
Watchmen Documentation

Release 0.1.0

Joel Akeret

August 10, 2016

1	Contents:	3
1.1	watchmen Package	3
1.2	Contributing	3
1.3	Credits	4
1.4	History	5
2	Feedback	7
	Python Module Index	9

Watchdog implementation to monitor functions or methods.

To watch a longrunning function you can do the following

```
import time
import watchmen

@watchmen.watch(max_time=1)
def longrunning(t):
    time.sleep(t)

longrunning(t=10)
```

After 1 second an exception is raised:

```
watchmen.watchmen.WatchmenException: Time limit exceeded
```

If you want to watch the memory consumption of a function

```
@watchmen.watch(max_mem=100)
def memory_hungry(s):
    l = [i*j for i in range(s) for j in range(s)]

memory_hungry(s=10000)
```

After a while an exception is raised:

```
watchmen.watchmen.WatchmenException: Memory limit exceeded. RSS: 152.09765625 MB
```

watchmen can also be used on instance methods

```
class Demo(object):
    @watchmen.watch(max_mem=100)
    def memory_hungry(self, s):
        l = [i*j for i in range(s) for j in range(s)]

Demo().memory_hungry(s=10000)
```

Contents:

1.1 watchmen Package

1.1.1 watchmen Package

1.1.2 watchmen Module

Created on Aug 8, 2016

author: jakeret

`watchmen.watchmen.watch` (*original_function=None, max_mem=None, max_time=None, sample_rate=None*)

Decorator to watch a function or method.

Parameters

- **max_mem** – (optional) maximal memory limit in MB
- **max_time** – (optional) maximal execution time in seconds
- **sample_rate** – (optional) Rate at which the process is being queried

1.2 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

1.2.1 Types of Contributions

Report Bugs

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Implement Features

Write Documentation

Watchmen could always use more documentation, whether as part of the official Watchmen docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

1.2.2 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, and 3.3, and for PyPy. make sure that the tests pass for all supported Python versions.

1.2.3 Tips

To run a subset of tests:

```
$ py.test test/test_watchmen.py
```

1.3 Credits

1.3.1 Development Lead

- Joel Akeret <jakeret@phys.ethz.ch>

1.3.2 Contributors

None yet. Why not be the first?

1.4 History

1.4.1 0.1.0 (2016-08-09)

- xxx

Feedback

If you have any suggestions or questions about **Watchmen** feel free to email me at jakeret@phys.ethz.ch.

If you encounter any errors or problems with **Watchmen**, please let me know!

W

`watchmen.__init__`, 3
`watchmen.watchmen`, 3

W

`watch()` (in module `watchmen.watchmen`), 3

`watchmen.__init__` (module), 3

`watchmen.watchmen` (module), 3