), 296
- shoop.front.admin_module (module), 269
- shoop.front.admin_module.carts (module), 269
- shoop.front.admin_module.carts.views (module), 268
- shoop.front.apps (module), 277
- shoop.front.apps.auth (module), 271
- shoop.front.apps.auth.forms (module), 269
- shoop.front.apps.auth.urls (module), 270
- shoop.front.apps.auth.views (module), 270
- shoop.front.apps.customer_information (module), 273
- shoop.front.apps.customer_information.forms (module), 271
- shoop.front.apps.customer_information.urls (module), 272
- shoop.front.apps.customer_information.views (module), 272
- shoop.front.apps.personal_order_history (module), 273
- shoop.front.apps.personal_order_history.urls (module), 273
- shoop.front.apps.personal_order_history.views (module), 273
- shoop.front.apps.registration (module), 274
- shoop.front.apps.registration.settings (module), 274
- shoop.front.apps.registration.urls (module), 274
- shoop.front.apps.registration.views (module), 274
- shoop.front.apps.simple_order_notification (module), 275
- shoop.front.apps.simple_order_notification.admin_module (module), 275
- shoop.front.apps.simple_order_notification.templates (module), 275
- shoop.front.apps.simple_search (module), 277
- shoop.front.apps.simple_search.template_helpers (module), 276
- shoop.front.apps.simple_search.urls (module), 276
- shoop.front.apps.simple_search.views (module), 276
- shoop.front.basket (module), 281
- shoop.front.basket.command_dispatcher (module), 277
- shoop.front.basket.commands (module), 278
- shoop.front.basket.objects (module), 279
- shoop.front.basket.order_creator (module), 280
- shoop.front.basket.storage (module), 280
- shoop.front.basket.update_methods (module), 281
- shoop.front.checkout (module), 285
- shoop.front.checkout.addresses (module), 282
- shoop.front.checkout.confirm (module), 283
- shoop.front.checkout.empty (module), 283

shoop.front.checkout.methods (module), 284
 shoop.front.checkout.single_page (module), 285
 shoop.front.error_handling (module), 293
 shoop.front.middleware (module), 293
 shoop.front.models (module), 287
 shoop.front.models.stored_basket (module), 287
 shoop.front.notify_events (module), 294
 shoop.front.settings (module), 295
 shoop.front.signals (module), 295
 shoop.front.template_helpers (module), 290
 shoop.front.template_helpers.category (module), 288
 shoop.front.template_helpers.general (module), 288
 shoop.front.template_helpers.product (module), 289
 shoop.front.template_helpers.urls (module), 289
 shoop.front.templatetags (module), 290
 shoop.front.templatetags.shoop_front (module), 290
 shoop.front.templatetags.thumbnails (module), 290
 shoop.front.urls (module), 296
 shoop.front.utils (module), 291
 shoop.front.utils.product_sorting (module), 290
 shoop.front.utils.product_statistics (module), 290
 shoop.front.utils.views (module), 291
 shoop.front.views (module), 293
 shoop.front.views.basket (module), 291
 shoop.front.views.category (module), 291
 shoop.front.views.checkout (module), 292
 shoop.front.views.index (module), 292
 shoop.front.views.order (module), 292
 shoop.front.views.payment (module), 293
 shoop.front.views.product (module), 293
 shoop.guide (module), 297
 shoop.guide.admin_module (module), 296
 shoop.guide.apps (module), 296
 shoop.guide.settings (module), 296
 shoop.notify (module), 313
 shoop.notify.actions (module), 298
 shoop.notify.actions.debug (module), 297
 shoop.notify.actions.email (module), 297
 shoop.notify.actions.notification (module), 297
 shoop.notify.actions.order (module), 298
 shoop.notify.admin_module (module), 301
 shoop.notify.admin_module.forms (module), 301
 shoop.notify.admin_module.utils (module), 301
 shoop.notify.admin_module.views (module), 300
 shoop.notify.admin_module.views.edit (module), 299
 shoop.notify.admin_module.views.editor (module), 299
 shoop.notify.admin_module.views.list (module), 300
 shoop.notify.base (module), 306
 shoop.notify.conditions (module), 303
 shoop.notify.conditions.simple (module), 302
 shoop.notify.enums (module), 309
 shoop.notify.models (module), 305
 shoop.notify.models.notification (module), 304
 shoop.notify.models.script (module), 305
 shoop.notify.runner (module), 310
 shoop.notify.script (module), 310
 shoop.notify.settings (module), 311
 shoop.notify.template (module), 311
 shoop.notify.typology (module), 311
 shoop.order_printouts (module), 316
 shoop.order_printouts.admin_module (module), 316
 shoop.order_printouts.admin_module.toolbar (module), 316
 shoop.order_printouts.admin_module.views (module), 316
 shoop.order_printouts.pdf_export (module), 316
 shoop.simple_cms (module), 321
 shoop.simple_cms.admin_module (module), 318
 shoop.simple_cms.admin_module.views (module), 317
 shoop.simple_cms.models (module), 318
 shoop.simple_cms.plugins (module), 319
 shoop.simple_cms.template_helpers (module), 320
 shoop.simple_cms.urls (module), 320
 shoop.simple_cms.views (module), 320
 shoop.simple_supplier (module), 325
 shoop.simple_supplier.admin_module (module), 322
 shoop.simple_supplier.admin_module.forms (module), 321
 shoop.simple_supplier.admin_module.views (module), 322
 shoop.simple_supplier.forms (module), 322
 shoop.simple_supplier.models (module), 323
 shoop.simple_supplier.module (module), 324
 shoop.simple_supplier.utils (module), 324
 shoop.testing (module), 336
 shoop.testing.admin_module (module), 326
 shoop.testing.admin_module.mock_view (module), 325
 shoop.testing.factories (module), 329
 shoop.testing.image_generator (module), 332
 shoop.testing.management (module), 326
 shoop.testing.management.commands (module), 326
 shoop.testing.management.commands.shoop_populate_mock (module), 326
 shoop.testing.mock_population (module), 333
 shoop.testing.models (module), 327
 shoop.testing.service_forms (module), 333
 shoop.testing.simple_checkout_phase (module), 334
 shoop.testing.soup_utils (module), 336
 shoop.testing.text_data (module), 336
 shoop.testing.utils (module), 336
 shoop.themes (module), 338
 shoop.themes.classic_gray (module), 337
 shoop.themes.classic_gray.plugins (module), 337
 shoop.themes.classic_gray.views (module), 336
 shoop.utils (module), 350
 shoop.utils.analog (module), 338
 shoop.utils.dates (module), 338

shoop.utils.decorators (module), 339
shoop.utils.excs (module), 339
shoop.utils.fields (module), 340
shoop.utils.filer (module), 340
shoop.utils.form_group (module), 341
shoop.utils.forms (module), 341
shoop.utils.http (module), 342
shoop.utils.i18n (module), 342
shoop.utils.importing (module), 343
shoop.utils.iterables (module), 343
shoop.utils.models (module), 343
shoop.utils.money (module), 344
shoop.utils.multilanguage_model_form (module), 344
shoop.utils.numbers (module), 344
shoop.utils.objects (module), 345
shoop.utils.patterns (module), 346
shoop.utils.properties (module), 347
shoop.utils.serialization (module), 348
shoop.utils.settings_doc (module), 349
shoop.utils.setup (module), 349
shoop.utils.text (module), 349
shoop.utils.translation (module), 350
shoop.xtheme (module), 373
shoop.xtheme.admin_module (module), 352
shoop.xtheme.admin_module.views (module), 351
shoop.xtheme.editing (module), 359
shoop.xtheme.engine (module), 360
shoop.xtheme.forms (module), 361
shoop.xtheme.layout (module), 361
shoop.xtheme.models (module), 365
shoop.xtheme.parsing (module), 366
shoop.xtheme.plugins (module), 357
shoop.xtheme.plugins.category_links (module), 353
shoop.xtheme.plugins.consts (module), 353
shoop.xtheme.plugins.forms (module), 353
shoop.xtheme.plugins.image (module), 354
shoop.xtheme.plugins.snippets (module), 355
shoop.xtheme.plugins.social_media_links (module), 355
shoop.xtheme.plugins.text (module), 356
shoop.xtheme.plugins.widgets (module), 356
shoop.xtheme.rendering (module), 369
shoop.xtheme.resources (module), 369
shoop.xtheme.template_ns (module), 371
shoop.xtheme.templatetags (module), 357
shoop.xtheme.templatetags.xtheme_tags (module), 357
shoop.xtheme.testing (module), 371
shoop.xtheme.urls (module), 372
shoop.xtheme.utils (module), 372
shoop.xtheme.view_config (module), 372
shoop.xtheme.views (module), 359
shoop.xtheme.views.command (module), 357
shoop.xtheme.views.editor (module), 358
shoop.xtheme.views.extra (module), 358
shoop.xtheme.views.forms (module), 358
SHOOP_ADDRESS_HOME_COUNTRY (in module shoop.core.settings), 261
SHOOP_ALLOW_ANONYMOUS_ORDERS (in module shoop.core.settings), 261
SHOOP_BASKET_CLASS_SPEC (in module shoop.front.settings), 295
SHOOP_BASKET_COMMAND_DISPATCHER_SPEC (in module shoop.front.settings), 295
SHOOP_BASKET_ORDER_CREATOR_SPEC (in module shoop.front.settings), 295
SHOOP_BASKET_STORAGE_CLASS_SPEC (in module shoop.front.settings), 295
SHOOP_BASKET_UPDATE_METHODS_SPEC (in module shoop.front.settings), 295
SHOOP_BASKET_VIEW_SPEC (in module shoop.front.settings), 295
SHOOP_CACHE_DURATIONS (in module shoop.core.settings), 262
SHOOP_CALCULATE_TAXES_AUTOMATICALLY_IF_POSSIBLE (in module shoop.core.settings), 262
SHOOP_CHECKOUT_VIEW_SPEC (in module shoop.front.settings), 295
SHOOP_DEFAULT_CACHE_DURATION (in module shoop.core.settings), 262
SHOOP_DEFAULT_ORDER_LABEL (in module shoop.core.settings), 262
SHOOP_DISCOUNT_MODULES (in module shoop.core.settings), 261
SHOOP_ENABLE_ATTRIBUTES (in module shoop.core.settings), 261
SHOOP_ENABLE_MULTIPLE_SHOPS (in module shoop.core.settings), 261
SHOOP_ENABLE_MULTIPLE_SUPPLIERS (in module shoop.core.settings), 262
SHOOP_FRONT_ADDRESS_FIELD_PROPERTIES (in module shoop.front.settings), 295
SHOOP_FRONT_INSTALL_ERROR_HANDLERS (in module shoop.front.settings), 295
SHOOP_GUIDE_API_URL (in module shoop.guide.settings), 296
SHOOP_GUIDE_FETCH_RESULTS (in module shoop.guide.settings), 296
SHOOP_GUIDE_LINK_URL (in module shoop.guide.settings), 296
SHOOP_GUIDE_TIMEOUT_LIMIT (in module shoop.guide.settings), 296
SHOOP_HOME_CURRENCY (in module shoop.core.settings), 261
SHOOP_ORDER_IDENTIFIER_METHOD (in module shoop.core.settings), 261
SHOOP_ORDER_KNOWN_EXTRA_DATA_KEYS (in module shoop.core.settings), 262
SHOOP_ORDER_KNOWN_PAYMENT_DATA_KEYS (in module shoop.core.settings), 262

