

---

**django***pageexporter* Documentation  
***Release 0.3.dev20170409134009***

**Python Competence Centre**

**Apr 09, 2017**



---

## Contents

---

<b>1</b>	<b>Developer Documentation</b>	<b>3</b>
	<b>Python Module Index</b>	<b>7</b>



Welcome to Django Page Exporter Documentation.



## Installation

### Install the Package

```
pip install page_exporter
```

## Configuration

### How to use

The followings SETTINGS should contain values as follow:

```
INSTALLED_APPS = (  
    ...  
    'page_exporter',  
    ...  
)
```

url.py should contain:

```
urlpatterns = patterns(  
    ...  
    (r'^capture/', include('page_exporter.urls')),  
    ...  
)
```

you can override default configuration using:

```
PAGE_EXPORTER_CAPTURE_SCRIPT = './capture.js'
PAGE_EXPORTER_CLI_ARGS = []
PAGE_EXPORTER_PHANTOMJS_CMD = None
PAGE_EXPORTER_WAIT = '2000'
```

## Parameters

You can then obtain a screenshot using the following GET or POST parameters :

### **url**

The website URL to capture. This can be a fully qualified URL, or the name of a URL to be reversed in your Django project. Note: do not forget to encode the url.

### **selector**

CSS3 selector. It will restrict the screenshot to the selected element.

### **method**

HTTP method to be used (default: GET)

### **width**

Viewport width (default: 1400)

### **height**

Viewport height (default: 900)

### **data**

HTTP data to be posted (default: {})

### **wait**

milliseconds used as timeout before capture

### **page\_status**

string. The screenshot will be performed only once document.page\_status is equal to the passed value. Typical usage: if your page contains a heavy javascript processing, you can add the page\_status variable at the end of the processing to make sure the screenshot will get the page properly rendered.

### **render**

png (default), pdf (will use media print css), jpg (PIL needed)

### **cookie\_name, cookie\_value, cookie\_domain**

cookie information to send for authenticated pages



## page\_exporter package

### Submodules

#### page\_exporter.config module

**class** `page_exporter.config.Settings` (*prefix*)  
 Bases: `object`

#### page\_exporter.urls module

#### page\_exporter.utils module

**exception** `page_exporter.utils.CaptureError`  
 Bases: `exceptions.Exception`

**exception** `page_exporter.utils.UnsupportedImageFormat`  
 Bases: `exceptions.Exception`

`page_exporter.utils.image_mimetype` (*render*)

`page_exporter.utils.image_postprocess` (*imagefile, output, size, crop, render*)  
 Resize and crop captured image, and saves to output. (can be stream or filename)

`page_exporter.utils.page_capture` (*stream, url, method=None, width=None, height=None, selector=None, data=None, waitfor=None, size=None, crop=None, render='png', wait=None, cookie\_name=None, cookie\_value=None, cookie\_domain=None, page\_status=None*)  
 Captures web pages using phantomjs

`page_exporter.utils.parse_render` (*render*)

`page_exporter.utils.parse_size` (*size\_raw*)

`page_exporter.utils.parse_url` (*request, url*)  
 Parse url URL parameter.

`page_exporter.utils.phantomjs_command` ()

`page_exporter.utils.phantomjs_command_kwargs` ()  
 will construct kwargs for cmd

`page_exporter.utils.process_phantomjs_stdout` (*stdout*)  
 Parse and digest capture script output.

#### page\_exporter.views module

`page_exporter.views.capture` (*request*)

### Module contents

`page_exporter.get_git_changeset` ()  
 Returns a numeric identifier of the latest git changeset.

The result is the UTC timestamp of the changeset in YYYYMMDDHHMMSS format. This value isn't guaranteed to be unique, but collisions are very unlikely, so it's sufficient for generating the development version numbers.

`page_exporter.get_version()`

Derives a PEP386-compliant version number from VERSION.

### p

- `page_exporter`, 5
- `page_exporter.config`, 5
- `page_exporter.urls`, 5
- `page_exporter.utils`, 5
- `page_exporter.views`, 5



### C

[capture\(\)](#) (in module `page_exporter.views`), 5  
[CaptureError](#), 5

### G

[get\\_git\\_changeset\(\)](#) (in module `page_exporter`), 5  
[get\\_version\(\)](#) (in module `page_exporter`), 6

### I

[image\\_mimetype\(\)](#) (in module `page_exporter.utils`), 5  
[image\\_postprocess\(\)](#) (in module `page_exporter.utils`), 5

### P

[page\\_capture\(\)](#) (in module `page_exporter.utils`), 5  
[page\\_exporter](#) (module), 5  
[page\\_exporter.config](#) (module), 5  
[page\\_exporter.urls](#) (module), 5  
[page\\_exporter.utils](#) (module), 5  
[page\\_exporter.views](#) (module), 5  
[parse\\_render\(\)](#) (in module `page_exporter.utils`), 5  
[parse\\_size\(\)](#) (in module `page_exporter.utils`), 5  
[parse\\_url\(\)](#) (in module `page_exporter.utils`), 5  
[phantomjs\\_command\(\)](#) (in module `page_exporter.utils`), 5  
[phantomjs\\_command\\_kwargs\(\)](#) (in module `page_exporter.utils`), 5  
[process\\_phantomjs\\_stdout\(\)](#) (in module `page_exporter.utils`), 5

### S

[Settings](#) (class in `page_exporter.config`), 5

### U

[UnsupportedImageFormat](#), 5