

---

# **django-wkhtmltopdf Documentation**

*Release 3.2.0*

**Incuna Ltd**

**Jun 05, 2018**



---

## Contents

---

<b>1 Quickstart</b>	<b>3</b>
<b>2 Contribute</b>	<b>5</b>
<b>3 Contents</b>	<b>7</b>



django-wkhtmltopdf allows a Django site to output dynamic PDFs. It utilises the [wkhtmltopdf](#) library, allowing you to write using the technologies you know - HTML and CSS - and output a PDF file.



# CHAPTER 1

---

## Quickstart

---

```
pip install django-wkhtmltopdf
```

Grab the wkhtmltopdf [binary](#) for your platform.

settings.py

```
INSTALLED_APPS = (  
    # ...  
    'wkhtmltopdf',  
    # ...  
)
```

urls.py

```
from django.conf.urls.defaults import url, patterns  
from wkhtmltopdf.views import PDFTemplateView  
  
urlpatterns = patterns('',  
    url(r'^pdf/$', PDFTemplateView.as_view(template_name='my_template.html',  
                                          filename='my_pdf.pdf'), name='pdf'),  
)
```





## CHAPTER 2

---

Contribute

---

You can fork the project on [Github](#).



## 3.1 Full Installation Notes

### 3.1.1 Installing the Package

#### From PyPI

```
pip install django-wkhtmltopdf
```

#### From source

```
git clone git://github.com/incuna/django-wkhtmltopdf.git
cd django-wkhtmltopdf
python setup.py install
```

### 3.1.2 Installing the Binary

Find the relevant version of the `wkhtmltopdf` binary from the project [downloads page](#).

### 3.1.3 Setting up your Django

Add `wkhtmltopdf` to your `INSTALLED_APPS`:

```
INSTALLED_APPS = (
    # ...
    'wkhtmltopdf',
    # ...
)
```

By default it will try to execute the `wkhtmltopdf` command from your `PATH`.

If you can't add `wkhtmltopdf` to your `PATH` or you want to use some other version, you can use the `WKHTMLTOPDF_CMD` setting:

```
WKHTMLTOPDF_CMD = '/path/to/my/wkhtmltopdf'
```

### 3.1.4 Display static files

Set `STATIC_ROOT` in your `settings.py`:

```
STATIC_ROOT = '/full/path/to/static/directory/'
```

Make sure your static files and directories are inside this directory.

**Note:** In production static files are supposed to reside outside the project folder, in a public directory. The `STATIC_ROOT`-setting gives the path to this directory. However, `django-wkhtmltopdf` requires that `STATIC_ROOT` is also set on your local machine.

In development the static files reside in their respective apps folder or in a cross-app directory defined by the `STATIC_DIRS`-setting. Refer to the `django` documentation for how you can move static files to the `STATIC_ROOT` directory through a `django` script.

## 3.2 Usage

The `PDFTemplateView` is a Django class-based view. By default, it uses `PDFTemplateResponse` to render an HTML template to PDF. It accepts the following class attributes:

**template\_name** The full name of a template to use as the body of the PDF.

**header\_template** Optional. The full name of a template to use as the header on each page.

**footer\_template** Optional. The full name of a template to use as the footer on each page.

**filename** The filename to use when responding with an attachment containing the PDF. Default is `'rendered_pdf.pdf'`.

If `None`, the view returns the PDF output inline, not as an attachment.

**response\_class** The response class to be returned by `render_to_response()` method. Default is `PDFTemplateResponse`.

**html\_response\_class** The response class to be returned by `render_to_response()` method, when rendering as HTML. See note below. Default is `TemplateResponse`.

**cmd\_options** The dictionary of command-line arguments passed to the underlying `wkhtmltopdf` binary. Default is `{}`.

`wkhtmltopdf` options can be found by running `wkhtmltopdf --help`. Unfortunately they don't provide hosted documentation.

---

**Note:** For convenience in development you can add the GET arg `?as=html` to the end of your URL to render the PDF as a web page.

---

### 3.2.1 Simple Example

Point a URL at PDFTemplateView:

```
from django.conf.urls.defaults import *
from wkhtmltopdf.views import PDFTemplateView

urlpatterns = patterns('',
    # ...
    url(r'^pdf/$', PDFTemplateView.as_view(template_name='my_template.html',
                                          filename='my_pdf.pdf'), name='pdf'),
    # ...
)
```

### 3.2.2 Advanced Example

Point a URL (as above) at your own view that subclasses PDFTemplateView and override the sections you need to.

```
from wkhtmltopdf.views import PDFTemplateView

class MyPDF(PDFTemplateView):
    filename = 'my_pdf.pdf'
    template_name = 'my_template.html'
    cmd_options = {
        'margin-top': 3,
    }
```

### 3.2.3 Unicode characters

Templates containing utf-8 characters should be supported. You will need to ensure that you set the content type in your template file for *wkhtmltopdf* to interpret it properly.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

## 3.3 Settings

### 3.3.1 Available settings

Here's a full list of available settings, in alphabetical order, and their default values.

#### WKHTMLTOPDF\_CMD

Default: 'wkhtmltopdf'

The name of the wkhtmltopdf binary.

If there are no path components, this app will look for the binary using the default OS paths.

## WKHTMLTOPDF\_CMD\_OPTIONS

Default: `{'encoding': 'utf8', 'quiet': True}`

A dictionary of command-line arguments to pass to the `wkhtmltopdf` binary. Keys are the name of the flag and values are arguments for the flag.

To pass a simple flag, for example: `wkhtmltopdf --disable-javascript:`

```
WKHTMLTOPDF_CMD_OPTIONS = {'disable-javascript': True}
```

To pass a flag with an argument, for example: `wkhtmltopdf --title 'TPS Report':`

```
WKHTMLTOPDF_CMD_OPTIONS = {'title': 'TPS Report'}
```

## WKHTMLTOPDF\_DEBUG

Default: same as `settings.DEBUG`

A boolean that turns on/off debug mode.

## WKHTMLTOPDF\_ENV

Default: None

An optional dictionary of environment variables to override, when running the `wkhtmltopdf` binary. Keys are the name of the environment variable.

A common use of this is to set the `DISPLAY` environment variable to another X server, when using `wkhtmltopdf --use-xserver:`

```
WKHTMLTOPDF_ENV = {'DISPLAY': ':2'}
```