- SHOOP_ORDER_KNOWN_SHIPPING_DATA_KEYS (in module shoop.core.settings), 262
- SHOOP_ORDER_LABELS (in module shoop.core.settings), 262
- SHOOP_ORDER_SOURCE_MODIFIER_MODULES (in module shoop.core.settings), 261
- SHOOP_PRICING_MODULE (in module shoop.core.settings), 261
- SHOOP_REFERENCE_NUMBER_LENGTH (in module shoop.core.settings), 261
- SHOOP_REFERENCE_NUMBER_METHOD (in module shoop.core.settings), 261
- SHOOP_REFERENCE_NUMBER_PREFIX (in module shoop.core.settings), 261
- SHOOP_REGISTRATION_REQUIRES_ACTIVATION (in module shoop.front.apps.registration.settings), 274
- SHOOP_TAX_MODULE (in module shoop.core.settings), 261
- SHOOP_TELEMETRY_ENABLED (in module shoop.core.settings), 262
- SHOOP_TELEMETRY_URL (in module shoop.core.settings), 262
- ShoopAdminAppConfig (class in shoop.admin), 130
- ShoopAdminForm (class in shoop.admin.forms), 46
- ShoopApiAppConfig (class in shoop.api), 132
- ShoopCoreAppConfig (class in shoop.core), 264
- ShoopFrontAppConfig (class in shoop.front), 296
- ShoopFrontMiddleware (class in shoop.front.middleware), 294
- ShoopModel (class in shoop.core.models), 229
- ShoopModel.Meta (class in shoop.core.models), 229
- ShoopSimpleSupplierAppConfig (class in shoop.simple_supplier), 325
- ShoopTestingAppConfig (class in shoop.testing), 336
- ShoopUtilsAppConfig (class in shoop.utils), 351
- shop, 18
- Shop (class in shoop.core.models), 229
- shop (shoop.campaigns.models.BasketCampaign attribute), 171
- shop (shoop.campaigns.models.Campaign attribute), 173
- shop (shoop.campaigns.models.campaigns.BasketCampaign attribute), 159
- shop (shoop.campaigns.models.campaigns.Campaign attribute), 157
- shop (shoop.campaigns.models.campaigns.CatalogCampaign attribute), 158
- shop (shoop.campaigns.models.CatalogCampaign attribute), 175
- shop (shoop.campaigns.models.contact_group_sales_ranges attribute), 164
- shop (shoop.campaigns.models.ContactGroupSalesRange attribute), 176
- shop (shoop.core.models.ConfigurationItem attribute), 192
- shop (shoop.core.models.Order attribute), 202
- shop (shoop.core.models.PaymentMethod attribute), 207
- shop (shoop.core.models.Service attribute), 225
- shop (shoop.core.models.ShippingMethod attribute), 229
- shop (shoop.core.models.ShopProduct attribute), 232
- shop (shoop.customer_group_pricing.models.CgpPrice attribute), 265
- shop (shoop.front.models.stored_basket.StoredBasket attribute), 287
- shop (shoop.front.models.StoredBasket attribute), 288
- Shop.DoesNotExist, 229
- Shop.MultipleObjectsReturned, 230
- shop_primary_image (shoop.core.models.ShopProduct attribute), 232
- shop_product (shoop.front.basket.objects.BasketLine attribute), 279
- shop_product_m2m (shoop.core.models.PaymentMethod attribute), 207
- shop_product_m2m (shoop.core.models.ShippingMethod attribute), 229
- shop_products (shoop.core.models.Category attribute), 190
- shop_products (shoop.core.models.Product attribute), 213, 217
- shop_products (shoop.core.models.Shop attribute), 230
- shop_products (shoop.core.models.Supplier attribute), 234
- shop_set (shoop.core.models.Contact attribute), 193
- shop_set (shoop.core.models.MutableAddress attribute), 199
- ShopBaseForm (class in shoop.admin.modules.shops.views.edit), 102
- ShopBaseForm.Meta (class in shoop.admin.modules.shops.views.edit), 102
- ShopBaseFormPart (class in shoop.admin.modules.shops.views.edit), 102
- ShopEditView (class in shoop.admin.modules.shops.views), 104
- ShopEditView (class in shoop.admin.modules.shops.views.edit), 103
- ShopFactory (class in shoop.testing.factories), 329
- ShopListView (class in shoop.admin.modules.shops.views), 104
- ShopListView (class in shoop.admin.modules.shops.views.list), 103
- ShopMismatchBasketCompatibilityError, 280
- ShopModule (class in shoop.admin.modules.shops), 104
- ShopProduct (class in shoop.core.models), 230
- ShopProduct.DoesNotExist, 231

- ShopProduct.MultipleObjectsReturned, [231](#)
- ShopProductFactory (class in shoop.testing.factories), [330](#)
- ShopProductForm (class in shoop.admin.modules.products.forms), [81](#)
- ShopProductForm (class in shoop.admin.modules.products.forms.base_forms), [76](#)
- ShopProductForm.Meta (class in shoop.admin.modules.products.forms), [81](#)
- ShopProductForm.Meta (class in shoop.admin.modules.products.forms.base_forms), [76](#)
- ShopProductFormPart (class in shoop.admin.modules.products.views.edit), [83](#)
- ShopProductSerializer (class in shoop.core.api.products), [179](#)
- ShopProductSerializer.Meta (class in shoop.core.api.products), [179](#)
- ShopProductViewSet (class in shoop.core.api.products), [180](#)
- shops (shoop.core.models.Category attribute), [190](#)
- shops (shoop.core.models.ProductMedia attribute), [219](#)
- ShopStatus (class in shoop.core.models), [232](#)
- short_name (shoop.core.models.SalesUnit attribute), [223](#)
- short_name (shoop.testing.factories.SalesUnitFactory attribute), [330](#)
- should_calculate_taxes_automatically() (in module shoop.core.taxing), [255](#)
- should_skip() (shoop.front.checkout.CheckoutPhaseViewMixin method), [286](#)
- SHOW_ON_PRODUCT_PAGE (shoop.core.models.AttributeVisibility attribute), [189](#)
- show_prices() (in module shoop.core.templatetags.prices), [255](#)
- SIMPLE_VARIATION_PARENT (shoop.core.models.ProductMode attribute), [220](#)
- SimpleCMSAdminModule (class in shoop.simple_cms.admin_module), [318](#)
- SimpleCMSTemplateHelpers (class in shoop.simple_cms.template_helpers), [320](#)
- SimpleOrderNotificationAppConfig (class in shoop.front.apps.simple_order_notification), [275](#)
- SimpleOrderNotificationModule (class in shoop.front.apps.simple_order_notification.admin_module), [275](#)
- SimplePrintoutsToolBarButton (class in shoop.order_printouts.admin_module.toolbar), [316](#)
- SimpleSearchAppConfig (class in shoop.front.apps.simple_search), [277](#)
- SimpleSupplierForm (class in shoop.simple_supplier.admin_module.forms), [321](#)
- SimpleSupplierFormPart (class in shoop.simple_supplier.admin_module.forms), [321](#)
- SimpleSupplierModule (class in shoop.simple_supplier.module), [324](#)
- SimpleVariationChildForm (class in shoop.admin.modules.products.forms), [81](#)
- SimpleVariationChildForm (class in shoop.admin.modules.products.forms.simple_variation_forms), [78](#)
- SimpleVariationChildFormSet (class in shoop.admin.modules.products.forms), [82](#)
- SimpleVariationChildFormSet (class in shoop.admin.modules.products.forms.simple_variation_forms), [79](#)
- simplify_variation() (shoop.core.models.Product method), [213](#), [217](#)
- SingleCheckoutPhase (class in shoop.front.checkout.single_page), [285](#)
- SIZES (shoop.admin.dashboard.blocks.DashboardBlock attribute), [42](#)
- SIZES (shoop.admin.dashboard.DashboardBlock attribute), [44](#)
- sku (shoop.testing.factories.ProductFactory attribute), [330](#)
- slug (shoop.core.models.Category attribute), [190](#)
- slug (shoop.core.models.Product attribute), [213](#), [217](#)
- slug (shoop.testing.factories.ShopFactory attribute), [329](#)
- slug_field (shoop.simple_cms.views.PageView attribute), [320](#)
- slug_url_kwarg (shoop.simple_cms.views.PageView attribute), [320](#)
- snake_case() (in module shoop.utils.text), [350](#)
- SnippetsPlugin (class in shoop.xtheme.plugins.snippets), [355](#)
- SocialMediaLinksPlugin (class in shoop.xtheme.plugins.social_media_links), [355](#)
- SocialMediaLinksPluginForm (class in shoop.xtheme.plugins.social_media_links), [355](#)
- soft_delete() (shoop.core.models.Product method), [213](#), [217](#)
- sort_products() (in module shoop.front.utils.product_sorting), [290](#)
- sort_queryset() (shoop.admin.utils.picotable.Column method), [119](#)
- SourceLine (class in shoop.core.order_creator), [241](#)
- SourceLineTax (class in shoop.core.taxing), [253](#)
- space_case() (in module shoop.utils.text), [350](#)

