
django*renderpdf*

Release 4.0.0

Apr 03, 2023

Contents

1	Index	3
2	Installation	5
3	Usage	7
4	API	9
4.1	PDFView	9
4.2	Helpers	9
5	Changelog	11
5.1	v4.0.0	11
5.2	v3.0.1	11
5.3	v3.0.0	11
5.4	v2.2.0	11
5.5	v2.1.0	12
5.6	v2.0.1	12
5.7	v2.0.0	12
6	Help	13
7	Background	15
8	Donations	17
9	Licence	19

django-renderpdf is a Django application for generating PDFs via Django.

Our intent is to properly integrate into Django, and have a very simple, yet flexible API. We follow common Django patterns, so rendering and exposing PDFs is simple.

CHAPTER 1

Index

CHAPTER 2

Installation

You'll first need to install the python package:

```
pip install django-renderpdf
```

You *don't* need to add it to `INSTALLED_APPS`.

This package depends on `weasyprint`, which will be installed as a dependency automatically. However, `weasyprint` itself has a few non-Python dependencies. Make sure the following works before proceeding:

```
python -c "import weasyprint"
```

If the above prints an error, please follow [weasyprint's installation documentation](#).

CHAPTER 3

Usage

As a usage example, let's assume we have to print some shipping labels to dispatch a product. We'll need to generate a PDF file for this to print.

```
from django.contrib.auth.mixins import LoginRequiredMixin
from django_renderpdf.views import PDFView

from myapp import Shipment

class LabelsView(LoginRequiredMixin, PDFView):
    """Generate labels for some Shipments.

    A PDFView behaves pretty much like a TemplateView, so you can treat it as such.
    """
    template_name = 'my_app/labels.html'

    def get_context_data(self, *args, **kwargs):
        """Pass some extra context to the template."""
        context = super().get_context_data(*args, **kwargs)

        context['shipments'] = Shipment.objects.filter(
            batch_id=kwargs['pk'],
        )

        return context
```

If anything in the above example seems completely new, I suggest you review the documentation for Django's [Class-based views](#).

You still need to include this view in your `urls.py` as usual:

```
from django.urls import path

path(
```

(continues on next page)

(continued from previous page)

```
    '/shipments/labels/<int:pk>',
    views.LabelsView.as_view(),
    name='shipment_labels',
)
```

Now visiting `/shipments/labels/17`, will return a PDF file which your browser will render. Note that, since we used the `LoginRequiredMixin`, anonymous users will be redirected to the usual login screen, and then back to this view after login.

CHAPTER 4

API

4.1 PDFView

4.2 Helpers

5.1 v4.0.0

- Dropped support for Python 3.6.
- Dropped support for Python 3.7.
- Supported Python versions are now 3.8, 3.9 and 3.10.
- Supported Django versions are now 3.2 and 4.0.

5.2 v3.0.1

- `render_pdf()` may take a list of templates or a single template. This restores compatibility with pre-v3.0.0 interface.

5.3 v3.0.0

- `get_template_name` has been deprecated in favour of `get_template_names`. This does not affect usages when `template_name` is defined.
- `get_download_name` has been deprecated. Override `download_name` as a property instead.

5.4 v2.2.0

- `django_renderpdf.views.PDFView.url_fetcher` is no longer a static method. If you were overriding this method, make sure you remove the `@staticmethod` decorator from your implementation.
- Improved documentation at RTD.

5.5 v2.1.0

- Add handling of relative URLs. CSS, image files, and other resources will be resolved using Django's internal URL routing. This includes scenarios like serving static or media files via Django, or serving things like custom CSS via custom Django views.
- Drop support for Python 3.5.

5.6 v2.0.1

- Improve handling of remote static files.

5.7 v2.0.0

- Support Python 3.7 and 3.8.
- Support Django 2.2, 3.0 and 3.1.
- Drop support for Django < 2.2.

CHAPTER 6

Help

If you think you found a bug, or need help with something specific:

- Please check existing issues for similar topics.
- If there's nothing relevant, please open a new issue describing your problem, and what you've tried so far.

Issues and source code are all in [GitHub](#).

CHAPTER 7

Background

django-renderpdf actually started out as code on multiple of my own projects (including some public ones).

After some time, it became clear that I'd been copy-pasting PDF-related bits across different projects, and since co-workers expressed interest in this design (using the Django templating system to generate PDFs), it finally made sense to move this into a separate library.

CHAPTER 8

Donations

Donations are welcome. See [here](#) for further details.

CHAPTER 9

Licence

django-renderpdf is licensed under the ISC licence.