
Flask-Environments Documentation

Release 0.1.0

Matt Wright

August 04, 2016

1	Resources	3
2	Contents	5
3	Installation	7
4	Getting Started	9
	Python Module Index	11

Flask-Environments provides environment tools and configuration for Flask applications

Resources

- Documentation
- Issue Tracker
- Code
- Development Version

Contents

- *Installation*
- *Getting Started*

Installation

```
$ pip install flask-environments
```

Getting Started

The following code illustrates how to setup environment based configuration. The first thing to do is create a configuration file. These can be created using Python or Yaml.

Python:

```
class Config(object):
    DEBUG = False
    TESTING = False

class Development(Config):
    DEBUG = True
    DATABASE = 'development_db'

class Production(Config):
    DATABASE = 'production_db'
```

Yaml:

```
COMMON: &common
  DEBUG: False
  TESTING: False

DEVELOPMENT: &development
  <<: *common
  DEBUG: True
  DATABASE: 'development_db'

PRODUCTION: &production
  <<: *common
  DATABASE: 'production_db'
```

Next, create your application and initialize the Environments extensions:

```
from flask import Flask
from flask_environments import Environments

app = Flask(__name__)
env = Environments(self.app)
```

Then simply use the `from_object` method or the `from_yaml` method to load the configuration:

```
env.from_object('myapp.config')
env.from_yaml(os.path.join(os.getcwd(), 'myapp', 'config.yml'))
```

Only the values for the specified environment will be applied.

Flask-Environments assumes an operating system environment variable named *FLASK_ENV* will be set to one of your possible environments. If it is not set, it will default to *DEVELOPMENT*.

To change the default environment or the environment variable name pass the *var_name* or *default_env* parameters to the Environments constructor like so:

```
from flask import Flask
from flask_environments import Environments

app = Flask(__name__)
env = Environments(self.app, var_name='CUSTOM_VAR_NAME', default_env='CUSTOM_ENV')
```

f

flask_environments, 3

F

`flask_environments` (module), [1](#)