- SPECIFIC_USER (shoop.notify.enums.RecipientType attribute), 309
- split_name (shoop.core.utils.name_mixin.NameMixin attribute), 257
- split_query() (in module shoop.admin.utils.search), 120
- stacked_value_added_taxes() (in module shoop.core.taxing.utils), 252
- StaffOnlyBehaviorComponent (class in shoop.core.models), 232
- staffonlybehaviorcomponent (shoop.core.models.ServiceBehaviorComponent attribute), 226
- StaffOnlyBehaviorComponent.DoesNotExist, 232
- StaffOnlyBehaviorComponent.MultipleObjectsReturned, 232
- StaffOnlyBehaviorComponentForm (class in shoop.admin.modules.services.forms), 100
- StaffOnlyBehaviorComponentForm.Meta (class in shoop.admin.modules.services.forms), 100
- status (shoop.core.models.Category attribute), 190
- status (shoop.core.models.Order attribute), 202
- status (shoop.core.models.ProductVariationResult attribute), 221
- status (shoop.core.models.SavedAddress attribute), 223
- status (shoop.core.models.Shipment attribute), 228
- status (shoop.core.models.Shop attribute), 230
- status (shoop.core.order_creator.OrderSource attribute), 240
- status (shoop.testing.factories.CategoryFactory attribute), 330
- status (shoop.xtheme.models.SavedViewConfig attribute), 365
- status_text (shoop.core.models.Product attribute), 213, 217
- Step (class in shoop.notify.script), 310
- step() (shoop.testing.image_generator.RandomImageGenerator method), 332
- StepConditionOperator (class in shoop.notify.enums), 309
- StepNext (class in shoop.notify.enums), 309
- stock_behavior (shoop.core.models.Product attribute), 214, 217
- stock_unit_price (shoop.simple_supplier.models.StockCount attribute), 323
- stock_unit_price_value (shoop.simple_supplier.models.StockCount attribute), 323
- stock_value (shoop.simple_supplier.models.StockCount attribute), 323
- StockAdjustment (class in shoop.simple_supplier.models), 323
- StockAdjustment.DoesNotExist, 323
- StockAdjustment.MultipleObjectsReturned, 323
- stockadjustment_set (shoop.core.models.Supplier attribute), 234
- StockAdjustmentForm (class in shoop.simple_supplier.forms), 322
- StockBehavior (class in shoop.core.models), 233
- StockCount (class in shoop.simple_supplier.models), 323
- StockCount.DoesNotExist, 323
- StockCount.MultipleObjectsReturned, 323
- stockcount_set (shoop.core.models.Supplier attribute), 234
- STOCKED (shoop.core.models.StockBehavior attribute), 233
- StocksAdminModule (class in shoop.simple_supplier.admin_module), 322
- StocksListView (class in shoop.simple_supplier.admin_module.views), 322
- STOP (shoop.notify.enums.StepNext attribute), 309
- storage (shoop.front.checkout.CheckoutPhaseViewMixin attribute), 286
- storage_identifier (shoop.testing.simple_checkout_phase.TestMethodCheck attribute), 335
- storage_identifier (shoop.testing.simple_checkout_phase.TestPaymentCheck attribute), 335
- storage_identifier (shoop.testing.simple_checkout_phase.TestShipmentCheck attribute), 335
- StoredBasket (class in shoop.front.models), 287
- StoredBasket (class in shoop.front.models.stored_basket), 287
- StoredBasket.DoesNotExist, 287
- StoredBasket.MultipleObjectsReturned, 287
- storedbasket_set (shoop.core.models.Product attribute), 214, 217
- storedbasket_set (shoop.core.models.Shop attribute), 230
- strip_non_float_chars() (in module shoop.utils.numbers), 344
- subject_template_name (shoop.front.apps.auth.forms.RecoverPasswordForm attribute), 269
- success_url (shoop.admin.modules.contact_groups.views.ContactGroupDelete attribute), 53
- success_url (shoop.admin.modules.contact_groups.views.delete.ContactGroupDelete attribute), 50
- success_url (shoop.admin.modules.service_providers.views.ServiceProviderDelete attribute), 93
- success_url (shoop.admin.modules.services.views.PaymentMethodDeleteView attribute), 95
- success_url (shoop.admin.modules.services.views.ShippingMethodDeleteView attribute), 95
- success_url (shoop.front.apps.auth.views.RecoverPasswordConfirmView attribute), 271
- success_url (shoop.front.apps.auth.views.RecoverPasswordView attribute), 270
- suffix (shoop.core.api.orders.OrderViewSet attribute), 179
- suffix (shoop.core.api.products.ProductViewSet attribute), 180

- suffix (shoop.core.api.products.ShopProductViewSet attribute), 180
- SuppliedProduct (class in shoop.core.models), 233
- SuppliedProduct.DoesNotExist, 233
- SuppliedProduct.MultipleObjectsReturned, 233
- suppliedproduct_set (shoop.core.models.Product attribute), 214, 217
- suppliedproduct_set (shoop.core.models.Supplier attribute), 234
- Supplier (class in shoop.core.models), 233
- supplier (shoop.core.models.OrderLine attribute), 203
- supplier (shoop.core.models.Shipment attribute), 228
- supplier (shoop.core.models.SuppliedProduct attribute), 233
- supplier (shoop.simple_supplier.models.StockAdjustment attribute), 323
- supplier (shoop.simple_supplier.models.StockCount attribute), 323
- Supplier.DoesNotExist, 233
- Supplier.MultipleObjectsReturned, 233
- SupplierEditView (class in shoop.admin.modules.suppliers.views), 106
- SupplierEditView (class in shoop.admin.modules.suppliers.views.edit), 105
- SupplierForm (class in shoop.admin.modules.suppliers.views.edit), 105
- SupplierForm.Meta (class in shoop.admin.modules.suppliers.views.edit), 105
- SupplierListView (class in shoop.admin.modules.suppliers.views), 106
- SupplierListView (class in shoop.admin.modules.suppliers.views.list), 105
- SupplierModule (class in shoop.admin.modules.suppliers), 106
- suppliers (shoop.core.models.ShopProduct attribute), 232
- SupplierType (class in shoop.core.models), 234
- SystemModule (class in shoop.admin.modules.system), 107
- T**
- tag_registry (in module shoop.core.fields.tagged_json), 183
- TaggedJSONEncoder (class in shoop.core.fields.tagged_json), 183
- TaggedJSONField (class in shoop.core.fields), 185
- TagRegistry (class in shoop.core.fields.tagged_json), 183
- tags (shoop.xtheme.parsing.LayoutPartExtension attribute), 368
- tags (shoop.xtheme.parsing.PlaceholderExtension attribute), 367
- tags (shoop.xtheme.parsing.PluginExtension attribute), 368
- target (shoop.core.models.OrderLogEntry attribute), 205
- target (shoop.utils.analog.BaseLogEntry attribute), 338
- Tax (class in shoop.core.models), 234
- tax (shoop.core.models.OrderLineTax attribute), 204
- tax (shoop.core.taxing.LineTax attribute), 253
- tax (shoop.default_tax.models.TaxRule attribute), 267
- Tax.DoesNotExist, 234
- Tax.MultipleObjectsReturned, 234
- tax_amount (shoop.core.models.OrderLine attribute), 203
- tax_amount (shoop.core.order_creator.SourceLine attribute), 242
- tax_amount (shoop.core.taxing.TaxedPrice attribute), 255
- tax_class (shoop.core.models.PaymentMethod attribute), 207
- tax_class (shoop.core.models.Product attribute), 214, 218
- tax_class (shoop.core.models.Service attribute), 225
- tax_class (shoop.core.models.ShippingMethod attribute), 229
- tax_class (shoop.core.order_creator.SourceLine attribute), 242
- tax_class (shoop.core.taxing.TaxableItem attribute), 254
- tax_class (shoop.testing.factories.ProductFactory attribute), 330
- tax_classes (shoop.default_tax.models.TaxRule attribute), 267
- tax_group (shoop.core.models.Contact attribute), 193
- tax_percentage (shoop.core.pricing.Priceful attribute), 247
- tax_rate (shoop.core.pricing.Priceful attribute), 247
- tax_rate (shoop.core.taxing.TaxedPrice attribute), 255
- TaxableItem (class in shoop.core.taxing), 254
- TaxClass (class in shoop.core.models), 235
- TaxClass.DoesNotExist, 235
- TaxClass.MultipleObjectsReturned, 235
- TaxClassEditView (class in shoop.admin.modules.taxes.views), 110
- TaxClassEditView (class in shoop.admin.modules.taxes.views.edit), 109
- TaxClassForm (class in shoop.admin.modules.taxes.views.edit), 108
- TaxClassForm.Meta (class in shoop.admin.modules.taxes.views.edit), 108
- TaxClassListView (class in shoop.admin.modules.taxes.views), 109
- TaxClassListView (class in shoop.admin.modules.taxes.views.list), 109
- TaxEditView (class in shoop.admin.modules.taxes.views), 110
- TaxEditView (class in shoop.admin.modules.taxes.views.edit), 108
- TaxedPrice (class in shoop.core.taxing), 254
- TaxedPriceInfo (class in shoop.core.utils.prices), 258

