

PACKAGE INDEX >>

[Browse packages](#)
[List trove classifiers](#)
[RSS \(latest 40 updates\)](#)
[RSS \(newest 40 packages\)](#)
[Terms of Service](#)
[PyPI Tutorial](#)
[PyPI Security](#)
[PyPI Support](#)
[PyPI Bug Reports](#)
[PyPI Discussion](#)
[PyPI Developer Info](#)

[ABOUT](#) >>[NEWS](#) >>[DOCUMENTATION](#) >>[DOWNLOAD](#) >>[COMMUNITY](#) >>[FOUNDATION](#) >>[CORE DEVELOPMENT](#) >>

pdfkit 0.6.1

Wkhtmltopdf python wrapper to convert html to pdf using the webkit rendering engine and qt

Downloads ↓

build passing pypi package 0.6.1

Python 2 and 3 wrapper for wkhtmltopdf utility to convert HTML to PDF using Webkit.

This is adapted version of [ruby PDFKit](#) library, so big thanks to them!

Installation

1. Install python-pdfkit:

```
$ pip install pdfkit
```

2. Install wkhtmltopdf:

- Debian/Ubuntu:

```
$ sudo apt-get install wkhtmltopdf
```

Warning! Version in debian/ubuntu repos have reduced functionality (because it compiled without the wkhtmltopdf QT patches), such as adding outlines, headers, footers, TOC etc. To use this options you should install static binary from [wkhtmltopdf](#) site or you can use [this script](#).

- Windows and other options: check wkhtmltopdf [homepage](#) for binary installers

Usage

For simple tasks:

```
import pdfkit

pdfkit.from_url('http://google.com', 'out.pdf')
pdfkit.from_file('test.html', 'out.pdf')
pdfkit.from_string('Hello!', 'out.pdf')
```

You can pass a list with multiple URLs or files:

```
pdfkit.from_url(['google.com', 'yandex.ru', 'engadget.com'], 'out.pdf')
pdfkit.from_file(['file1.html', 'file2.html'], 'out.pdf')
```

Also you can pass an opened file:



```
with open('file.html') as f:
    pdfkit.from_file(f, 'out.pdf')
```

If you wish to further process generated PDF, you can read it to a variable:

```
# Use False instead of output path to save pdf to a variable
pdf = pdfkit.from_url('http://google.com', False)
```

You can specify all wkhtmltopdf [options](#). You can drop '-' in option name. If option without value, use *None*, *False* or '' for dict value:. For repeatable options (incl. allow, cookie, custom-

Not Logged In

[Login](#)
[Register](#)
[Lost Login?](#)
[Login with OpenID](#) 
[Login with Google](#) 

Status

[Nothing to report](#)

header, post, postfile, run-script, replace) you may use a list or a tuple. With option that need multiple values (e.g. `--custom-header Authorization secret`) we may use a 2-tuple (see example below).

```
options = {
    'page-size': 'Letter',
    'margin-top': '0.75in',
    'margin-right': '0.75in',
    'margin-bottom': '0.75in',
    'margin-left': '0.75in',
    'encoding': "UTF-8",
    'custom-header' : [
        ('Accept-Encoding', 'gzip')
    ]
    'cookie': [
        ('cookie-name1', 'cookie-value1'),
        ('cookie-name2', 'cookie-value2'),
    ],
    'no-outline': None
}

pdfkit.from_url('http://google.com', 'out.pdf', options=options)
```

By default, PDFKit will show all wkhtmltopdf output. If you dont want it, you need to pass `quiet` option:

```
options = {
    'quiet': ''
}

pdfkit.from_url('google.com', 'out.pdf', options=options)
```

Due to wkhtmltopdf command syntax, **TOC** and **Cover** options must be specified separately. If you need cover before TOC, use `cover_first` option:

```
toc = {
    'xsl-style-sheet': 'toc.xsl'
}

cover = 'cover.html'

pdfkit.from_file('file.html', options=options, toc=toc, cover=cover)
pdfkit.from_file('file.html', options=options, toc=toc, cover=cover, cover_first=True)
```

You can specify external CSS files when converting files or strings using `css` option.

