

---

# FlaskBB Documentation

*Release 0.1-dev*

**sh4nks**

February 15, 2014







FlaskBB is a lightweight forum software in Flask.



## 1.1 Installation

- Basic Setup
- Configuration
- Deploying

## 1.2 Basic Setup

### 1.2.1 Virtualenv Setup

Before you can start, you need to create a *virtualenv*. You can install the *virtualenvwrapper* with your package manager or via *pip*. Be sure that *pip* is installed. If you don't know how to install *pip*, have a look at their [documentation](#).

For example, on archlinux you can install it with

```
$ sudo pacman -S python2-virtualenvwrapper
```

or, if you own a Mac, you can simply install it with

```
$ sudo pip install virtualenvwrapper
```

For more information checkout the [virtualenvwrapper](#) installation.

After that you can create your *virtualenv* with

```
$ mkvirtualenv -a /path/to/flaskbb -p $(which python2) flaskbb
```

and you should be switched automatically to your newly created *virtualenv*. To deactivate it you just have to type `deactivate` and if you want to work on it again, you need to type `workon flaskbb`.

### 1.2.2 Required Dependencies

Now you can install the required dependencies.

```
$ pip install -r requirements.txt
```

### 1.2.3 Optional Dependencies

We have one optional dependency, redis (the python package is installed automatically). If you want to use it, be sure that a redis-server is running. If you decide to use redis, the *online guests* and *online users* are being tracked by redis, else it will only track the *online users* via a simple SQL query.

#### On Archlinux

```
# Install redis
$ sudo pacman -S redis

# Check if redis is already running.
$ systemctl status redis

# If not, start it.
$ sudo systemctl start redis

# Optional: Start redis everytime you boot your machine
$ sudo systemctl enable redis
```

#### On Debian 7.0 (Wheezy)

```
# Install redis
$ sudo apt-get install redis-server

# Check if redis is already running.
$ service redis-server status

# If not, start it
$ sudo service redis-server start

# Optional: Start redis everytime you boot your machine
# I can't remember if this is done automatically..
$ sudo update-rc.d redis-server defaults
```

## 1.3 Configuration

Before you can start, you need to configure *FlaskBB*.

### 1.3.1 Development

For development, you need to copy `flaskbb/configs/development.py.example` to `flaskbb/configs/development.py`.

```
cp flaskbb/configs/development.py.example flaskbb/configs/development.py
```

The reCAPTCHA keys should work fine on localhost. If you don't want to use Google Mail, see Mail Examples for more options.



### 1.3.2 Production

If you plan, to use *FlaskBB* in a production environment (not recommended at the moment, because it's still in development), you need to copy `flaskbb/configs/production.py.example` to `flaskbb/configs/production.py`.

```
cp flaskbb/configs/production.py.example flaskbb/configs/production.py
```

Now open `flaskbb/configs/production.py` with your favourite editor and adjust the config variables to your needs. If you don't want to use Google Mail, see Mail Examples for more options.

### 1.3.3 Mail Examples

#### Google Mail

```
MAIL_SERVER = "smtp.gmail.com"
MAIL_PORT = 465
MAIL_USE_SSL = True
MAIL_USERNAME = "your_username@gmail.com"
MAIL_PASSWORD = "your_password"
MAIL_DEFAULT_SENDER = ("Your Name", "your_username@gmail.com")
# Where to logger should send the emails to
ADMINS = ["your_admin_user@gmail.com"]
```

#### Local SMTP Server

```
MAIL_SERVER = "localhost"
MAIL_PORT = 25
MAIL_USE_SSL = False
MAIL_USERNAME = ""
MAIL_PASSWORD = ""
MAIL_DEFAULT_SENDER = "noreply@example.org"
# Where to logger should send the emails to
ADMINS = ["your_admin_user@example.org"]
```

## 1.4 Installation

Now, you should be able to install *FlaskBB* and can run therefore

```
python manage.py initflaskbb
```

Here you are asked about what your username is, which email adress you use and last but not least, which password your admin user has (please choose a secure one).

To test if everything worked, run the development server with `python manage.py runserver`.

## 1.5 Deploying

### 1.5.1 Supervisor

### 1.5.2 uWSGI

### 1.5.3 nginx