- taxes (shoop.core.models.OrderLine attribute), 203
- taxes (shoop.core.order_creator.SourceLine attribute), 242
- taxes (shoop.core.taxing.TaxedPrice attribute), 255
- TaxesNotCalculated, 242
- TaxForm (class in shoop.admin.modules.taxes.views.edit), 107
- TaxForm.Meta (class in shoop.admin.modules.taxes.views.edit), 107
- taxful (shoop.core.taxing.TaxedPrice attribute), 254
- taxful_base_price (shoop.core.pricing.Priceful attribute), 247
- taxful_base_unit_price (shoop.core.pricing.Priceful attribute), 247
- taxful_discount_amount (shoop.core.pricing.Priceful attribute), 247
- taxful_discounted_unit_price (shoop.core.pricing.Priceful attribute), 247
- taxful_price (shoop.core.pricing.Priceful attribute), 247
- taxful_total_discount (shoop.core.order_creator.OrderSource attribute), 240
- taxful_total_discount_or_none (shoop.core.order_creator.OrderSource attribute), 240
- taxful_total_price (shoop.core.models.Order attribute), 202
- taxful_total_price (shoop.core.order_creator.OrderSource attribute), 240
- taxful_total_price (shoop.front.models.stored_basket.StoredBasket attribute), 287
- taxful_total_price (shoop.front.models.StoredBasket attribute), 288
- taxful_total_price_or_none (shoop.core.order_creator.OrderSource attribute), 240
- taxful_unit_discount_amount (shoop.core.pricing.Priceful attribute), 247
- TaxfulPrice (class in shoop.core.pricing), 250
- TaxfulPriceProperty (class in shoop.utils.properties), 348
- taxing_context_class (shoop.core.taxing.TaxModule attribute), 254
- TaxingContext (class in shoop.core.taxing), 255
- taxless (shoop.core.taxing.TaxedPrice attribute), 254
- taxless_base_price (shoop.core.pricing.Priceful attribute), 247
- taxless_base_unit_price (shoop.core.pricing.Priceful attribute), 247
- taxless_discount_amount (shoop.core.pricing.Priceful attribute), 247
- taxless_discounted_unit_price (shoop.core.pricing.Priceful attribute), 247
- taxless_price (shoop.core.pricing.Priceful attribute), 247
- taxless_total_discount (shoop.core.order_creator.OrderSource attribute), 241
- taxless_total_discount_or_none (shoop.core.order_creator.OrderSource attribute), 241
- taxless_total_price (shoop.core.models.Order attribute), 202
- taxless_total_price (shoop.core.order_creator.OrderSource attribute), 241
- taxless_total_price (shoop.front.models.stored_basket.StoredBasket attribute), 287
- taxless_total_price (shoop.front.models.StoredBasket attribute), 288
- taxless_total_price_or_none (shoop.core.order_creator.OrderSource attribute), 241
- taxless_unit_discount_amount (shoop.core.pricing.Priceful attribute), 248
- TaxlessPrice (class in shoop.core.pricing), 250
- TaxlessPriceProperty (class in shoop.utils.properties), 348
- TaxListView (class in shoop.admin.modules.taxes.views), 110
- TaxListView (class in shoop.admin.modules.taxes.views.list), 109
- TaxModule (class in shoop.admin.modules.taxes), 111
- TaxModule (class in shoop.core.taxing), 253
- TaxRule (class in shoop.default_tax.models), 267
- TaxRule.DoesNotExist, 267
- TaxRule.MultipleObjectsReturned, 267
- taxrule_set (shoop.core.models.CustomerTaxGroup attribute), 195
- taxrule_set (shoop.core.models.Tax attribute), 235
- taxrule_set (shoop.core.models.TaxClass attribute), 235
- TaxRuleEditView (class in shoop.default_tax.admin_module.views), 266
- TaxRuleForm (class in shoop.default_tax.admin_module.views), 266
- TaxRuleForm.Meta (class in shoop.default_tax.admin_module.views), 266
- TaxRuleListView (class in shoop.default_tax.admin_module.views), 266
- TaxRuleQuerySet (class in shoop.default_tax.models), 267
- TaxRulesAdminModule (class in shoop.default_tax.admin_module), 266
- TaxSummary (class in shoop.core.taxing), 254
- TelemetryNotSent, 263
- TelemetryView (class in shoop.admin.modules.system.views.telemetry), 107
- Template (class in shoop.notify.template), 311

template (shoop.admin.base.OrderSection attribute), 126
template (shoop.admin.modules.orders.sections.ContentsOrderSection attribute), 59
 attribute), 71
template (shoop.admin.modules.orders.sections.LogEntriesOrderSection attribute), 59
 attribute), 71
template (shoop.admin.modules.orders.sections.PaymentOrderSection attribute), 60
 attribute), 71
template_class (shoop.xtheme.engine.XthemeEnvironment attribute), 360
template_dir (shoop.themes.classic_gray.ClassicGrayTheme attribute), 337
template_dir (shoop.xtheme.Theme attribute), 376
template_fields (shoop.notify.Action attribute), 314
template_fields (shoop.notify.actions.email.SendEmail attribute), 297
template_fields (shoop.notify.actions.SendEmail attribute), 298
template_fields (shoop.notify.base.Action attribute), 308
template_name (shoop.addons.admin_module.views.AddonListView attribute), 40
template_name (shoop.addons.admin_module.views.AddonUploadConfirmView attribute), 40
template_name (shoop.addons.admin_module.views.AddonUploadView attribute), 40
template_name (shoop.addons.admin_module.views.list.AddonListView attribute), 38
template_name (shoop.addons.admin_module.views.reload.RefreshView attribute), 38
template_name (shoop.addons.admin_module.views.ReloadView attribute), 40
template_name (shoop.addons.admin_module.views.upload.AddonUploadView attribute), 39
template_name (shoop.addons.admin_module.views.upload.AddonUploadView attribute), 39
template_name (shoop.admin.modules.attributes.views.AttributeEditView attribute), 48
template_name (shoop.admin.modules.attributes.views.edit.AttributeEditView attribute), 47
template_name (shoop.admin.modules.categories.views.CategoryEditView attribute), 49
template_name (shoop.admin.modules.categories.views.edit.CategoryEditView attribute), 49
template_name (shoop.admin.modules.contact_groups.views.ContactGroupEditView attribute), 53
template_name (shoop.admin.modules.contact_groups.views.edit.ContactGroupEditView attribute), 51
template_name (shoop.admin.modules.contacts.views.ContactEditView attribute), 56
template_name (shoop.admin.modules.contacts.views.ContactEditView attribute), 57
template_name (shoop.admin.modules.contacts.views.detail.ContactDetailView attribute), 54
template_name (shoop.admin.modules.contacts.views.edit.ContactEditView attribute), 55
template_name (shoop.admin.modules.manufacturers.views.edit.ManufacturerEditView attribute), 61
template_name (shoop.admin.modules.manufacturers.views.ManufacturerEditView attribute), 62
template_name (shoop.admin.modules.media.views.MediaBrowserView attribute), 61
template_name (shoop.admin.modules.orders.views.detail.OrderDetailView attribute), 61
template_name (shoop.admin.modules.orders.views.edit.OrderEditView attribute), 62
template_name (shoop.admin.modules.orders.views.OrderCreateFullRefundView attribute), 68
template_name (shoop.admin.modules.orders.views.OrderCreatePaymentView attribute), 68
template_name (shoop.admin.modules.orders.views.OrderCreateRefundView attribute), 68
template_name (shoop.admin.modules.orders.views.OrderCreateShipmentView attribute), 69
template_name (shoop.admin.modules.orders.views.OrderDetailView attribute), 67
template_name (shoop.admin.modules.orders.views.OrderEditView attribute), 67
template_name (shoop.admin.modules.orders.views.payment.OrderCreatePaymentView attribute), 63
template_name (shoop.admin.modules.orders.views.refund.OrderCreateFullRefundView attribute), 65
template_name (shoop.admin.modules.orders.views.refund.OrderCreateRefundView attribute), 64
template_name (shoop.admin.modules.orders.views.shipment.OrderCreateShipmentView attribute), 66
template_name (shoop.admin.modules.permission_groups.views.edit.PermissionGroupEditView attribute), 72
template_name (shoop.admin.modules.permission_groups.views.PermissionGroupEditView attribute), 73
template_name (shoop.admin.modules.product_types.views.edit.ProductTypeEditView attribute), 74
template_name (shoop.admin.modules.product_types.views.ProductTypeEditView attribute), 75
template_name (shoop.admin.modules.products.views.edit.ProductEditView attribute), 84
template_name (shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSellEditView attribute), 85
template_name (shoop.admin.modules.products.views.edit_media.ProductMediaEditView attribute), 86
template_name (shoop.admin.modules.products.views.edit_package.ProductPackageEditView attribute), 86
template_name (shoop.admin.modules.products.views.edit_variation.ProductVariationEditView attribute), 88
template_name (shoop.admin.modules.products.views.ProductCrossSellEditView attribute), 89
template_name (shoop.admin.modules.products.views.ProductEditView attribute), 89
template_name (shoop.admin.modules.products.views.ProductMediaEditView attribute), 90

