gunstar Documentation

Release 0.1-dev

Allisson Azevedo

August 24, 2013

CONTENTS

Gunstar is a WSGI micro web framework.

Github Repo.

CHAPTER

ONE

FEATURES

- Based on WebOb.
- Simple url routing.
- Jinja2 templates.
- Session interface with signed cookies.
- Signals with blinker.
- Unit testing support.
- Supports Python 2.6, 2.7, 3.3 and PyPy.

CHAPTER

TWO

EXAMPLE

Hello World App:

```
# -*- coding: utf-8 -*-
from gunstar.app import Application
from gunstar.http import RequestHandler

class IndexHandler(RequestHandler):
    def get(self):
        self.response.write(’Hello World’)


routes = (
    (’/’, IndexHandler, ’index_named_url’),
)


app = Application(routes=routes)


if __name__ == ’__main__’:
    from wsgiref.simple_server import make_server
    server = make_server(’127.0.0.1’, 8080, app)
    server.serve_forever()
```

CHAPTER

THREE

CONTENTS

3.1 Installation

Install the latest stable release via PyPI:

```
pip install gunstar
```

Gunstar runs with Python 2.6, 2.7, 3.3 and PyPy .

3.2 Quickstart

Let's start, create a new project structure for myapp:

```
mkdir myapp
cd myapp
mkdir static # for static files
mkdir templates # for jinja2 templates
touch app.py # main app
touch handlers.py # handlers classes
touch tests.py # for testing the app
```

Now, edit the content of app.py file:

```
# -*- coding: utf-8 -*-
from gunstar.app import Application


myapp = Application()


if __name__ == '__main__':
    from wsgiref.simple_server import make_server
    server = make_server('127.0.0.1', 8080, myapp)
    server.serve_forever()
```

Running a development server:

```
python app.py
```

Go to your browser and visit http://127.0.0.1:8080, you see a 404 page.

It's time to create your first request handler class, go to handlers.py and edit:

```
# -*- coding: utf-8 -*-
from gunstar.http import RequestHandler


class IndexHandler(RequestHandler):

    def get(self):
        self.response.write('Index Handler')
```

Go back to app.py and create routes tuple:

```
# -*- coding: utf-8 -*-
from gunstar.app import Application


routes = (
    ('/', 'handlers.IndexHandler', 'index'),
)


myapp = Application(routes=routes)


if __name__ == '__main__':
    from wsgiref.simple_server import make_server
    server = make_server('127.0.0.1', 8080, myapp)
    server.serve_forever()
```

Restart your development server and visit http://127.0.0.1:8080.

The tuple maps a url(‘/’) to a handler(‘handlers.IndexHandler’) and have a name (‘index’).

Congratulations, your first app is working now!

3.2.1 Working with templates

We need to set up a TEMPLATE_PATH variable in config:

```
# -*- coding: utf-8 -*-
from gunstar.app import Application
import os


PROJECT_PATH = os.path.dirname(os.path.abspath(__file__))


class ConfigSettings(object):

    TEMPLATE_PATH = os.path.join(PROJECT_PATH, 'templates')


routes = (
    ('/', 'handlers.IndexHandler', 'index'),
)


myapp = Application(routes=routes, config=ConfigSettings)
```

```
if __name__ == '__main__':
    from wsgiref.simple_server import make_server
    server = make_server('127.0.0.1', 8080, myapp)
    server.serve_forever()
```

Create file templates/index.html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>MyApp - Index</title>
    <link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css">
  </head>

  <body>

    <div class="container">
      <h1>Hello Stranger!</h1>
    </div>

  </body>
</html>
```

Edit handlers.py to use render_template:

```
# -*- coding: utf-8 -*-
from gunstar.http import RequestHandler


class IndexHandler(RequestHandler):

    def get(self):
        self.render_template('index.html')
```

Restart your development server and visit http://127.0.0.1:8080.

3.2.2 Serving static files

Set STATIC_PATH and STATIC_ROOT in config:

```
class ConfigSettings(object):

    TEMPLATE_PATH = os.path.join(PROJECT_PATH, 'templates')
    STATIC_ROOT = os.path.join(PROJECT_PATH, 'static')
    STATIC_PATH = '/static/'
```

Create static/index.html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>MyApp - Index</title>
    <link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css">
  </head>
  <body>

    <div class="container">
      Index from static files.
    </div>

  </body>
</html>
```

Restart your development server and visit http://127.0.0.1:8080/static/index.html.

3.2.3 Working with session

The session is available in RequestHandler.session if you set SECRET_KEY in config:

```
class ConfigSettings(object):

    TEMPLATE_PATH = os.path.join(PROJECT_PATH, 'templates')
    STATIC_ROOT = os.path.join(PROJECT_PATH, 'static')
    STATIC_PATH = '/static/'
    SECRET_KEY = 'my-secret-key'
```

Edit handlers.py:

```
# -*- coding: utf-8 -*-
from gunstar.http import RequestHandler


class IndexHandler(RequestHandler):

    def get(self):
        view_count = self.session.get('view_count', 0)
        view_count += 1
        self.session.set('view_count', view_count)
        self.session.save()
        self.render_template('index.html', view_count=view_count)
```

Edit templates/index.html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>MyApp - Index</title>
    <link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css">
  </head>

  <body>

    <div class="container">
      <h1>Hello Stranger!</h1>
      <h2>It's your {{ view_count }} visit to this page</h2>
    </div>

  </body>
</html>
```

Restart your development server and reload page to see view_count increment.

3.2.4 Testing

Do a favor to yourself and use nose to run the tests:

```
pip install nose
```

Gunstar has a TestCase with a nice test client. You have to override get_app method and return your app, that's it.

Edit tests.py:

```
# -*- coding: utf-8 -*-
from gunstar.testing import TestCase
from app import myapp


class AppTestCase(TestCase):

    def get_app(self):
        return myapp

    def test_index_handler(self):
        resp = self.client.get('/')
        self.assertEqual(resp.status_code, 200)
        self.assertTrue('<h1>Hello Stranger!</h1>' in resp.text)
        self.assertTrue('1 visit to this page' in resp.text)
        self.assertEqual(resp.context['view_count'], 1)

        resp = self.client.get('/')
        self.assertEqual(resp.status_code, 200)
        self.assertEqual(resp.context['view_count'], 2)

        resp = self.client.get('/')
        self.assertEqual(resp.status_code, 200)
        self.assertEqual(resp.context['view_count'], 3)

    def test_static_file(self):
        resp = self.client.get('/static/index.html')
        self.assertEqual(resp.status_code, 200)
        self.assertTrue('<h1>Index from static files.</h1>' in resp.text)
```

And run nose to call the tests:

```
nosetests
.
----------------------------------------------------------------------
Ran 1 test in 0.117s

OK
```

3.3 Configuration

3.4 Routing

3.5 Application

3.6 Handlers

3.7 Session

3.8 Signals

3.9 Testing

CHAPTER

FOUR

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*