pdfkit 0.6.1

```
pip install pdfkit
```

✓ Latest version

Released: Jan 9, 2017

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

Navigation

- Project description
- Release history
- Download files

Project links

- Download

Statistics

GitHub statistics:

- **Stars:** 1335
- **Forks:** 252
- **Open issues/PRs:** 101

View statistics for this project via Libraries.io, or by using our public dataset on Google BigQuery

Project description

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

Meta

License: MIT

Author: Golovanov Stanislav

Maintainers

JazzCore

Classifiers

Programming Language

- Python
- Python :: 2.7
- Python :: 3.2
- Python :: 3.3

Topic

- Text Processing
- Text Processing :: General
- Text Processing :: Markup
- Text Processing :: Markup :: HTML
- Text Processing :: Markup :: XML
- Utilities

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-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_fi
```

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=
```

Configuration

Each API call takes an optional configuration paramater. This should be an instance of `pdfkit.configuration()` API call. It takes the configuration options as initial paramaters. 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 cause 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
  - Rise 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

Help

Installing packages

Uploading packages

User guide

FAQs

About PyPI

PyPI on Twitter

Infrastructure dashboard

Package index name retention

Our sponsors

Contributing to PyPI

Bugs and feedback

Contribute on GitHub

Translate PyPI

Development credits

Using PyPI

Code of conduct

Report security issue

Privacy policy

Terms of use

Status: all systems operational

Developed and maintained by the Python community, for the Python community.
Donate today!

Pingdom
Monitoring

Google
Object Storage and
Download Analytics

Sentry
Error logging

AWS
Cloud computing

DataDog
Monitoring

Fastly
CDN

DigiCert
EV certificate

StatusPage
Status page