template_name (shoop.admin.modules.products.views.ProductPackageView attribute), 90
 template_name (shoop.admin.modules.products.views.ProductVariationView attribute), 90
 template_name (shoop.admin.modules.sales_units.views.edit.SalesUnitEditView attribute), 92
 template_name (shoop.admin.modules.sales_units.views.SalesUnitListView attribute), 92
 template_name (shoop.admin.modules.service_providers.views.edit.ServiceProviderEditView attribute), 93
 template_name (shoop.admin.modules.services.behavior_forms.edit.BehaviorFormEditView attribute), 97
 template_name (shoop.admin.modules.shops.views.edit.ShopEditView attribute), 103
 template_name (shoop.admin.modules.shops.views.ShopEditView attribute), 104
 template_name (shoop.admin.modules.suppliers.views.edit.SupplierEditView attribute), 105
 template_name (shoop.admin.modules.suppliers.views.SupplierListView attribute), 106
 template_name (shoop.admin.modules.system.views.telemetry.TelemetryView attribute), 107
 template_name (shoop.admin.modules.taxes.views.CustomerTaxGroupEditView attribute), 111
 template_name (shoop.admin.modules.taxes.views.edit.CustomerTaxGroupEditView attribute), 108
 template_name (shoop.admin.modules.taxes.views.edit.TaxClassEditView attribute), 109
 template_name (shoop.admin.modules.taxes.views.edit.TaxEditView attribute), 108
 template_name (shoop.admin.modules.taxes.views.TaxClassEditView attribute), 110
 template_name (shoop.admin.modules.taxes.views.TaxEditView attribute), 110
 template_name (shoop.admin.modules.users.views.detail.UserDetailView attribute), 112
 template_name (shoop.admin.modules.users.views.password_change.PasswordChangeView attribute), 113
 template_name (shoop.admin.modules.users.views.password_reset.PasswordResetView attribute), 113
 template_name (shoop.admin.modules.users.views.permissions_change.PermissionsChangeView attribute), 114
 template_name (shoop.admin.modules.users.views.UserChangePermissionsView attribute), 115
 template_name (shoop.admin.modules.users.views.UserChangePermissionsView attribute), 115
 template_name (shoop.admin.modules.users.views.UserDetailView attribute), 114
 template_name (shoop.admin.modules.users.views.UserResetPasswordView attribute), 115
 template_name (shoop.admin.utils.picotable.PicotableView attribute), 120
 template_name (shoop.admin.views.dashboard.DashboardView attribute), 123
 template_name (shoop.admin.views.menu.MenuView attribute), 123
 template_name (shoop.campaigns.admin_module.form_parts.BaseFormPart attribute), 142
 template_name (shoop.campaigns.admin_module.views.CouponEditView attribute), 141
 template_name (shoop.default_tax.admin_module.views.TaxRuleEditView attribute), 266
 template_name (shoop.front.apps.auth.views.LoginView attribute), 270
 template_name (shoop.front.apps.auth.views.LogoutView attribute), 270
 template_name (shoop.front.apps.auth.views.RecoverPasswordCompleteView attribute), 271
 template_name (shoop.front.apps.auth.views.RecoverPasswordConfirmView attribute), 270
 template_name (shoop.front.apps.auth.views.RecoverPasswordSentView attribute), 271
 template_name (shoop.front.apps.auth.views.RecoverPasswordView attribute), 270
 template_name (shoop.front.apps.customer_information.views.CompanyEditView attribute), 272
 template_name (shoop.front.apps.customer_information.views.CustomerEditView attribute), 272
 template_name (shoop.front.apps.personal_order_history.views.OrderDetailView attribute), 273
 template_name (shoop.front.apps.personal_order_history.views.OrderListView attribute), 273
 template_name (shoop.front.apps.registration.views.ActivationView attribute), 274
 template_name (shoop.front.apps.registration.views.RegistrationViewMixing attribute), 274
 template_name (shoop.front.apps.simple_search.views.SearchView attribute), 276
 template_name (shoop.front.checkout.addresses.AddressesPhase attribute), 282
 template_name (shoop.front.checkout.confirm.ConfirmPhase attribute), 283
 template_name (shoop.front.checkout.empty.EmptyPhase attribute), 283
 template_name (shoop.front.checkout.methods.MethodsPhase attribute), 284
 template_name (shoop.front.checkout.single_page.SingleCheckoutPhase attribute), 285
 template_name (shoop.front.views.basket.DefaultBasketView attribute), 291
 template_name (shoop.front.views.category.CategoryView attribute), 291
 template_name (shoop.front.views.index.IndexView attribute), 292
 template_name (shoop.front.views.order.OrderCompleteView attribute), 292
 template_name (shoop.front.views.order.OrderRequiresVerificationView attribute), 292

template_name (shoop.front.views.product.ProductDetailView attribute), 293	TemplateHelpers (class in shoop.front.apps.simple_search.template_helpers), 276
template_name (shoop.notify.admin_module.views.edit.ScriptEditView attribute), 299	TemplateUse (class in shoop.notify), 315
template_name (shoop.notify.admin_module.views.editor.EditorScriptContentView attribute), 299	TestScriptContentView (class in shoop.notify.enums), 309
template_name (shoop.notify.admin_module.views.EditScriptContentView attribute), 300	test() (shoop.admin.utils.search.FuzzyMatcher method), 319
template_name (shoop.notify.admin_module.views.ScriptEditView attribute), 300	test() (shoop.notify.base.Condition method), 308
template_name (shoop.simple_cms.admin_module.views.PagesEditView attribute), 317	test() (shoop.notify.Condition method), 315
template_name (shoop.simple_cms.plugins.PageLinksPlugin attribute), 320	test() (shoop.notify.conditions.Empty method), 303
template_name (shoop.simple_cms.views.PageView attribute), 320	test() (shoop.notify.conditions.NonEmpty method), 304
template_name (shoop.simple_supplier.admin_module.views.StocksListView attribute), 322	test() (shoop.notify.conditions.simple.BaseEqual method), 302
template_name (shoop.testing.admin_module.mock_view.MockableView attribute), 326	test() (shoop.notify.conditions.simple.CaseInsensitiveStringEqual method), 303
template_name (shoop.testing.simple_checkout_phase.TestMethodCheckoutPhase attribute), 334	test() (shoop.notify.conditions.simple.Empty method), 303
template_name (shoop.themes.classic_gray.plugins.ProductCrossSellsPlugin attribute), 337	test() (shoop.notify.conditions.simple.NonEmpty method), 304
template_name (shoop.themes.classic_gray.plugins.ProductHighlightPlugin attribute), 337	TestingAdminModule (class in shoop.testing.simple_checkout_phase), 334
template_name (shoop.xtheme.admin_module.views.ThemeConfigDetailView attribute), 352	TestPaymentCheckoutPhase (class in shoop.testing.simple_checkout_phase), 335
template_name (shoop.xtheme.admin_module.views.ThemeConfigView attribute), 351	TestPaymentCheckoutPhaseForm (class in shoop.testing.simple_checkout_phase), 334
template_name (shoop.xtheme.plugins.category_links.CategoryLinksPlugin attribute), 353	TestShipmentCheckoutPhase (class in shoop.testing.simple_checkout_phase), 334
template_name (shoop.xtheme.plugins.image.ImagePlugin attribute), 354	TestShipmentCheckoutPhaseForm (class in shoop.testing.simple_checkout_phase), 334
template_name (shoop.xtheme.plugins.social_media_links.SocialMediaLinksPlugin attribute), 355	Text (class in shoop.notify.typology), 312
template_name (shoop.xtheme.TemplatedPlugin attribute), 375	TextEqual (class in shoop.notify.conditions), 304
template_name (shoop.xtheme.views.editor.EditorView attribute), 358	TextFilter (class in shoop.admin.utils.picotable), 119
template_object_name (shoop.front.views.category.CategoryView attribute), 291	Theme (class in shoop.xtheme), 375
template_use (shoop.notify.Action attribute), 314	ThemeConfigDetailView (class in shoop.xtheme.admin_module.views), 352
template_use (shoop.notify.actions.email.SendEmail attribute), 297	ThemeConfigView (class in shoop.xtheme.admin_module.views), 351
template_use (shoop.notify.actions.SendEmail attribute), 298	ThemeSettings (class in shoop.xtheme.models), 366
template_use (shoop.notify.base.Action attribute), 308	ThemeSettings.DoesNotExist, 366
templated_plugin_factory() (in module shoop.xtheme), 377	ThemeSettings.MultipleObjectsReturned, 366
TemplatedBinding (class in shoop.notify.base), 307	thumbnail() (in shoop.front.templatetags.thumbnails), 290
TemplatedFormDef (class in shoop.admin.form_part), 127	thumbnailer() (in shoop.front.templatetags.thumbnails), 290
TemplatedPlugin (class in shoop.xtheme), 375	TIMEDELTA (shoop.core.models.AttributeType attribute), 189
	timezone (shoop.core.models.Contact attribute), 193
	title (shoop.addons.admin_module.views.AddonUploadConfirmView attribute), 40
	title (shoop.addons.admin_module.views.AddonUploadView attribute), 40

