
Django Report Scaffold Documentation

Release 1.0

Burke Software and Consulting LLC

October 03, 2016

1	Quickstart Guide	3
1.1	Installation	3
1.2	Creating your first report	3
1.3	Setting up templates	3
2	API Reference	5
2.1	ScaffoldReport	5
2.2	Filters	6
2.3	Fields	8
3	What is Django Report Scaffold?	9
	Python Module Index	11

Contents:

Quickstart Guide

1.1 Installation

1. `pip install django-report-scaffold`
2. Add `scaffold_report` to `INSTALLED_APPS`
3. Add `(r'^reports/', include('scaffold_report.urls'))`, to `urls.py`. I choose to name it “reports” but make it whatever you want.
4. It’s installed but won’t do anything yet!

1.2 Creating your first report

Create a file called `scaffold_reports.py` in your projects app folder. Here is a sample report to get started.:

```
from foo.models import Order
from scaffold_report.report import ScaffoldReport, scaffold_reports

class FooReport(ScaffoldReport):
    name = "Reports"
    model = Order
    filters = (
    )

scaffold_reports.register('foo_report', FooReport)
```

Now go to `/reports/foo_report/` to view it in a browser.

1.3 Setting up templates

Next you want to customize your templates to make the reports feel at home. The default template to use is `admin_base.html` which could be a base template for your project. You could also rename it.

Here is a example `admin_base.html`:

```
{% extends 'base.html' %}
{% load static %}

{% block scripts %}
```

```
{% include "scaffold_report/includes/header.html" %}
{% endblock %}

{% block content %}
{% endblock %}
```

base.html is a base template for the Foo project. This can be anything you want. In my base.html I have blocks called scripts and content. The content block is needed actually!

This method is easy but to customize further you could copy and change any of the scaffold report templates.

API Reference

2.1 ScaffoldReport

class `scaffold_report.report.ScaffoldReport`

Bases: `object`

Base class for any actual scaffold reports A scaffold report is named after UI effects for moving various filters and previews into the report building screen. All reports require customized options set by the programmer.

name = ''

Unique name of report.

name_verbose = None

Verbose name of report to show users.

model = None

Base model for this report's queryset.

preview_fields = ['id']

Array of field names to show on previews.

num_preview = 3

How many objects in the queryset should be show in preview.

filters = []

Filters that can be applied to the report.

report_buttons = []

Buttons will show in sidebar area. These can be premade reports and give the user more options than just clicking submit

permissions_required = []

Report is only viewable to those with these permissions. Will default to the model's change permission if not set

appy_template = None

A statically defined template. Could override `get_template` instead.

get_appy_template()

Return a appy template This could be hard coded or perhaps get `report_context` from a filter.

check_permissions(request)

Return true is user has permission to view page

get_name

Return `name_verbose` if it has been set; otherwise return `name`. Replaces all spaces with underscores.

get_queryset ()

Return a queryset of the model filtering any active filters

get_appy_context ()

Return a context dict for use in an appy template Acts a like context in Django template rendering.

report_to_list (user, preview=False)

Convert to python list

class scaffold_report.report.ReportButton

Bases: object

An alternative way to submit a report. Could be used for one off reports that behave differently.

name = ''

For button name attr

value = ''

For button value attr

accepts_queryset = True

When true the queryset is processed first (so filters will run) and passed to get_report

get_name

Return name_verbose if it has been set; otherwise return name. Replaces all spaces with underscores.

get_report (context=None)

This function will call to generate the actual report Should return a valid response

scaffold_report.report.autodiscover ()

Auto-discover INSTALLED_APPS report.py modules and fail silently when not present. Borrowed from django.contrib.admin

class scaffold_report.report.ScaffoldReportClassManager

Bases: object

Class to handle registered reports class. Borrowed from django-model-report Thanks!

2.2 Filters

class scaffold_report.filters.Filter (kwargs)**

Bases: object

A customized filter for querysets

name = None

Unique name of filter

verbose_name = None

Human readable name of filter

template_name = None

If set the filter will render using this django template If not set the filter will render using scaffold_report/filter.html

fields = None

Define fields here that will be appended together to make a generic form

form_class = None

Optional form class to use.

form = None

Optional form. If not set, an instance of the form_class will be used

raw_form_data = None

uncleaned_form data from the post

add_fields = []

Add these fields to the preview and spreadsheet reports

default = False

Show this filter as on by default

can_remove = True

User is able to delete this filter. Should be used with default = True.

can_add = True

User is able to add this filter

queryset_filter (*queryset, report_context=None, form=None*)

Allow custom handling of queryset Must return the queryset.

render_form ()

Render the form using a template Only called if template_name is defined

get_add_fields ()

Returns the fields to add to previews and spreadsheet reports

process_filter (*queryset, report_context=None*)

Run the actual filter based on client data

get_template_context ()

Get the context to be shown when rendering a template just for this filter

get_report_context (*report_context*)

Process any data that needs set for an entire report

build_form ()

Construct form out of fields or form

get_name ()

return unique name of this filter

class scaffold_report.filters.**DecimalCompareFilter** (**kwargs)

Bases: *scaffold_report.filters.Filter*

X greater, less, etc than decimal field

class scaffold_report.filters.**ModelChoiceFilter** (**kwargs)

