
Django Decorator Plus Documentation

Release 0.1

Andrew Pinkham

September 03, 2015

1	Installing Django Decorator Plus	3
2	View Decorator Usage	5
2.1	Controlling HTTP Methods in Function Views	5
3	Django Decorator API	7
3.1	View Decorator API	7
4	Indices and tables	9
	Python Module Index	11

Contents:

Installing Django Decorator Plus

Releases of the project are stored on [PyPI](#), and may be installed with `pip`.

```
$ pip install django-decorator-plus
```

To install the development version straight from [Github](#), it is possible to issue the following command.

```
$ pip install git+https://github.com/jambronrose/django-decorator-plus.git
```

View Decorator Usage

The package currently supplies decorators to improve your views.

- *Controlling HTTP Methods in Function Views*

2.1 Controlling HTTP Methods in Function Views

The view decorators provided are meant to restrict the HTTP methods allowed on a view. At the base of all of these decorators is the `require_http_methods()` decorator, which is an enhanced version of the `decorator` supplied by Django by the same name.

```
from decorator_plus import require_http_methods

@require_http_methods(["GET", "POST"])
def function_view(request):
    # The site will respond with an HTTP 405 error code if
    # the HTTP method is not: GET, HEAD, POST, OPTIONS
```

The `require_http_methods()` (in this package, not Django's) automatically supplies the `OPTIONS` HTTP method, and will automatically add the `HEAD` HTTP method if the `GET` method is allowed.

The package also supplies two shortcut decorators for your most common tasks:

- `require_safe_methods` limits views to `GET` and `HEAD` (and `OPTIONS`),
- `require_form_methods` limits views to `GET`, `HEAD`, and `POST` (and `OPTIONS`)

```
from decorator_plus import (
    require_form_methods,
    require_safe_methods,
)

@require_form_methods
def function_view_form(request):
    # equivalent to :
    # @require_http_methods(["GET", "HEAD", "POST"])

@require_safe_methods
def function_view_safe(request):
    # equivalent to :
    # @require_http_methods(["GET", "HEAD"])
```

Django Decorator API

- *View Decorator API*

3.1 View Decorator API

`decorator_plus.view_decorators.require_form_methods(func)`

Decorator to require that a function view only accept GET, HEAD and POST methods.

`decorator_plus.view_decorators.require_http_methods(request_methods)`

Decorator to make a function view only accept particular request methods. Usage:

```
@require_http_methods(["GET", "POST"])
def function_view(request):
    # HTTP methods != GET or POST results in 405 error code response
```

`decorator_plus.view_decorators.require_safe_methods(func)`

Decorator to require that a function view only accept safe methods: GET and HEAD.

Indices and tables

- `genindex`
- `modindex`
- `search`

d

`decorator_plus.view_decorators`, [7](#)

D

`decorator_plus.view_decorators` (module), [7](#)

R

`require_form_methods()` (in module `decorator_plus.view_decorators`), [7](#)

`require_http_methods()` (in module `decorator_plus.view_decorators`), [7](#)

`require_safe_methods()` (in module `decorator_plus.view_decorators`), [7](#)