

---

# Flask-Split Documentation

*Release 0.1.0*

**Janne Vanhala**

March 11, 2012



# CONTENTS



Flask-Split is a Flask extension for [A/B testing](#) your web application. It is a port of Andrew Nesbitt's excellent [Split](#) A/B testing framework to Python and Flask.



# INSTALLATION

The easiest way to install Flask-Split is with pip:

```
pip install Flask-Split
```

You will also need Redis as Flask-Split uses it as a datastore. Flask-Split only supports Redis 2.0 or greater.

In case you are on OS X, the easiest way to install Redis is with Homebrew:

```
brew install redis
```

If you are on Ubuntu or other Debian-based Linux, you can install Redis with APT:

```
sudo apt-get install redis-server
```





# QUICKSTART

In order to start using Flask-Split, you need to first need to register the Flask-Split blueprint to your Flask application:

```
from flask import Flask
from flask.ext.split import split

app = Flask(__name__)
app.register_blueprint(split)
```

After that you can start A/B testing your application.

## 2.1 Defining an A/B test

You can define experiments with the `ab_test()` function in a view or a template. For example, in a template you can define an experiment like so:

```
<button type="submit">
  {{ ab_test('signup_btn_text', 'Register', 'Sign up') }}
</button>
```

This will set up a new experiment called *signup\_btn\_text* with two alternatives: *Register* and *Sign up*. The first alternative is the control. It should be the original text that was already on the page and the text you test new alternative against. You should not add only new alternatives as then you won't be able to tell if you have improved over the original or not.

## 2.2 Tracking conversions

To measure how the alternative has impacted the conversion rate of your experiment you need to mark a visitor reaching the conversion point. You can do this with the `finished()` function:

```
finished('signup_btn_text')
```

You should place this in a view, for example after a user has completed the sign up process.



# CONFIGURATION

The following configuration values exist for Flask-Split. Flask-Split loads these values from your main Flask config which can be populated in various ways.

A list of configuration keys currently understood by the extension:

SPLIT_ALLOW_MULTIPLE_EXPERIMENTS	<p>If set to <i>True</i> Flask-Split will allow users to participate in multiple experiments.</p> <p>If set to <i>False</i> Flask-Split will avoid users participating in multiple experiments at once. This means you are less likely to skew results by adding in more variation to your tests.</p> <p>Defaults to <i>False</i>.</p>
SPLIT_IGNORE_IP_ADDRESSES	<p>Specifies a list of IP addresses to ignore visits from. You may wish to use this to prevent yourself or people from your office from skewing the results.</p> <p>Defaults to <code>[]</code>, i.e. no IP addresses are ignored by default.</p>
SPLIT_ROBOT_REGEX	<p>Flask-Split ignores visitors that appear to be robots or spider in order to avoid them from skewing any results. Flask-Split detects robots and spiders by comparing the user agent of each request with the regular expression in this setting.</p> <p>Defaults to:</p> <pre>r""" (?:i)\b(     Baidu/     Gigabot/     Googlebot/     libwww-perl/     lwp-trivial/     msnbot/     SiteUptime/     Slurp/     WordPress/     ZIBB/     ZyBorg )\b """</pre>
SPLIT_DB_FAILOVER	<p>If set to <i>True</i> Flask-Split will not let <code>ab_test()</code> or <code>finished()</code> to crash in case of a Redis connection error. In that case <code>ab_test()</code> always delivers the first alternative i.e. the control.</p> <p>Defaults to <i>True</i>.</p>

# WEB INTERFACE

Flask-Split comes with a web frontend to get an overview of how your experiments are doing. You can find the web interface from the address `/split/`.

If you would like to restrict the access to the web interface, you can take advantage of blueprint's hooks:

```
from flask import abort
from flask.ext.split import split

@split.before_request
def require_login():
    if not user_logged_in():
        abort(401)
```



# API REFERENCE

This part of the documentation covers all the public classes and functions in Flask-Split.

`flask.ext.split.ab_test(experiment_name, *alternatives)`

Start a new A/B test.

Returns one of the alternatives. If the user has already seen the test, they will get the same alternative as before.

## Parameters

- **experiment\_name** – Name of the experiment. You should never use the same experiment name to refer to a second experiment.
- **alternatives** – A list of alternatives. Each item can be either a string or a two-tuple of the form (alternative name, weight). By default each alternative has the weight of 1. The first alternative is the control. Every experiment must have at least two alternatives.

`flask.ext.split.finished(experiment_name, reset=True)`

Track a conversion.

## Parameters

- **experiment\_name** – Name of the experiment.
- **reset** – If set to *True* current user's session is reset so that they may start the test again in the future. If set to *False* the user will always see the alternative they started with. Defaults to *True*.





# CHANGELOG

Here you can see the full list of changes between each Flask-Split release.

## 6.1 0.1.0 (2012-03-11)

- Initial public release



# LICENSE

Copyright (c) 2012 Andrew Nesbitt

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



# PYTHON MODULE INDEX

f

`flask.ext.split, ??`