

Writing IRC Bots

Spam, Glorious Spam

Craig Gavagan

November 4, 2015

What is IRC?

- IRC stands for Internet Relay Chat
- It is a chat protocol that allows users to communicate via text
- Though its use is in decline, major networks such as freenode and quakenet are still going strong
- Super awesome video time!

What is an IRC Bot?

- Essentially it's just a piece of code that connects to an IRC network and does something
- They can do any number of things, such as monitor channels and log messages
- They can respond to commands
- They can play games
- If you think it could be an IRC Bot, it most likely can be

How do I write one?

- 1. Pick your language and library, any will work, we're going to use python and irc3
- 2. Code
- 3. ???
- 4. Spam

Popular Bots

- teh
- TinyURL
- Dakdak
- stinson
- mincebot

What makes a good IRC Bot?

- Text based things work best
- Card games work quite well
- Utilities displaying information work great
- Graphics based games such as board games are particularly difficult to adapt successfully

Virtualenv

- Virtualenv is a way of installing python libraries on a shared host
- It is particularly useful on shared systems where you don't have root
- You can also have multiple environments so that you can have different versions of a particular library for multiple applications
- If developing a python application on Redbrick, Virtualenv is recommended

Virtualenv Installation

- Virtualenv is quite simple to install
- Simply head on over to pygmalion and enter the following commands

Virtualenv installation commands

```
virtualenv ircbot  
source ircbot/bin/activate
```

- This will have you setup with a new virtual environment and will activate it

Libraries

- Next we'll need to setup the libraries we'll be using
- Go ahead and run the following command to install the irc3 library

Installing libraries

```
pip install irc3
```

- We'll be installing another library later, but for now this will do

Bot Setup

- To get our first bot started we're going to run a simple command to use a basic template that irc3 provides for us

Create the template

```
mkdir talkbot  
cd talkbot  
python -m irc3.template talkbot
```

- This will give us a config.ini file with the bot configuration and a talkbot_plugin.py file that will echo anything with an !echo command and will say hello to anyone joining a channel
- Lets have a look at the configuration file

Config File

config.ini

```
nick = ircbot  
realname = ircbot  
  
host = localhost  
port = 6667
```

- Change nick and realname to the name you have chosen for your bot
- Change host to irc.redbrick.dcu.ie

Config File

config.ini

```
# the bot will join #ircbot_channel
autojoins =
    ircbot_channel
```

- Change this to botshop

- Finally, run the bot with the following command and lets see what irc3 has baked in to the bot for us

Starting the bot

```
irc3 config.ini
```

- Before we start making awesome bots, lets have a look at the talkbot_plugin.py file
- In particular lets look at the lines beginning with an @
- These are known as decorators and can modify the functionality of a function or class without modifying the source code
- The @irc3.plugin decorator registers a class as an irc3 plugin
- The @irc3.event decorator registers a function to be called when the specified event occurs
- The @command decorator makes the command that follows into a trigger

Getting started

- Now that we've had a look at the functionality of the bot lets get started on our first command
- We're going to create a simple trigger that will print the current time when someone says !time in a channel
- Start by deleting the say_hi and echo functions in talkbot_plugin.py

Time code

- Add the following line to the top of the talkbot_plugin.py where the other imports are

Import code

```
from datetime import datetime
```


Time code

- And then add the following code in the Plugin class

Time Command

```
@command(permission='view')
def time(self, mask, target, args):
    """Show the current time
    %%time
    """
    yield str(datetime.now())
```

- Not only does this give us a !time command we can use, but the docstring will be used in command !help time

Rock Paper Scissors

- Before we can get started on our second bot lets create another plugin
- We'll also add it to our config.ini so it will be used alongside our time plugin
- We'll also see how to make commands only available in private messages

RPS Plugin

rps_plugin.py

```
from irc3.plugins.command import command
import irc3

@irc3.plugin
class Plugin(object):
    def __init__(self, bot):
        self.bot = bot
        self.channel = '#botshop'
        self.played = {}
```

