
Watson - Dev

Release 1.0.2

April 08, 2015

1	Build Status	3
2	Installation	5
3	Testing	7
4	Contributing	9
5	Table of Contents	11
5.1	Reference Library	11
	Python Module Index	15

Work with WSGI applications locally.

Build Status

Installation

```
pip install watson-dev
```

Testing

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

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.

Table of Contents

5.1 Reference Library

5.1.1 watson.dev.middleware

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

A WSGI compatible 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

```
1  # -*- coding: utf-8 -*-
2  # Autoreloading launcher.
3  # Borrowed from Peter Hunt and the CherryPy project (http://www.cherrypy.org).
4  # Some taken from Ian Bicking's Paste (http://pythonpaste.org/).
5  # Sourced from the Django project (http://djangoproject.com)
6  #
7  # Portions copyright (c) 2004, CherryPy Team (team@cherrypy.org)
8  # All rights reserved.
9  #
10 # Redistribution and use in source and binary forms, with or without modification,
11 # are permitted provided that the following conditions are met:
12 #
13 #     * Redistributions of source code must retain the above copyright notice,
14 #       this list of conditions and the following disclaimer.
15 #     * Redistributions in binary form must reproduce the above copyright notice,
16 #       this list of conditions and the following disclaimer in the documentation
17 #       and/or other materials provided with the distribution.
18 #     * Neither the name of the CherryPy Team nor the names of its contributors
19 #       may be used to endorse or promote products derived from this software
20 #       without specific prior written permission.
```

```
21 #
22 # THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
23 # ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
24 # WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
25 # DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
26 # FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
27 # DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
28 # SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
29 # CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
30 # OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
31 # OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
32 import sys
33 import os
34 import time
35 import _thread as thread
36 from watson.common.contextmanagers import suppress
37
38 _mtimes = {}
39
40
41 def code_changed():
42     global _mtimes
43     filenames = [getattr(m, "__file__", None) for m in sys.modules.values()]
44     for filename in filter(None, filenames):
45         if filename.endswith(".pyc") or filename.endswith(".pyo"):
46             filename = filename[:-1]
47             if filename.endswith("$py.class"):
48                 filename = filename[:-9] + ".py"
49             if not os.path.exists(filename):
50                 continue
51             stat = os.stat(filename)
52             mtime = stat.st_mtime
53             if filename not in _mtimes:
54                 _mtimes[filename] = mtime
55                 continue
56             if mtime != _mtimes[filename]:
57                 _mtimes = {}
58                 return True
59     return False
60
61
62 def reloader_thread():
63     while True:
64         if code_changed():
65             sys.exit(3)
66         time.sleep(1)
67
68
69 def restart_with_reloader(script_dir=None):
70     import __main__
71     while True:
72         if not script_dir:
73             script = os.path.abspath(__main__.__file__)
74         else:
75             script = os.path.abspath(
76                 os.path.join(script_dir, __main__.__file__))
77         args = [sys.executable, script]
78         sys_argv = sys.argv[:]
```



```

79         if len(sys_argv) > 1:
80             sys_argv.pop(0)
81             args = args + sys_argv
82         new_environ = os.environ.copy()
83         new_environ['RUN_MAIN'] = 'true'
84         exit_code = os.spawnve(os.P_WAIT, sys.executable, args, new_environ)
85         if exit_code != 3:
86             return exit_code
87
88
89 def reloader(main_func, args, kwargs, script_dir=None):
90     if os.environ.get('RUN_MAIN') == 'true':
91         thread.start_new_thread(main_func, args, kwargs)
92         with suppress(KeyboardInterrupt):
93             reloader_thread()
94     else:
95         try:
96             exit_code = restart_with_reloader(script_dir)
97             if exit_code < 0:
98                 os.kill(os.getpid(), -exit_code)
99             else:
100                 sys.exit(exit_code)
101         except KeyboardInterrupt:
102             print('\nTerminated.')
103
104
105 def main(main_func, args=None, kwargs=None, script_dir=None):
106     reloader(main_func, args or (), kwargs or {}, script_dir)

```

5.1.3 watson.dev.server

`watson.dev.server.make_dev_server` (*app*, *host*='0.0.0.0', *port*=8000, *do_reload*=True, *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
- **do_reload** – Whether or not to automatically reload the application when source code changes.

W

`watson.dev.middleware`, [11](#)
`watson.dev.server`, [13](#)

Symbols

`__init__()` (watson.dev.middleware.StaticFileMiddleware
method), [11](#)

M

`make_dev_server()` (in module watson.dev.server), [13](#)

S

`StaticFileMiddleware` (class in watson.dev.middleware),
[11](#)

W

`watson.dev.middleware` (module), [11](#)
`watson.dev.server` (module), [13](#)