Bases: *scaffold_report.filters.Filter*

Select object from a queryset

compare_field_string = None

String used in the Django orm filter function. See queryset_filter()

model = None

Model used for queryset. Set this for any object of such model.

queryset = None

queryset that populates the widget. Model is not needed if this is set.

get_queryset ()

Get the queryset that will populate the widget

class `scaffold_report.filters.ModelMultipleChoiceFilter` (**kwargs)
Bases: `scaffold_report.filters.ModelChoiceFilter`

Select multiple objects from a queryset

class `scaffold_report.filters.IntCompareFilter` (**kwargs)
Bases: `scaffold_report.filters.DecimalCompareFilter`

x greater, less, etc than int field

2.3 Fields

class `scaffold_report.fields.SimpleCompareField` (**kwargs)
Bases: `django.forms.fields.ChoiceField`

Field for selecting a comparison. Includes django syntax and verbose choices

Hey, Read the Docs!

What is Django Report Scaffold?

You have an app and it probably needs reports. Scaffold reports will help you streamline the process and present users with a consistent interface. Think of it like Django admin for reports. It features:

- Stock report filtering
- Highly extensible - Keep with stock filters or make your own.
- Export to xlsx, admin change list page, django-report-builder, or appy POD templates. Extend and add your own exports!
- Ready to go user interface allows users to quickly filter reports.
- Filter system encourages you to keep your report logic organized. Filters can change a queryset or change report context that might affect other filters.

f

`filters` (*Unix, Windows*), [6](#)

s

`scaffold_report.fields`, [8](#)

`scaffold_report.filters`, [6](#)

`scaffold_report.report`, [5](#)

A

accepts_queryset (scaffold_report.report.ReportButton attribute), 6
 add_fields (scaffold_report.filters.Filter attribute), 7
 appy_template (scaffold_report.report.ScaffoldReport attribute), 5
 autodiscover() (in module scaffold_report.report), 6

B

build_form() (scaffold_report.filters.Filter method), 7

C

can_add (scaffold_report.filters.Filter attribute), 7
 can_remove (scaffold_report.filters.Filter attribute), 7
 check_permissions() (scaffold_report.report.ScaffoldReport method), 5
 compare_field_string (scaffold_report.filters.ModelChoiceFilter attribute), 7

D

DecimalCompareFilter (class in scaffold_report.filters), 7
 default (scaffold_report.filters.Filter attribute), 7

F

fields (scaffold_report.filters.Filter attribute), 6
 Filter (class in scaffold_report.filters), 6
 filters (module), 6
 filters (scaffold_report.report.ScaffoldReport attribute), 5
 form (scaffold_report.filters.Filter attribute), 6
 form_class (scaffold_report.filters.Filter attribute), 6

G

get_add_fields() (scaffold_report.filters.Filter method), 7
 get_appy_context() (scaffold_report.report.ScaffoldReport method), 6

get_appy_template() (scaffold_report.report.ScaffoldReport method), 5
 get_name (scaffold_report.report.ReportButton attribute), 6
 get_name (scaffold_report.report.ScaffoldReport attribute), 5
 get_name() (scaffold_report.filters.Filter method), 7
 get_queryset() (scaffold_report.filters.ModelChoiceFilter method), 7
 get_queryset() (scaffold_report.report.ScaffoldReport method), 5
 get_report() (scaffold_report.report.ReportButton method), 6
 get_report_context() (scaffold_report.filters.Filter method), 7
 get_template_context() (scaffold_report.filters.Filter method), 7

I

IntCompareFilter (class in scaffold_report.filters), 8

M

model (scaffold_report.filters.ModelChoiceFilter attribute), 7
 model (scaffold_report.report.ScaffoldReport attribute), 5
 ModelChoiceFilter (class in scaffold_report.filters), 7
 ModelMultipleChoiceFilter (class in scaffold_report.filters), 7

N

name (scaffold_report.filters.Filter attribute), 6
 name (scaffold_report.report.ReportButton attribute), 6
 name (scaffold_report.report.ScaffoldReport attribute), 5
 name_verbose (scaffold_report.report.ScaffoldReport attribute), 5
 num_preview (scaffold_report.report.ScaffoldReport attribute), 5

P

`permissions_required` (scaffold_report.report.ScaffoldReport attribute), 5

`preview_fields` (scaffold_report.report.ScaffoldReport attribute), 5

`process_filter()` (scaffold_report.filters.Filter method), 7

Q

`queryset` (scaffold_report.filters.ModelChoiceFilter attribute), 7

`queryset_filter()` (scaffold_report.filters.Filter method), 7

R

`raw_form_data` (scaffold_report.filters.Filter attribute), 7

`render_form()` (scaffold_report.filters.Filter method), 7

`report_buttons` (scaffold_report.report.ScaffoldReport attribute), 5

`report_to_list()` (scaffold_report.report.ScaffoldReport method), 6

`ReportButton` (class in scaffold_report.report), 6

S

`scaffold_report.fields` (module), 8

`scaffold_report.filters` (module), 6

`scaffold_report.report` (module), 5

`ScaffoldReport` (class in scaffold_report.report), 5

`ScaffoldReportClassManager` (class in scaffold_report.report), 6

`SimpleCompareField` (class in scaffold_report.fields), 8

T

`template_name` (scaffold_report.filters.Filter attribute), 6

V

`value` (scaffold_report.report.ReportButton attribute), 6

`verbose_name` (scaffold_report.filters.Filter attribute), 6