[title \(shoop.addons.admin_module.views.upload.AddonUploadConfirmView attribute\), 39](#)
[title \(shoop.addons.admin_module.views.upload.AddonUploadView attribute\), 39](#)
[title \(shoop.addons.reloader.DevServerReloadMethod attribute\), 41](#)
[title \(shoop.addons.reloader.GunicornReloadMethod attribute\), 42](#)
[title \(shoop.addons.reloader.ModWSGIReloadMethod attribute\), 42](#)
[title \(shoop.addons.reloader.ReloadMethod attribute\), 41](#)
[title \(shoop.addons.reloader.UwsgiReloadMethod attribute\), 41](#)
[title \(shoop.admin.modules.media.views.MediaBrowserView attribute\), 60](#)
[title \(shoop.admin.modules.orders.views.edit.OrderEditView attribute\), 62](#)
[title \(shoop.admin.modules.orders.views.OrderEditView attribute\), 67](#)
[title \(shoop.admin.modules.service_providers.views.ServiceProviderEditView attribute\), 93](#)
[title \(shoop.admin.modules.users.views.password.UserChangePasswordView attribute\), 113](#)
[title \(shoop.admin.modules.users.views.password.UserResetPasswordView attribute\), 113](#)
[title \(shoop.admin.modules.users.views.permissions.UserChangePermissionsView attribute\), 114](#)
[title \(shoop.admin.modules.users.views.UserChangePasswordView attribute\), 115](#)
[title \(shoop.admin.modules.users.views.UserChangePermissionsView attribute\), 115](#)
[title \(shoop.admin.modules.users.views.UserResetPasswordView attribute\), 115](#)
[title \(shoop.core.models.ProductMedia attribute\), 219](#)
[title \(shoop.front.checkout.addresses.AddressesPhase attribute\), 282](#)
[title \(shoop.front.checkout.CheckoutPhaseViewMixin attribute\), 286](#)
[title \(shoop.front.checkout.confirm.ConfirmPhase attribute\), 283](#)
[title \(shoop.front.checkout.empty.EmptyPhase attribute\), 283](#)
[title \(shoop.front.checkout.methods.MethodsPhase attribute\), 284](#)
[title \(shoop.simple_cms.models.Page attribute\), 319](#)
[title \(shoop.testing.simple_checkout_phase.TestPaymentCheckoutPhase attribute\), 335](#)
[title \(shoop.testing.simple_checkout_phase.TestShipmentCheckoutPhase attribute\), 335](#)
[to_dict\(\) \(shoop.core.order_creator.SourceLine method\), 242](#)
[to_immutable\(\) \(shoop.core.models.ImmutableAddress method\), 198](#)
[to_json\(\) \(shoop.admin.base.SearchResult method\), 125](#)
[to_json\(\) \(shoop.admin.utils.picotable.ChoicesFilter method\), 118](#)
[to_json\(\) \(shoop.admin.utils.picotable.Column method\), 119](#)
[to_json\(\) \(shoop.admin.utils.picotable.Filter method\), 118](#)
[to_json\(\) \(shoop.admin.utils.picotable.RangeFilter method\), 119](#)
[to_json\(\) \(shoop.admin.utils.picotable.TextFilter method\), 119](#)
[to_language_codes\(\) \(in module shoop.utils.multilanguage_model_form\), 344](#)
[to_python\(\) \(shoop.admin.forms.fields.PercentageField method\), 45](#)
[to_python\(\) \(shoop.admin.forms.fields.Select2MultipleField method\), 45](#)
[to_python\(\) \(shoop.utils.fields.RelaxedModelChoiceField method\), 340](#)
[to_python\(\) \(shoop.front.apps.auth.forms.RecoverPasswordForm attribute\), 269](#)
[to_python\(\) \(shoop.front.apps.auth.views.RecoverPasswordConfirmView attribute\), 271](#)
[toolbar_class \(shoop.admin.toolbar\), 130](#)
[toolbar_class \(shoop.admin.modules.products.views.edit_package.ProductPackageView attribute\), 86](#)
[toolbar_class \(shoop.admin.modules.products.views.edit_parent.ProductParentView attribute\), 87](#)
[toolbar_class \(shoop.admin.modules.products.views.edit_variation.ProductVariationView attribute\), 88](#)
[toolbar_class \(shoop.admin.modules.products.views.ProductPackageView attribute\), 90](#)
[toolbar_class \(shoop.admin.modules.products.views.ProductVariationView attribute\), 90](#)
[total_discount \(shoop.core.order_creator.OrderSource attribute\), 241](#)
[total_gross_weight \(shoop.core.order_creator.OrderSource attribute\), 241](#)
[total_price \(shoop.core.order_creator.OrderSource attribute\), 241](#)
[total_price_of_products \(shoop.core.order_creator.OrderSource attribute\), 241](#)
[total_products \(shoop.core.models.Shipment attribute\), 228](#)
[TotalPriceDisplayFilter \(class in shoop.core.utils.price_display\), 258](#)
[TranslatableField \(class in shoop.xtheme.plugins.forms\), 354](#)
[TranslatableFieldWidget \(class in shoop.xtheme.plugins.widgets\), 356](#)
[TranslatableShoopModel \(class in shoop.core.models\), 235](#)
[TranslatableShoopModel.Meta \(class in shoop.core.models\), 235](#)

- TRANSLATED_STRING
(shoop.core.models.AttributeType attribute), 189
- translated_string_value (shoop.core.models.ProductAttribute attribute), 218
- translations (shoop.campaigns.models.BasketCampaign attribute), 171
- translations (shoop.campaigns.models.campaigns.BasketCampaign attribute), 158
- translations (shoop.campaigns.models.campaigns.CatalogCampaign attribute), 158
- translations (shoop.campaigns.models.CatalogCampaign attribute), 175
- translations (shoop.core.models.Attribute attribute), 188
- translations (shoop.core.models.Category attribute), 190
- translations (shoop.core.models.ContactGroup attribute), 194
- translations (shoop.core.models.CustomerTaxGroup attribute), 195
- translations (shoop.core.models.FixedCostBehaviorComponent attribute), 197
- translations (shoop.core.models.OrderStatus attribute), 205
- translations (shoop.core.models.PaymentMethod attribute), 207
- translations (shoop.core.models.Product attribute), 214, 218
- translations (shoop.core.models.ProductAttribute attribute), 218
- translations (shoop.core.models.ProductMedia attribute), 219
- translations (shoop.core.models.ProductType attribute), 220
- translations (shoop.core.models.ProductVariationVariable attribute), 221
- translations (shoop.core.models.ProductVariationVariableValue attribute), 221
- translations (shoop.core.models.SalesUnit attribute), 223
- translations (shoop.core.models.ShippingMethod attribute), 229
- translations (shoop.core.models.Shop attribute), 230
- translations (shoop.core.models.Tax attribute), 235
- translations (shoop.core.models.TaxClass attribute), 235
- translations (shoop.core.models.WaivingCostBehaviorComponent attribute), 236
- translations (shoop.core.models.WeightBasedPriceRange attribute), 236
- translations (shoop.simple_cms.models.Page attribute), 318
- try_parse_date() (in module shoop.utils.dates), 339
- try_parse_decimal_string() (in module shoop.utils.numbers), 344
- try_parse_time() (in module shoop.utils.dates), 339
- try_reverse() (in module shoop.admin.toolbar), 130
- try_send_telemetry() (in module shoop.core.telemetry), 263
- Type (class in shoop.notify.typology), 311
- type (shoop.admin.dashboard.blocks.DashboardBlock attribute), 42
- type (shoop.admin.dashboard.blocks.DashboardChartBlock attribute), 43
- type (shoop.admin.dashboard.blocks.DashboardContentBlock attribute), 42
- type (shoop.admin.dashboard.blocks.DashboardValueBlock attribute), 42
- type (shoop.admin.dashboard.DashboardBlock attribute), 44
- type (shoop.admin.dashboard.DashboardChartBlock attribute), 44
- type (shoop.admin.dashboard.DashboardContentBlock attribute), 44
- type (shoop.admin.dashboard.DashboardValueBlock attribute), 44
- type (shoop.admin.utils.picotable.ChoicesFilter attribute), 118
- type (shoop.admin.utils.picotable.Filter attribute), 118
- type (shoop.admin.utils.picotable.RangeFilter attribute), 119
- type (shoop.admin.utils.picotable.TextFilter attribute), 119
- type (shoop.campaigns.models.Campaign attribute), 173
- type (shoop.campaigns.models.campaigns.Campaign attribute), 157
- type (shoop.core.models.Attribute attribute), 188
- type (shoop.core.models.OrderLine attribute), 203
- type (shoop.core.models.Product attribute), 214, 218
- type (shoop.core.models.ProductCrossSell attribute), 218
- type (shoop.core.models.Supplier attribute), 234
- type (shoop.front.basket.objects.BasketLine attribute), 279
- type (shoop.testing.factories.ProductFactory attribute), 330
- ## U
- uncache() (shoop.core.order_creator.OrderSource method), 241
- UNDISCLOSED (shoop.core.models.Gender attribute), 197
- Unflattenable, 366
- UNILINGUAL (shoop.notify.enums.TemplateUse attribute), 309
- UNILINGUAL (shoop.notify.TemplateUse attribute), 315
- unit_discount_amount (shoop.core.pricing.Priceful attribute), 248
- unit_matches_with() (shoop.core.pricing.Price method), 245
- unit_matches_with() (shoop.utils.money.Money method), 344

