django-model-report Documentation

Release 0.1.8

juanpex

April 27, 2014

Contents

django-model-report is a Django application and library for reports integrated with highcharts.

CHAPTER 1

Demo

http://django-model-reports.herokuapp.com

1.1 User Guide

1.1.1 Installation

django-model-report is on the Python Package Index (PyPI), so it can be installed with standard Python tools like pip or easy_install.:

```
$ pip install django-model-report
```

- Add the `model_report` directory to your Python path.
- Add `model_report` to your `INSTALLED_APPS` setting.
- Create file “reports.py” within any application directory (just like admin.py).
- Edit the file “reports.py” and register your reports like this:

```
from anyapp.models import AnyModel
from model_report.report import reports, ReportAdmin

class AnyModelReport(ReportAdmin):
    title = _(’AnyModel Report Name’)
    model = AnyModel
    fields = [
        ’anymodelfield’,
    ]
    list_order_by = (’anymodelfield’,)
    type = ’report’

reports.register(’anymodel-report’, AnyModelReport)
```

- Activate your reports calling the autodiscover in `urls.py` (just like admin.py).:

```
from model_report import report
report.autodiscover()
```

- Add reports urls.:

```
urlpatterns = patterns('',
    ...
    (r'', include('model_report.urls')),
    ...
)
```

1.1.2 Example app

`cd example && ./manage.py runserver`

Username and password for admin are ‘admin’, ‘admin’.

1.1.3 Configuration

Extend your reports from `model_report.report.ReportAdmin()`:

```
from model_report.report import ReportAdmin

class ReportExample(ReportAdmin):
    pass
```

To configure the report see the `model_report.report.ReportAdmin()` documentation

1.1.4 Contributing

Clone the repo and help to be better this app :)

1.2 Modules

1.2.1 Modules

highcharts

Highchart Module

__init__

base

options

export_pdf

`model_report.export_pdf.`**`render_to_pdf`**(*report*, *template_src*, *context_dict*, *pdf_encoding=’UTF-8’*)

Render the report results to pdf format.

Keyword arguments: report – a report instance template_src – template file path context_dict – context dictionary pdf_encoding – encoding to render string

report

utils

class `model_report.utils.`**`ReportRow`**

Class to represent report rows

Attributes:

- `is_total` - defined as True if row is a group total row or report total row
- `is_caption` - TODO

`get_css_class``()`

Render css classes to this row into html representation

`is_value``()`

Evaluate True if the row is a normal row or not

class `model_report.utils.`**`ReportValue`**`(`*value*`)`

Class to represent cells values for report rows

Attributes:

- `value` - real value from database
- `is_report_total` - defined as True if value is for showing in report total row
- `is_group_total` - defined as True if value is for showing in group total row
- `is_value` - defined as True if value is for showing in normal row

`format``(`*value*, *instance*`)`

Format the value instance.

`text``()`

Render as text the value. This function also format the value.

`model_report.utils.`**`avg_column`**`(`*values*`)`

Average values for any column

`model_report.utils.`**`base_label`**`(`*report*, *field*`)`

Basic label

`model_report.utils.`**`count_column`**`(`*values*`)`

Count values for any column

`model_report.utils.`**`date_format`**`(`*value*, *instance*`)`

Format cell value to friendly date string

`model_report.utils.`**`round_format`**`(`*value*, *instance*`)`

Format value to render with 2 decimal places

`model_report.utils.`**`sum_column`**`(`*values*`)`

Sum values for any column

`model_report.utils.`**`usd_format`**`(`*value*, *instance*`)`

Format cell value to money

`model_report.utils.`**`yesno_format`**`(`*value*, *instance*`)`

Format boolean value to render Yes or No if True or False

widgets

class `model_report.widgets.`**`RangeField`**(*field_class*, *widget=<class 'django.forms.widgets.TextInput'>*, **args*, ***kwargs*)

Field to filter date values by range.

class `model_report.widgets.`**`RangeWidget`**(*widget*, **args*, ***kwargs*)

Render 2 inputs with vDatepicker class in order to select a date range.

Python Module Index

m

model_report.export_pdf, **??**
model_report.utils, **??**
model_report.widgets, **??**