Warning This is a workaround for [this bug](#) in wkhtmltopdf. You should try `--user-style-sheet` option first.

```
# Single CSS file
css = 'example.css'
pdfkit.from_file('file.html', options=options, css=css)

# Multiple CSS files
css = ['example.css', 'example2.css']
pdfkit.from_file('file.html', options=options, css=css)
```

You can also pass any options through meta tags in your HTML:

```
body = """
<html>
  <head>
    <meta name="pdfkit-page-size" content="Legal"/>
    <meta name="pdfkit-orientation" content="Landscape"/>
  </head>
  Hello World!
</html>
"""

pdfkit.from_string(body, 'out.pdf') #with --page-size=Legal and --orientation=Landscape
```

Configuration

Each API call takes an optional configuration parameter. This should be an instance of `pdfkit.configuration()` API call. It takes the configuration options as initial parameters. The available options are:

- `wkhtmltopdf` - the location of the `wkhtmltopdf` binary. By default `pdfkit` will attempt to locate this using `which` (on UNIX type systems) or `where` (on Windows).
- `meta_tag_prefix` - the prefix for `pdfkit` specific meta tags - by default this is `pdfkit-`

Example - for when `wkhtmltopdf` is not on `$PATH`:

```
config = pdfkit.configuration(wkhtmltopdf='/opt/bin/wkhtmltopdf')
pdfkit.from_string(html_string, output_file, configuration=config)
```

Troubleshooting

- `IOError: 'No wkhtmltopdf executable found':`

Make sure that you have `wkhtmltopdf` in your `$PATH` or set via custom configuration (see preceding section). *where* `wkhtmltopdf` in Windows or *which* `wkhtmltopdf` on Linux should return actual path to binary.

- `IOError: 'Command Failed'`

This error means that PDFKit was unable to process an input. You can try to directly run a command from error message and see what error caused failure (on some `wkhtmltopdf` versions this can be caused by segmentation faults)

Changelog

- 0.6.1
 - Fix regression on python 3+ when trying to decode pdf output
- 0.6.0
 - Support repeatable options
 - Support multiple values for some options
 - Fix some corner cases when specific argument order is required
 - Some Python 3+ compatibility fixes
 - Update README
- 0.5.0
 - Allow passing multiple css files
 - Fix problems with external file encodings
 - Raise an error when X server is missing on *nix systems
 - Fix tests that was broken with latest `wkhtmltopdf` release
 - Update README
- 0.4.1
 - More easier custom configuration setting
 - Update README
- 0.4.0
 - Allow passing file-like objects
 - Ability to return PDF as a string
 - Allow user specification of configuration
 - API calls now returns True on success
 - bugfixes
- 0.3.0
 - Python 3 support
- 0.2.4
 - Add History
 - Update setup.py

- 0.2.3
 - Fix installing with setup.py
 - Update README

File	Type	Py Version	Uploaded on	Size
pdfkit-0.6.1-py2-none-any.whl (md5)	Python Wheel	2.7	2017-01-09	12KB
pdfkit-0.6.1-py2.7.egg (md5)	Python Egg	2.7	2017-01-09	15KB
pdfkit-0.6.1-py3-none-any.whl (md5)	Python Wheel	3.5	2017-01-09	12KB
pdfkit-0.6.1.tar.gz (md5)	Source		2017-01-09	9KB

Author: Golovanov Stanislav

Download URL: <https://github.com/JazzCore/python-pdfkit>

License: MIT

Categories

[Programming Language :: Python](#)
[Programming Language :: Python :: 2.7](#)
[Programming Language :: Python :: 3.2](#)
[Programming Language :: Python :: 3.3](#)
[Topic :: Text Processing](#)
[Topic :: Text Processing :: General](#)
[Topic :: Text Processing :: Markup](#)
[Topic :: Text Processing :: Markup :: HTML](#)
[Topic :: Text Processing :: Markup :: XML](#)
[Topic :: Utilities](#)

Package Index Owner: JazzCore

DOAP record: [pdfkit-0.6.1.xml](#)

Copyright © 1990-2018, Python Software Foundation
 Terms of Use



Website maintained by the Python community
 Real-time CDN by Fastly / Hosting by Rackspace
 Object storage by Amazon S3 / Design by Tim Parkin