- unit_matches_with() (shoop.utils.numbers.UnittedDecimal method), 345
- UnitMixupError, 345
- UnittedDecimal (class in shoop.utils.numbers), 345
- unlink_from_parent() (shoop.core.models.Product method), 214, 218
- unprotected_fields (shoop.core.models.Tax attribute), 235
- unread_for_user() (shoop.notify.models.notification.Notification method), 304
- UnsavedForeignKey (class in shoop.core.fields), 185
- unserialize() (shoop.notify.base.ScriptItem class method), 308
- unserialize() (shoop.notify.script.Step class method), 310
- unserialize() (shoop.notify.typology.Enum method), 313
- unserialize() (shoop.notify.typology.Model method), 313
- unserialize() (shoop.notify.typology.Type method), 312
- unserialize() (shoop.xtheme.layout.Layout class method), 363
- unserialize() (shoop.xtheme.layout.LayoutCell class method), 362
- unserialize() (shoop.xtheme.layout.LayoutRow class method), 362
- UNSTOCKED (shoop.core.models.StockBehavior attribute), 233
- UNTRANSLATED_STRING (shoop.core.models.AttributeType attribute), 189
- update() (shoop.core.order_creator.OrderSource method), 241
- update() (shoop.core.order_creator.SourceLine method), 242
- update_customers_groups() (in module shoop.campaigns.signal_handlers), 178
- update_from_order() (shoop.core.order_creator.OrderSource method), 241
- update_line() (shoop.front.basket.objects.BaseBasket method), 279
- update_module_attributes() (in module shoop.utils), 350
- update_order_from_source() (shoop.core.order_creator.OrderModifier method), 239
- update_order_from_state() (in module shoop.admin.modules.orders.views.edit), 62
- update_order_from_state() (shoop.admin.modules.orders.json_order_creator.JsonOrderCreator method), 70
- update_order_line_from_product() (in module shoop.core.shortcuts), 250
- update_quantity() (shoop.front.basket.update_methods.BasketUpdateMethod method), 281
- update_settings() (shoop.xtheme.models.ThemeSettings method), 366
- update_shipping_status() (shoop.core.models.Order method), 202
- update_stock() (shoop.core.models.Supplier method), 234
- update_stock() (shoop.core.suppliers.base.BaseSupplierModule method), 251
- update_stock() (shoop.core.suppliers.BaseSupplierModule method), 252
- update_stock() (shoop.simple_supplier.module.SimpleSupplierModule method), 324
- update_stocks() (shoop.core.models.Supplier method), 234
- update_stocks() (shoop.core.suppliers.base.BaseSupplierModule method), 251
- update_stocks() (shoop.core.suppliers.BaseSupplierModule method), 252
- URL (class in shoop.notify.typology), 312
- url (shoop.admin.base.Resolvable attribute), 125
- url (shoop.core.models.ProductMedia attribute), 219
- url (shoop.notify.models.Notification attribute), 306
- url (shoop.notify.models.notification.Notification attribute), 305
- url (shoop.simple_cms.models.Page attribute), 319
- url_name_prefix (shoop.admin.modules.services.PaymentMethodModule attribute), 102
- url_name_prefix (shoop.admin.modules.services.ServiceModule attribute), 101
- url_name_prefix (shoop.admin.modules.services.ShippingMethodModule attribute), 101
- url_prefix (shoop.admin.modules.services.PaymentMethodModule attribute), 102
- url_prefix (shoop.admin.modules.services.ServiceModule attribute), 101
- url_prefix (shoop.admin.modules.services.ShippingMethodModule attribute), 101
- URLActionButton (class in shoop.admin.toolbar), 128
- usages (shoop.campaigns.models.campaigns.Coupon attribute), 160
- usages (shoop.campaigns.models.Coupon attribute), 177
- use() (shoop.campaigns.models.campaigns.Coupon method), 160
- use() (shoop.campaigns.models.Coupon method), 177
- use_code() (shoop.campaigns.modules.BasketCampaignModule method), 178
- use_code() (shoop.core.order_creator.OrderSourceModifierModule method), 241
- User (class in shoop.core.models), 205
- user (shoop.core.models.OrderLogEntry attribute), 205
- user (shoop.core.models.PersonContact attribute), 209
- user (shoop.utils.analog.BaseLogEntry attribute), 338
- UserChangePasswordView (class in shoop.admin.modules.users.views), 114
- UserChangePasswordView (class in shoop.admin.modules.users.views.password), 113

UserChangePermissionsView	(class in shoop.admin.modules.users.views), 115	in value (shoop.campaigns.models.basket_conditions.BasketMaxTotalAmount attribute), 149
UserChangePermissionsView	(class in shoop.admin.modules.users.views.permissions), 114	in value (shoop.campaigns.models.basket_conditions.BasketMaxTotalProduct attribute), 148
UserDetailToolbar	(class in shoop.admin.modules.users.views.detail), 111	in value (shoop.campaigns.models.basket_conditions.BasketTotalAmountCon attribute), 147
UserDetailView	(class in shoop.admin.modules.users.views), 114	in value (shoop.campaigns.models.basket_conditions.BasketTotalProductAmo attribute), 147
UserDetailView	(class in shoop.admin.modules.users.views.detail), 111	in value (shoop.campaigns.models.basket_effects.BasketDiscountAmount attribute), 153
UserFactory	(class in shoop.testing.factories), 329	in value (shoop.campaigns.models.basket_effects.BasketDiscountPercentage attribute), 154
UserListView	(class in shoop.admin.modules.users.views), 114	in value (shoop.campaigns.models.product_effects.ProductDiscountAmount attribute), 168
UserListView	(class in shoop.admin.modules.users.views.list), 112	in value (shoop.campaigns.models.product_effects.ProductDiscountPercentage attribute), 169
UserModule	(class in shoop.admin.modules.users), 116	in value (shoop.core.models.ConfigurationItem attribute), 192
username	(shoop.testing.factories.UserFactory attribute), 329	in value (shoop.core.models.ProductVariationVariableValue attribute), 221
UserResetPasswordView	(class in shoop.admin.modules.users.views), 115	in value (shoop.core.models.ProductVariationVariableValue attribute), 221
UserResetPasswordView	(class in shoop.admin.modules.users.views.password), 113	in value (shoop.core.models.ProductVariationVariableValue attribute), 221
UwsgiReloadMethod	(class in shoop.addons.reloader), 41	in value (shoop.core.models.ProductVariationVariableValue attribute), 221
V		
validate()	(in module shoop.core.utils.tax_numbers), 259	in value (shoop.core.models.ProductVariationVariableValue attribute), 221
validate()	(shoop.notify.typology.Type method), 312	in value (shoop.core.models.ProductVariationVariableValue attribute), 221
validate_max	(shoop.admin.modules.products.forms.base_forms.BaseProductMediaFormSet attribute), 76	in value (shoop.core.models.ProductVariationVariableValue attribute), 221
validate_max	(shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSellFormSet attribute), 84	in value (shoop.core.models.ProductVariationVariableValue attribute), 221
validate_max	(shoop.admin.modules.products.views.edit_media.ProductMediaFormSet attribute), 85	in value (shoop.core.models.ProductVariationVariableValue attribute), 221
validate_max	(shoop.admin.modules.services.behavior_form_part.BehaviorFormSet attribute), 96	in value (shoop.core.models.ProductVariationVariableValue attribute), 221
validate_max	(shoop.campaigns.admin_module.form_sets.BaseFormSet attribute), 144	in value (shoop.core.models.ProductVariationVariableValue attribute), 221
validate_min	(shoop.admin.modules.products.forms.base_forms.BaseProductMediaFormSet attribute), 76	in value (shoop.core.models.ProductVariationVariableValue attribute), 221
validate_min	(shoop.admin.modules.products.views.edit_cross_sell.ProductCrossSellFormSet attribute), 84	in value (shoop.core.models.ProductVariationVariableValue attribute), 221
validate_min	(shoop.admin.modules.products.views.edit_media.ProductMediaFormSet attribute), 85	in value (shoop.core.models.ProductVariationVariableValue attribute), 221
validate_min	(shoop.admin.modules.services.behavior_form_part.BehaviorFormSet attribute), 96	in value (shoop.core.models.ProductVariationVariableValue attribute), 221
validate_min	(shoop.campaigns.admin_module.form_sets.BaseFormSet attribute), 144	in value (shoop.core.models.ProductVariationVariableValue attribute), 221
validate_templates_configuration()	(in module shoop.apps.settings), 133	in value (shoop.core.models.ProductVariationVariableValue attribute), 221

- values() (shoop.utils.setup.Setup method), 349
- Variable (class in shoop.notify), 315
- Variable (class in shoop.notify.base), 306
- variable (shoop.core.models.ProductVariationVariableValue attribute), 222
- VARIABLE_ONLY (shoop.notify.ConstantUse attribute), 315
- VARIABLE_ONLY (shoop.notify.enums.ConstantUse attribute), 309
- VARIABLE_OR_CONSTANT (shoop.notify.ConstantUse attribute), 315
- VARIABLE_OR_CONSTANT (shoop.notify.enums.ConstantUse attribute), 309
- VARIABLE_VARIATION_PARENT (shoop.core.models.ProductMode attribute), 220
- variables (shoop.front.notify_events.OrderReceived attribute), 294
- variables (shoop.notify.Action attribute), 314
- variables (shoop.notify.actions.AddNotification attribute), 298
- variables (shoop.notify.actions.AddOrderLogEntry attribute), 298
- variables (shoop.notify.actions.debug.SetDebugFlag attribute), 297
- variables (shoop.notify.actions.email.SendEmail attribute), 297
- variables (shoop.notify.actions.notification.AddNotification attribute), 297
- variables (shoop.notify.actions.order.AddOrderLogEntry attribute), 298
- variables (shoop.notify.actions.SendEmail attribute), 298
- variables (shoop.notify.actions.SetDebugFlag attribute), 299
- variables (shoop.notify.base.Action attribute), 309
- variables (shoop.notify.base.Base attribute), 307
- variables (shoop.notify.base.Condition attribute), 308
- variables (shoop.notify.base.Event attribute), 307
- variables (shoop.notify.base.ScriptItem attribute), 308
- variables (shoop.notify.Condition attribute), 315
- variables (shoop.notify.conditions.BooleanEqual attribute), 303
- variables (shoop.notify.conditions.Empty attribute), 303
- variables (shoop.notify.conditions.IntegerEqual attribute), 303
- variables (shoop.notify.conditions.LanguageEqual attribute), 303
- variables (shoop.notify.conditions.NonEmpty attribute), 304
- variables (shoop.notify.conditions.simple.BaseEqual attribute), 302
- variables (shoop.notify.conditions.simple.CaseInsensitiveStringEqual attribute), 303
- variables (shoop.notify.conditions.simple.Empty attribute), 302
- variables (shoop.notify.conditions.simple.NonEmpty attribute), 302
- variables (shoop.notify.conditions.TextEqual attribute), 304
- variables (shoop.notify.Event attribute), 315
- VariableVariationChildrenForm (class in shoop.admin.modules.products.forms), 82
- VariableVariationChildrenForm (class in shoop.admin.modules.products.forms.variable_variation_forms), 79
- VARIATION_CHILD (shoop.core.models.ProductMode attribute), 220
- variation_children (shoop.core.models.Product attribute), 214, 218
- variation_name (shoop.core.models.Product attribute), 214, 218
- variation_parent (shoop.core.models.Product attribute), 214, 218
- variation_result_subs (shoop.core.models.Product attribute), 214, 218
- variation_result_supers (shoop.core.models.Product attribute), 214, 218
- variation_variables (shoop.core.models.Product attribute), 214, 218
- VariationChildrenFormPart (class in shoop.admin.modules.products.views.edit_variation), 88
- VariationVariablesDataForm (class in shoop.admin.modules.products.forms), 82
- VariationVariablesDataForm (class in shoop.admin.modules.products.forms.variable_variation_forms), 79
- VariationVariablesFormPart (class in shoop.admin.modules.products.views.edit_variation), 88
- vat_id (shoop.testing.factories.CompanyFactory attribute), 329
- VatCannotIdentifyValidationError, 259
- VatInvalidValidationError, 260
- VatValidationError, 259
- verbose_name (shoop.admin.ShoopAdminAppConfig attribute), 130
- verbose_name (shoop.api.ShoopApiAppConfig attribute), 132
- verbose_name (shoop.campaigns.apps.CampaignAppConfig attribute), 177
- verbose_name (shoop.campaigns.models.Campaign.Meta attribute), 173
- verbose_name (shoop.campaigns.models.campaigns.Campaign.Meta attribute), 157
- verbose_name (shoop.core.ShoopCoreAppConfig attribute), 264

