
Grigori Documentation

Release 1.0.0

Steven Liebregt

Oct 22, 2019

Contents:

1	Quickstart	1
1.1	Requirements	1
1.2	Installation	1
1.3	Example	1
2	Usage	3
3	grigori	5
3.1	grigori package	5
	Python Module Index	7
	Index	9

1.1 Requirements

This module has no requirements. The *requirements.dev.txt* is for development to the module only and contains packages like Sphinx and setuptools.

1.2 Installation

Install the package using the following command.

```
pip install grigori
```

1.3 Example

The simple example below watches the current directory for any changes, and notifies us of the file and type of the change.

For more detailed usage please see the *Usage*

```
from grigori import Watcher

directory = './'

w = Watcher(directory)

for changes in w.watch():
    for change in changes:
        print("file '%s' has been changed in the following way: '%s'" % (change["file",
↪], change["type"]))
```


CHAPTER 2

Usage

More usage

3.1 grigori package

3.1.1 Submodules

3.1.2 grigori.watcher module

class grigori.watcher.Change

Bases: enum.IntEnum

Simple Enum representing the different types of changes to a file.

ADDED = 1

DELETED = 3

MODIFIED = 2

class grigori.watcher.Watcher(*directory: str, recursive: bool = False, polling_interval: int = 1000, file_pattern: str = '.+', directory_pattern: str = '.+', cache: bool = False*)

Bases: object

Class that starts watching your files.

on (*change_type: grigori.watcher.Change, callback: function*) → None

Register a callback function for a type of change.

Parameters

- **change_type** – The type of the change.
- **callback** – A function to call when a change of the given type occurs.

wait () → None

Hacky method to use the ‘watch’ method without a ‘for loop’.

watch() → generator

Keep polling for file changes and yield them.

Returns A generator that yields a list of changes.

3.1.3 Module contents

Basic watcher module.

The *grigori* module contains two classes:

- `grigori.Change`
- `grigori.Watcher`

You can use the `grigori.Watcher` class to create a watcher instance. Then you have the choice if you want to use callbacks or a for loop. To use callbacks, use the `grigori.Watcher.wait()` method. To use the for loop, use the `grigori.Watcher.watch()` method.

g

`grigori`, [6](#)
`grigori.watcher`, [5](#)

A

ADDED (*grigori.watcher.Change attribute*), 5

C

Change (*class in grigori.watcher*), 5

D

DELETED (*grigori.watcher.Change attribute*), 5

G

grigori (*module*), 6

grigori.watcher (*module*), 5

M

MODIFIED (*grigori.watcher.Change attribute*), 5

O

on () (*grigori.watcher.Watcher method*), 5

W

wait () (*grigori.watcher.Watcher method*), 5

watch () (*grigori.watcher.Watcher method*), 5

Watcher (*class in grigori.watcher*), 5