Part 6: Running our bot

Running the bot application and its tests.

Running tests + the bot with Twisted plugins

Running tests

Let’s first test to see if our tests pass. Twisted has a plugin command, `trial`, that will easily pick up the tests that we wrote. Within our `new-coder/network` directory, with our virtualenv activated, we’ll run our tests:

```
(NetworkProj)$ pwd
/Users/lynnroot/Dev/new-coder/network
(NetworkProj)$ trial tests
```

You should see something like the following:

```
(NetworkProj)$ trial tests
tests.test_quote_picker
  TestQuotePicker
    test_pick ...                                    [OK]
tests.test_talkbackbot
  TestTalkBackBot
    test_privmsgAttribution ...                      [OK]
    test_privmsgNoTrigger ...                        [OK]
    test_privmsgPrivateMessage ...                   [OK]
    test_privmsgWithTrigger ...                      [OK]

---------------------------------------------------------
Ran 5 tests in 0.020s

PASSED (successes=5)
```

Running our bot

Time to play with our bot! First, copy `settings.ini.EXAMPLE` to `settings.ini` and edit the file as necessary:

```
[irc]
endpoint = ssl:host=irc.freenode.net:port=7000
nickName = shereallysaid
realName = bot: provides quotations from notable women

channel = #newcoder

[talkback]
quotesFilename = quotes.txt
# Trigger phrases, in lowercase
triggers =
    that's what she said
```

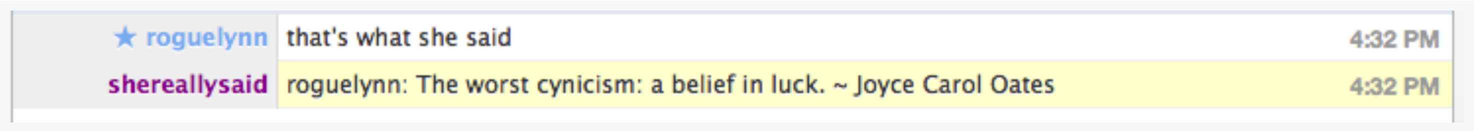
Then run the following command with our virtualenv activated:

```
(NetworkProj)$ twistd -n twsrs
```

and you should see the following:

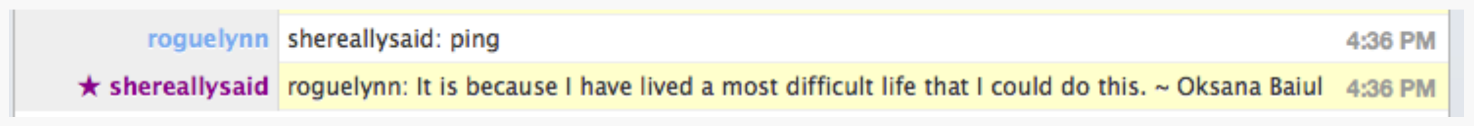
```
(NetworkProj)$ twistd -n twsrs
2013-06-27 16:27:51+0200 [-] Log opened.
2013-06-27 16:27:51+0200 [-] twistd 13.0.0 (/Users/lynnroot/.virtualenvs/NetworkP
2013-06-27 16:27:51+0200 [-] reactor class: twisted.internet.selectreactor.Select
2013-06-27 16:27:51+0200 [-] Starting factory <talkback.bot.TalkBackBotFactory ir
2013-06-27 16:27:51+0200 [Uninitialized] connectionMade
2013-06-27 16:28:01+0200 [TalkBackBot (TLSMemoryBIOProtocol),client] Signed on
2013-06-27 16:28:07+0200 [TalkBackBot (TLSMemoryBIOProtocol),client] [shereallysa
```

Now if you log onto irc.freenode.net as yourself and join the #newcoder channel, you should see `shereallysaid` in the channel. Try sending `that's what she said` in the channel:



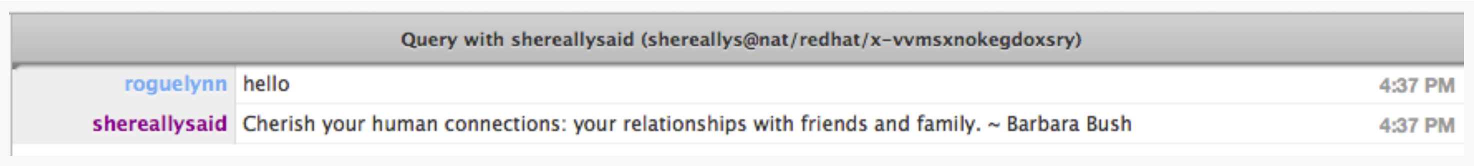
★ roguelynn that's what she said 4:32 PM
shereallysaid roguelynn: The worst cynicism: a belief in luck. ~ Joyce Carol Oates 4:32 PM