[verbose_name \(shoop.customer_group_pricing.CustomerGroupPricingAppConfig attribute\), 265](#)
[verbose_name \(shoop.default_tax.AppConfig attribute\), 268](#)
[verbose_name \(shoop.front.apps.auth.AuthAppConfig attribute\), 271](#)
[verbose_name \(shoop.front.apps.customer_information.AppConfig attribute\), 273](#)
[verbose_name \(shoop.front.apps.personal_order_history.AppConfig attribute\), 273](#)
[verbose_name \(shoop.front.apps.registration.RegistrationAppConfig attribute\), 275](#)
[verbose_name \(shoop.front.apps.simple_order_notification.SimpleOrderNotificationAppConfig attribute\), 276](#)
[verbose_name \(shoop.front.apps.simple_search.SimpleSearchAppConfig attribute\), 277](#)
[verbose_name \(shoop.front.ShoopFrontAppConfig attribute\), 296](#)
[verbose_name \(shoop.guide.apps.AppConfig attribute\), 296](#)
[verbose_name \(shoop.order_printouts.AppConfig attribute\), 316](#)
[verbose_name \(shoop.simple_cms.AppConfig attribute\), 321](#)
[verbose_name \(shoop.simple_supplier.ShoopSimpleSupplierAppConfig attribute\), 325](#)
[verbose_name \(shoop.testing.ShoopTestingAppConfig attribute\), 336](#)
[verbose_name \(shoop.themes.classic_gray.ClassicGrayThemeAppConfig attribute\), 337](#)
[verbose_name \(shoop.utils.ShoopUtilsAppConfig attribute\), 351](#)
[verbose_name_plural \(shoop.campaigns.models.Campaign.Meta attribute\), 173](#)
[verbose_name_plural \(shoop.campaigns.models.campaigns.Campaign.Meta attribute\), 157](#)
[verify_bindings\(\) \(shoop.notify.base.ScriptItem method\), 307](#)
[verify_mode\(\) \(shoop.core.models.Product method\), 214, 218](#)
[verify_vat\(\) \(in module shoop.core.utils.vat\), 260](#)
[VersionedCache \(class in shoop.core.cache\), 182](#)
[VersionedCache \(class in shoop.core.cache.impl\), 181](#)
[view_template \(shoop.admin.modules.services.PaymentMethodModule attribute\), 102](#)
[view_template \(shoop.admin.modules.services.ServiceModule attribute\), 101](#)
[view_template \(shoop.admin.modules.services.ShippingMethodModule attribute\), 101](#)
[ViewConfig \(class in shoop.xtheme.view_config\), 372](#)
[viewset_factory\(\) \(in module shoop.api.factories\), 131](#)
[visibility \(shoop.core.models.Category attribute\), 191](#)
[visibility_groups \(shoop.core.models.Category attribute\), 191](#)
[visible_for_order\(\) \(shoop.admin.base.OrderSection static method\), 126](#)
[visible_for_order\(\) \(shoop.admin.modules.orders.sections.ContentsOrderSection static method\), 71](#)
[visible_for_order\(\) \(shoop.admin.modules.orders.sections.LogEntriesOrderSection static method\), 71](#)
[visible_for_order\(\) \(shoop.admin.modules.orders.sections.PaymentOrderSection static method\), 71](#)
[visible_for_order\(\) \(shoop.admin.modules.orders.sections.ProductVariationLinkStatus static method\), 221](#)
[visible_for_order\(\) \(shoop.admin.modules.orders.sections.RegistrationAppConfig static method\), 330](#)
[visible_for_order\(\) \(shoop.admin.modules.orders.sections.SimpleOrderNotificationAppConfig static method\), 318](#)
[visible_for_order\(\) \(shoop.admin.modules.orders.sections.SimpleSearchAppConfig static method\), 194](#)
[visible_for_order\(\) \(shoop.admin.modules.orders.sections.ShippingMethodModule static method\), 194](#)
[VISIBLE_TO_ALL \(shoop.core.models.CategoryVisibility attribute\), 191](#)
[VISIBLE_TO_ALL \(shoop.core.models.ProductVisibility attribute\), 222](#)
[VISIBLE_TO_GROUPS \(shoop.core.models.CategoryVisibility attribute\), 191](#)
[VISIBLE_TO_GROUPS \(shoop.core.models.ProductVisibility attribute\), 222](#)
[VISIBLE_TO_LOGGED_IN \(shoop.core.models.CategoryVisibility attribute\), 191](#)
[VISIBLE_TO_LOGGED_IN \(shoop.core.models.ProductVisibility attribute\), 222](#)
W
[WaivingCostBehaviorComponent \(class in shoop.core.models\), 236](#)
[waivingcostbehaviorcomponent \(shoop.core.models.ServiceBehaviorComponent attribute\), 226](#)
[WaivingCostBehaviorComponent.DoesNotExist, 236](#)
[WaivingCostBehaviorComponent.MultipleObjectsReturned, 236](#)
[WaivingCostBehaviorComponentForm \(class in shoop.admin.modules.services.forms\), 98](#)

- WaivingCostBehaviorComponentForm.Meta (class in shoop.admin.modules.services.forms), 98
- WARNING (shoop.utils.analog.LogEntryKind attribute), 338
- WeightBasedPriceRange (class in shoop.core.models), 236
- WeightBasedPriceRange.DoesNotExist, 236
- WeightBasedPriceRange.MultipleObjectsReturned, 236
- WeightBasedPricingBehaviorComponent (class in shoop.core.models), 236
- weightbasedpricingbehaviorcomponent (shoop.core.models.ServiceBehaviorComponent attribute), 226
- WeightBasedPricingBehaviorComponent.DoesNotExist, 236
- WeightBasedPricingBehaviorComponent.MultipleObjectsReturned, 237
- WeightBasedPricingFormPart (class in shoop.admin.modules.services.weight_based_pricing), 95
- WeightLimitsBehaviorComponent (class in shoop.core.models), 237
- weightlimitsbehaviorcomponent (shoop.core.models.ServiceBehaviorComponent attribute), 226
- WeightLimitsBehaviorComponent.DoesNotExist, 237
- WeightLimitsBehaviorComponent.MultipleObjectsReturned, 237
- WeightLimitsBehaviorComponentForm (class in shoop.admin.modules.services.forms), 99
- WeightLimitsBehaviorComponentForm.Meta (class in shoop.admin.modules.services.forms), 99
- widget (shoop.admin.forms.fields.Select2MultipleField attribute), 45
- widget (shoop.xtheme.plugins.forms.TranslatableField attribute), 354
- widget (shoop.xtheme.plugins.image.ImageIDField attribute), 354
- widget (shoop.xtheme.plugins.widgets.XThemeModelChoiceField attribute), 357
- widget_attrs() (shoop.admin.forms.fields.PercentageField method), 45
- widget_attrs() (shoop.admin.forms.fields.Select2MultipleField method), 45
- widget_attrs() (shoop.core.fields.FormattedDecimalFormField method), 184
- widgets (shoop.admin.modules.categories.views.edit.CategoryBaseForm.Meta attribute), 48
- widgets (shoop.admin.modules.product_types.views.edit.ProductTypeForm.Meta attribute), 74
- widgets (shoop.admin.modules.products.forms.base_forms.ProductBaseForm.Meta attribute), 75
- widgets (shoop.admin.modules.products.forms.ProductBaseForm.Meta attribute), 80
- widgets (shoop.admin.modules.suppliers.views.edit.SupplierForm.Meta attribute), 105
- with_link() (shoop.utils.excs.Problem method), 339
- wrap_pdf_in_response() (in shoop.order_printouts.pdf_export module), 316
- wrap_with_permissions() (shoop.admin.utils.urls.AdminRegexURLPattern method), 121
- write_po_file() (shoop.core.management.commands.makemessages.Command method), 186
- ## X
- xtheme_injection (shoop.xtheme.views.editor.EditorView attribute), 358
- XthemeAdminModule (class in shoop.xtheme.admin_module), 352
- XthemeEnvironment (class in shoop.xtheme.engine), 360
- XThemeModelChoiceField (class in shoop.xtheme.plugins.widgets), 356
- XThemeModelChoiceWidget (class in shoop.xtheme.plugins.widgets), 356
- XthemeNamespace (class in shoop.xtheme.template_ns), 371
- XthemeTemplate (class in shoop.xtheme.engine), 360