
Twistless Documentation

Release 1.0.0

Taylor "Nekroze" Lawson

January 13, 2016

1 Usage	3
2 Feedback	5

A simple to use bridge between stackless python (available in pypy) and the twisted networking library.

The underlying implementation is based off of the excelent article by Stephen Coursen found here <http://www.stevencoursen.com/209/stackless-python-meets-twisted-matrix/> and was a huge help when writing twistless.

Twistless is designed to give a quick and easy start to using stackless python and twisted together. The following shows an overly brief example of a twisted reactor being started with stackless support.

```
1 from twistless import twistless
2 from twisted.internet import reactor
3
4
5 @twistless
6 def entry():
7     reactor.run()
8
9
10 if __name__ == "__main__":
11     entry()
```

Contents:

Usage

The following is a simple example of using the Twistless decorator to start a twisted reactor with stackless support.

```
1  """
2  This example is based off of the echo server example from twisted matrix
3  documentation.
4
5  The original client will work fine and can be found here
6  http://twistedmatrix.com/documents/13.1.0/core/examples/simpleclient.py
7  """
8  import time
9  from twistless import twistless, tasklet
10 from stackless import schedule
11 from twisted.internet import reactor, protocol
12
13
14 @tasklet
15 def async():
16     """A deferred executed in another tasklet."""
17     #Schedule this function to be continued at a later time.
18     schedule()
19     #Do something lengthy
20     time.sleep(5)
21     print("Tasklets!")
22
23
24 class Echo(protocol.Protocol):
25     """This is just about the simplest possible protocol"""
26
27     def dataReceived(self, data):
28         """
29         As soon as any data is received, write it back ASAP. But first setup a
30         function to be called when there is time for it.
31         """
32         #call the async deferred function in another tasklet
33         #The server will echo a response and then return to the tasklet
34         #schedule which has the async method waiting to be returned to.
35         async()
36         self.transport.write(data)
37
38
39 @twistless
40 def main():
41     """This runs the protocol on port 8000"""
```

```
42     factory = protocol.ServerFactory()
43     factory.protocol = Echo
44     reactor.listenTCP(8000, factory)
45     reactor.run()
46
47
48 # this only runs if the module was *not* imported
49 if __name__ == '__main__':
50     main()
```


Feedback

If you have any suggestions or questions about *Twistless* feel free to email me at nekroze@eturnilnetwork.com.

You can check out more of what I am doing at <http://nekroze.eturnilnetwork.com> my blog.

If you encounter any errors or problems with *Twistless*, please let me know! Open an Issue at the GitHub <http://github.com/Nekroze/twistless> main repository.