What is Twisted

Twisted is an open source event-driven networking framework written entirely in Python. It allows you to create a SMTP, HTTP, proxy and ssh servers in Python with minimal effort. Twisted is Asynchronous and event driven and allows applications to respond to different network connections without the use of traditional threading models.

Why to use Twisted

Twisted is a framework for writing asynchronous, event-driven networked programs in Python -- both clients and servers. Twisted is very fast. But is not suitable for writing conventional webapps. If you want to achieve some low-level networking functionality, twisted should be preferred.

How is Twisted different

Asynchronous: Twisted is asynchronous unlike many frameworks.
Protocol support: Twisted includes lots and lots of protocol implementations, meaning that more likely than not there will be an API you can use to talk to some remote system (either client or server in most cases) - be it HTTP, FTP, SMTP, POP3, IMAP4, DNS, IRC, MSN, OSCAR, XMPP/Jabber, telnet, SSH, SSL, NNTP, or one of the really obscure protocols like Finger, or ident, or one of the lower level protocol-building-protocols like DJB's netstrings, simple line-oriented protocols, or even one of Twisted's custom protocols like Perspective Broker (PB) or Asynchronous Messaging Protocol (AMP).
Threading: A new thread for every operation is not needed , which contributes to scalability, handling thousands of connections in a single thread, which tends to work better than having thousands of threads, each for a single connection.
Reliability: Avoiding threading is also beneficial for testing and debugging (and hence reliability in general). You generally don't need to worry about locking. Race conditions that depend on the order of different network events happening can easily be unit tested by simulating those network events.

Installation

The installation of twisted is fairly simple and quick. It can be installed using Pip as below,

```
$ pip install
twisted
```

Example

```
from twisted.web.server import Site
from twisted.web.resource import Resource
from twisted.internet import reactor, endpoints
import cgi

class FormPage(Resource):
    def render_GET(self, request):
        return (b"<!DOCTYPE html><html><head><meta charset='utf-8'>"
                b"<title></title></head><body>"
                b"<form method='POST'><input name='the-field'></form>")

    def render_POST(self, request):
        args = request.args[b"the-field"][0].decode("utf-8")
        escapedArgs = cgi.escape(args)
        return (b"<!DOCTYPE html><html><head><meta charset='utf-8'>"
                b"<title></title></head><body>"
                b"You submitted: " + escapedArgs.encode('utf-8'))

root = Resource()
root.putChild(b"form", FormPage())
factory = Site(root)
endpoint = endpoints.TCP4ServerEndpoint(reactor, 8880)
endpoint.listen(factory)
reactor.run()
```

Note: While it’s convenient for this example, it’s often not a good idea to make a resource that POSTs to itself; this isn’t about Twisted Web, but the nature of HTTP general; if you do this in a real application, make sure you understand the possible consequences.

You can run the example program with the following command in the terminal

```
python twisted_demo.py
```

When you visit the following URL on the browser it will display an input field, after submitting the page the same data from the input field will be printed on the screen, this is because we are handling both the ‘GET’ and ‘POST’ conditions within the same form, the browser will give an 404 error for any other request methods such as ‘DELETE’ which is not handled in the same class. However, in this way we can handle different request methods for the same resource endpoint or URL.

```
http://localhost:8880/form
```

Advantages

- Twisted's codebase is small and it contains many hooks for dynamic content, which makes it very flexible.
- Easy to make networking based applications
- Supports majority of networking protocols
- Highly secure and stable

Disadvantages

- Not suitable for conventional webapps and REST api’s.

Comparison with other Frameworks

As compared to other asynchronous or networking frameworks that python has to provide, Twisted is easy to understand and more flexible. At its heart is an event-driven networking engine called reactor. It is used for scheduling and calling user-defined callbacks.Unlike other event-driven frameworks Twisted boasts a rich library of event-driven utility classes for managing common protocols and programming tasks.