Now try sending `shereallysaid: ping` in the #newcoder channel:



roguelynn shereallysaid: ping 4:36 PM
★ shereallysaid roguelynn: It is because I have lived a most difficult life that I could do this. ~ Oksana Baiul 4:36 PM

And last, try sending a message to the bot directly: `/msg shereallysaid hello`:



Query with shereallysaid (shereallys@nat/redhat/x-vvmsxnokegdoxsry)

roguelynn hello 4:37 PM

shereallysaid Cherish your human connections: your relationships with friends and family. ~ Barbara Bush 4:37 PM

Looking back at our terminal, we can see the output of our logs:

```
(NetworkProj)$ twistd -n twsrs
2013-06-27 16:27:51+0200 [-] Log opened.
2013-06-27 16:27:51+0200 [-] twistd 13.0.0 (/Users/lynnroot/.virtualenvs/IRCbot/b
2013-06-27 16:27:51+0200 [-] reactor class: twisted.internet.selectreactor.Select
2013-06-27 16:27:51+0200 [-] Starting factory <talkback.bot.TalkBackBotFactory in
2013-06-27 16:27:51+0200 [Uninitialized] connectionMade
2013-06-27 16:28:01+0200 [TalkBackBot (TLSMemoryBIOProtocol),client] Signed on
2013-06-27 16:28:07+0200 [TalkBackBot (TLSMemoryBIOProtocol),client] [shereallysa
2013-06-27 16:32:34+0200 [TalkBackBot (TLSMemoryBIOProtocol),client] sent message
        The worst cynicism: a belief in luck. ~ Joyce Carol Oates
2013-06-27 16:36:48+0200 [TalkBackBot (TLSMemoryBIOProtocol),client] sent message
        It is because I have lived a most difficult life that I could do this. ~
2013-06-27 16:37:46+0200 [TalkBackBot (TLSMemoryBIOProtocol),client] sent message
        Cherish your human connections: your relationships with friends and famil
```

When you're done, within the terminal, press `CTRL+C` to terminate the bot. When doing so, you'll see the following:

```
2013-06-27 16:41:16+0200 [-] Received SIGINT, shutting down.
2013-06-27 16:41:16+0200 [TalkBackBot (TLSMemoryBIOProtocol),client] connectionLo
2013-06-27 16:41:16+0200 [TalkBackBot (TLSMemoryBIOProtocol),client] Stopping fac
2013-06-27 16:41:16+0200 [-] Main loop terminated.
2013-06-27 16:41:16+0200 [-] Server Shut Down.
```

Running tests + the bot with a Makefile

If you have `Make` installed (test in your terminal via `make --version`), you can create a Makefile, like so:

```
run:
    twistd -n twsrs

cov:
    coverage run --branch --source talkback  `which trial` tests
    coverage report
    coverage html

.PHONY: run cov
```

A Makefile simplifies commands for us – notice that these are just commands we could run in our terminal directly.. If we want to run the bot, all we'd have to do is `make run` to start our bot. If we want to run our tests, all we'd have to do is `make cov`.

