Watson - Dev

Release 3.0.1

Jan 15, 2018

Contents

Work with WSGI applications locally.

CHAPTER 1

Build Status

build passing coverage 100%

CHAPTER 2

Installation

```
pip install watson-dev
```

CHAPTER 3

Testing

Watson can be tested with py.test. Simply activate your virtualenv and run `python setup.py test`.

CHAPTER 4

Contributing

If you would like to contribute to Watson, please feel free to issue a pull request via Github with the associated tests for your code. Your name will be added to the AUTHORS file under contributors.

CHAPTER 5

Table of Contents

5.1 Reference Library

5.1.1 watson.dev.middleware

class watson.dev.middleware.**StaticFileMiddleware**(*app*, *initial_dir=None*)

A WSGI compatibile Middleware class that allows content to be retrieved from the directory that the __main__ is called from.

Example:

```
def app(environ, start_response):
    start_response('200 OK', [('Content-Type', 'text/plain')])
    return [b'Hello World!']

my_app = StaticFileMiddleware(app)
```

__init__(*app*, *initial_dir=None*)

5.1.2 watson.dev.reloader

```
# -*- coding: utf-8 -*-
import sys
import os
import time
import _thread as thread

_mtimes = {}


def code_changed():
    global _mtimes
    filenames = [getattr(m, "__file__", None) for m in sys.modules.values()]
    for filename in filter(None, filenames):
        if filename.endswith(".pyc") or filename.endswith(".pyo"):
            filename = filename[:-1]
        if filename.endswith("$py.class"):
            filename = filename[:-9] + ".py"
        if not os.path.exists(filename):
            continue
        stat = os.stat(filename)
        mtime = stat.st_mtime
        if filename not in _mtimes:
            _mtimes[filename] = mtime
            continue
        if mtime != _mtimes[filename]:
            _mtimes = {}
            return True
    return False


def main(main_func, args=None, kwargs=None, script_dir=None):
    import __main__
    thread.start_new_thread(main_func, args or (), kwargs or {})
    while True:
        if code_changed():
            script = __main__.__file__
            print('\nCode changed, reloading...\n')
            if script_dir:
                script = os.path.join(script_dir, script)
            script = os.path.abspath(script)
            python = sys.executable
            args = [script] + sys.argv[1:]
            os.execl(python, python, *args)
            sys.exit(3)
        try:
            time.sleep(1)
        except KeyboardInterrupt:
            print('\nTerminated.')
            sys.exit(0)
```

5.1.3 watson.dev.server

watson.dev.server.**make_dev_server**(*app*, *host='0.0.0.0'*, *port=8000*, *noreload=False*, *script_dir=None*, *public_dir=None*)

A simple local development server utilizing the existing simple_server module, but allows for serving of static files.

Never use this in production. EVER.

Example:

```
def my_app(environ, start_response):
    start_response('200 OK', [('Content-Type', 'text/html')])
    return [b'<h1>Hello World!</h1>']

if __name__ == '__main__':
    make_dev_server(my_app)
```

Parameters

- **`app`** – A WSGI callable
- **`host`** – The host to bind to
- **`port`** – The port
- **`noreload`** – Whether or not to automatically reload the application when source code changes.

Python Module Index

Index