Rock Trigger

- Now, lets create our first trigger for this bot

```
@command(permission='view', public=False)
def rock(self, mask, target, args):
    """Play rock
       %%rock
    """
    if self.played == {}:
        self.bot.privmsg(self.channel, '%s has played
            !' % mask.nick)
        self.played = {'nick': mask.nick, 'choice': '
            rock'}
    elif self.played['choice'] == 'scissors':
        self.show_win(mask.nick, 'rock', self.played[
            'nick'], 'scissors')
    elif self.played['choice'] == 'paper':
        self.show_win(self.played['nick'], 'paper',
            mask.nick, 'rock')
    elif self.played['choice'] == 'rock':
        self.show_draw(self.played['nick'], mask.nick
            , 'rock')
```

```
@command(permission='view', public=False)
def paper(self, mask, target, args):
    """Play paper
       %%paper
    """
    if self.played == {}:
        self.bot.privmsg(self.channel, '%s has played
            !' % mask.nick)
        self.played = {'nick': mask.nick, 'choice': '
            paper'}
    elif self.played['choice'] == 'rock':
        self.show_win(mask.nick, 'paper', self.played
            ['nick'], 'rock')
    elif self.played['choice'] == 'scissors':
        self.show_win(self.played['nick'], 'scissors'
            , mask.nick, 'paper')
    elif self.played['choice'] == 'paper':
        self.show_draw(self.played['nick'], mask.nick
            , 'paper')
```

```
@command(permission='view', public=False)
def scissors(self, mask, target, args):
    """Play scissors
       %%scissors
    """
    if self.played == {}:
        self.bot.privmsg(self.channel, '%s has played
            !' % mask.nick)
        self.played = {'nick': mask.nick, 'choice': '
            scissors'}
    elif self.played['choice'] == 'paper':
        self.show_win(mask.nick, 'scissors', self.
            played['nick'], 'paper')
    elif self.played['choice'] == 'rock':
        self.show_win(self.played['nick'], 'rock',
            mask.nick, 'scissors')
    elif self.played['choice'] == 'scissors':
        self.show_draw(self.played['nick'], mask.nick
            , 'scissors')
```

Showing Win/Draw

```
def show_win(self, winner, winner_choice, loser,
             loser_choice):
    self.bot.privmsg(self.channel, '%s has beaten %s\'s %s with %s.' % (winner, loser,
                                                                           loser_choice, winner_choice))
    self.played = {}

def show_draw(self, first_player, second_player,
              choice):
    self.bot.privmsg(self.channel, '%s and %s have both played %s and drew.' % (first_player,
                                                                                       second_player,
                                                                                       choice))
    self.played = {}
```


Configuring the plugin

- Now we need to add the plugin to the config.ini file so irc3 loads the new plugin

```
includes =  
    irc3.plugins.command  
    ircbot_plugin
```

- Lets test the bot again and see if we have any errors

Lizard Spock

- Of course this plugin could be extended, some ideas:
- Lizard Spock
- Gambling

Permissions

- Throughout this talk, we've allowed all users to use all the commands
- Sometimes this is bad, since we don't want to create a command that can restart the bot, or even kick someone from a game, and we don't want all users to be able to use that command
- irc3 allows us to very easily control who can use which commands through the use of permissions.
- Lets have a look at the config file

Permissions

irc3 Permissions

```
guard = irc3.plugins.command.mask_based_policy

[irc3.plugins.command.masks]
# this section is used by the guard to secure the
  bot's command
# change your nickname and uncomment the line below
# creadak!*@* = all_permissions
* = view
```

Permissions

- If I uncomment the line with my nickname I will have all permissions, which means I will be able to execute all commands and functions this bot offers.
- The last line shows that all players have 'view' permissions
- This is a very flexible system, so you can create any permission model you need.

Parting Thoughts

- This has been a brief crash course, and there are an infinite amount of bots that we could have covered
- If you're interested in writing more bots, some ideas:
- A bot that monitors a github repository
- A bot that alerts you if you're running out of disk space
- A bot that shows information from an internet source, such as Dublin Bus or Irish Rail
- A bot that can interact with your favorite web app

Getting help

- Feel free to contact me, either on IRC, or at creadak@redbrick.dcu.ie
- Ask on IRC! The #Bots channel has lots of people who have written bots before
- <https://irc3.readthedocs.org/en/latest/>

Pizza! Pints!

PIZZA! PINTS!