
retry.it Documentation

Release 0.0.1

Eli Uriegas

Nov 30, 2018

Contents

1	Indices and tables	3
	Python Module Index	5

Contents: A simple python module to add a retry function decorator

exception `retry.MaximumRetriesExceeded`

exception `retry.MaximumTimeoutExceeded`

`retry.retry` (*exceptions*=(<type 'exceptions.Exception'>,), *interval*=0, *max_retries*=10, *success*=None, *timeout*=-1)

Decorator to retry a function 'max_retries' amount of times

Parameters

- **exceptions** (*tuple*) – Exceptions to be caught for retries
- **interval** (*int*) – Interval between retries in seconds
- **max_retries** (*int*) – Maximum number of retries to have, if set to -1 the decorator will loop forever
- **success** (*function*) – Function to indicate success criteria
- **timeout** (*int*) – Timeout interval in seconds, if -1 will retry forever

Raises

- **MaximumRetriesExceeded** – Maximum number of retries hit without reaching the success criteria
- **TypeError** – Both exceptions and success were left None causing the decorator to have no valid exit criteria.

Example: Use it to decorate a function!

```
from retry import retry

@retry(exceptions=(ArithmeticError,), success=lambda x: x > 0)
def foo(bar):
    if bar < 0:
        raise ArithmeticError('testing this')
    return bar

foo(5)
# Should return 5
foo(-1)
# Should raise ArithmeticError
foo(0)
# Should raise MaximumRetriesExceeded
```


CHAPTER 1

Indices and tables

- `genindex`
- `modindex`
- `search`

r

`retry`, [1](#)

M

`MaximumRetriesExceeded`, [1](#)

`MaximumTimeoutExceeded`, [1](#)

R

`retry` (*module*), [1](#)

`retry()` (*in module `retry`*), [1](#)