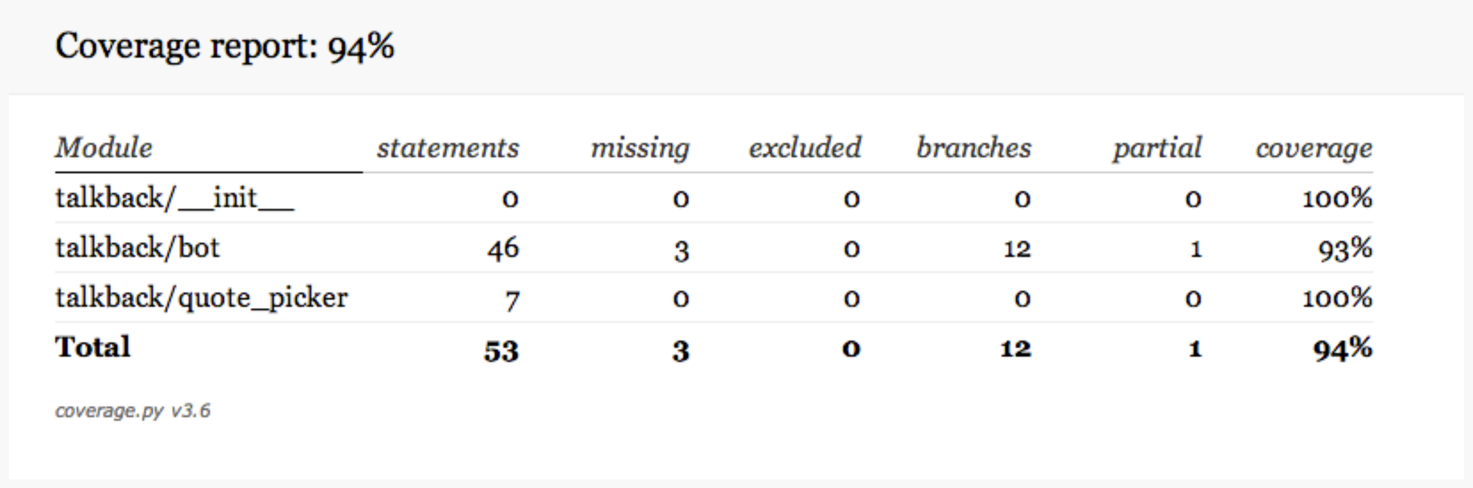
Running our tests via `make cov` makes use of `coverage` (a requirement that is listed in `requirements.txt` and should be in your virtualenv already). When we run `make cov`, we see this:

```
(NetworkProj)$ make cov
coverage run --branch --source talkback  `which trial` tests
tests.test_quote_picker
  TestQuotePicker
    test_pick ...                                                           [OK]
tests.test_talkbackbot
  TestTalkBackBot
    test_privmsgAttribution ...                                             [OK]
    test_privmsgNoTrigger ...                                               [OK]
    test_privmsgPrivateMessage ...                                          [OK]
    test_privmsgWithTrigger ...                                             [OK]

-------------------------------------------------------------------------------
Ran 5 tests in 0.087s

PASSED (successes=5)
coverage report
Name                    Stmts   Miss Branch BrMiss  Cover
---------------------------------------------------------
talkback/__init__           0      0      0      0   100%
talkback/bot               46      3     12      1    93%
talkback/quote_picker       7      0      0      0   100%
---------------------------------------------------------
TOTAL                      53      3     12      1    94%
coverage html
```

It runs our tests, but then also runs `coverage`, which assesses the amount of code that is covered by our tests. After its run, `coverage` creates a nice html page within `htmlcov` that you can view in your browser. Just open up the `index.html` file in `new-coder/networks/htmlcov/` and it will launch in your browser:

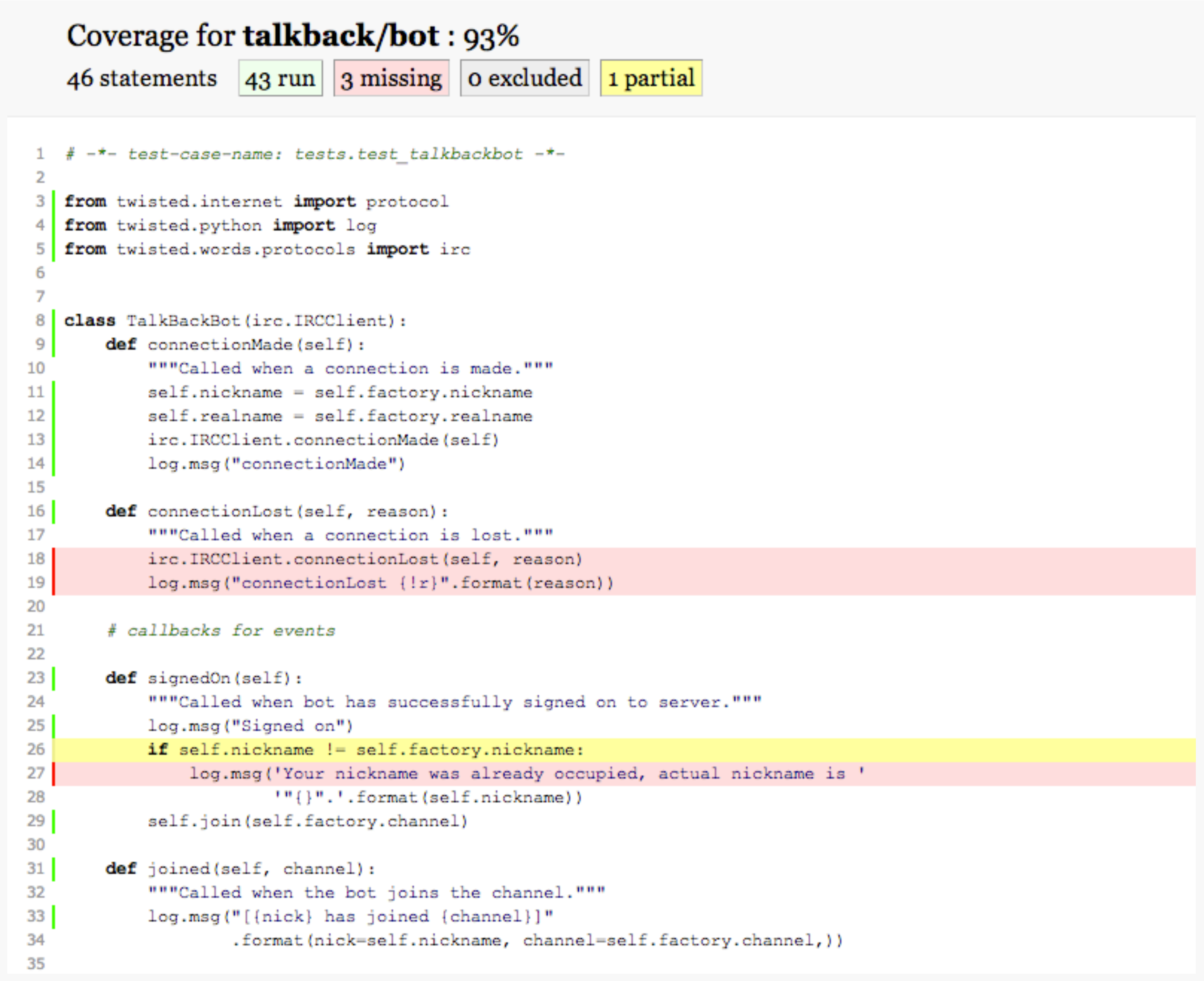


Coverage report: 94%

Module	*statements*	*missing*	*excluded*	*branches*	*partial*	*coverage*
talkback/__init__	0	0	0	0	0	100%
talkback/bot	46	3	0	12	1	93%
talkback/quote_picker	7	0	0	0	0	100%
Total	**53**	**3**	**0**	**12**	**1**	**94%**

coverage.py v3.6

You can click on line items to see more detail:



Coverage for **talkback/bot** : 93%

46 statements 43 run 3 missing 0 excluded 1 partial

```
# -*- test-case-name: tests.test_talkbackbot -*-

from twisted.internet import protocol
from twisted.python import log
from twisted.words.protocols import irc


class TalkBackBot(irc.IRCClient):
    def connectionMade(self):
        """Called when a connection is made."""
        self.nickname = self.factory.nickname
        self.realname = self.factory.realname
        irc.IRCClient.connectionMade(self)
        log.msg("connectionMade")

    def connectionLost(self, reason):
        """Called when a connection is lost."""
        irc.IRCClient.connectionLost(self, reason)
        log.msg("connectionLost {!r}".format(reason))

    # callbacks for events

    def signedOn(self):
        """Called when bot has successfully signed on to server."""
        log.msg("Signed on")
        if self.nickname != self.factory.nickname:
            log.msg('Your nickname was already occupied, actual nickname is '
                    '"{}".'.format(self.nickname))
        self.join(self.factory.channel)

    def joined(self, channel):
        """Called when the bot joins the channel."""
        log.msg("[{nick} has joined {channel}]"
                .format(nick=self.nickname, channel=self.factory.channel,))

```

You'll notice that we don't have 100% coverage. This is because there's no need to test log messages. Having a goal of 100% coverage can be unrealistic, and is not typical in the industry. To read more about what reasonable test coverage is, StackOverflow has a great answer (http://stackoverflow.com/questions/90002/what-is-a-reasonable-code-coverage-for-unit-tests-and-why) to such a question.

If you do not have `make` installed: you can still run coverage, just run the commands that are detailed in the Makefile:

```
(NetworkProj)$ coverage run --branch --source talkback  `which trial` tests
(NetworkProj)$ coverage report
(NetworkProj)$ coverage html
```

← Part 5: Testing the Bot (/networks/part-5/) Extended → (/networks/extended/)