Goshu IRC Bot Documentation

Release 4.0

Daniel Oaks

February 28, 2017

Contents

Goshu

A Modular, Extensible IRC Bot

Goshu is a powerful IRC bot, written in Python3. Utilising simple YAML configuration files, a fast, extensible module system and useful, configurable logging, Goshu allows you to create your own personalised IRC bot!

Goshu was written, initially, over the course of a few months to replace and replicate the functionality of a few other IRC bots. After a few revisions and rounds of updates, Goshu has a flexible, powerful feature-set, including built-in YAML-Python link recognition and data printing.

Warning: Although I do try my best to not break things when working on Goshu, I'm not aware of anyone besides myself using it. Thus, I'm not as worried about breaking module interfaces and such in minor ways, because I can fix my own modules without issue.
If you can't program in Python, it may be best to use another well-proven IRC bot project that's less likely to break compatability. That said, if you do run into anything, please feel free to mail me at daniel@danieloaks.net or make an issue in the Issue Tracker.

The documentation is split into two sections below, the first being for users of Goshu and the second for those who want to work on or write their own modules for Goshu.

CHAPTER 1

Installing and Configuring Goshu

Goshu makes it fairly easy to install, so long as you have a Unix-like environment. You can do the same thing on a Windows machine, but there will be more playing around to get packages and such working properly.

Prerequisites

- Git (to download the project itself)
- Python 3, with Pip installed

Installing

First off, get a copy of Goshu using Git – this method makes it extremely easy to update and get new patches and fixes.

```
$ git clone https://github.com/DanielOaks/goshu.git
```

Install Goshu's Python dependencies – this may take a while:

```
$ pip3 install -r requirements.txt
```

And start the bot itself with:

```
$ python3 goshu.py
```

Configuring

Once started as above, Goshu will ask you several questions including nicknames to use, master passwords, and IRC networks to connect to.

- Nick: This is the default nickname Goshu will use when connecting to IRC networks
- Password: This is the password users will enter to become a 'Bot Owner'
- Bot Command Prefix: This is what users will use to access Goshu's commands. For instance, `.` would require commands like `.help`, `'` would require commands like `'help`

Warning: The Password above is EXTREMELY IMPORTANT. If a user can guess that, they can take control of your IRC bot and do anything they want with it.
MAKE SURE THIS PASSWORD IS SECURE.

Account Setup

Once the bot connects to your given IRC networks, you'll need to make sure you can control the bot! You need to create a bot account, login, and make yourself a 'Bot Owner', and the below log shows how you can do that.

```
<dan> 'register coolguy57 usrpaswd
<goshu> Account registered!
<dan> 'login coolguy57 usrpaswd
<goshu> Login accepted!
```

```
<dan> 'list
<goshu> *** Commands: calc, d, def, egg, google, help, list, loggedin, login, nickserv, pokemon, poke
```

```
<dan> 'owner botpassword
<goshu> You are now a bot owner
```

```
<dan> 'list
<goshu> *** Commands: calc, d, def, egg, google, help, info, join, list, loggedin, login, me, module,
```

Once you've done this, you're right to go, Goshu is configured and ready to use! Simply invite her into whatever channel you want, and you can relax.

The next section will deal with day-to-day operations of the bot, and how to appoint Bot Admins to keep an eye on your bot when you can't.

CHAPTER 2

Day-to-Day Operations

When running an IRC bot, it's useful to have administrative control over it. Things like module reloading, user privilege control, and even just being able to send messages and `/me` statements as your bot can be very useful tools.

Appointing 'Bot Admins'

Bot Admins are people who can access lots of the backend functions of Goshu. This includes loading / unloading / reloading modules, doing the same with the internal JSON stores, using the `'me`, `'msg`, and other sorts of commands.

To appoint a Bot Admin, you need to make sure the user has created their own account with your Goshu IRC bot. After this, you simply set their account level with the following:

```
<dan> 'login coolguy57 usrpaswd
<goshu> Login accepted!
<dan> 'setaccess my_good_friend admin
<goshu> Setting my_good_friend's access level to 5
```

This lets them do things with your Goshu bot if, for instance, it plays up and you're not online.

Note: Users are only able to give other users access levels up to and including their own. If you're an admin, the max level you can give another user is admin level, etc.

In addition, only Admin and above are able to use the `setaccess` command.

Access Levels

Access levels go from 0 to 10, or:

```
*(5)* Admin
*(7)* Superadmin
*(10)* Owner
```

CHAPTER 3

Frequently Asked Questions

Why did you create Goshu?

Basically, a bunch of users were bringing bots into a channel I was a part of to provide information on links, provide silly and fun commands, and all sorts of unrelated features. It got to the point where there were about six bots on a 20-user channel, so I decided to create something that incorporated the features of the rest of them.

Goshu makes it extremely easy to create commands that are not only useful, but also commands that serve no real purpose besides having fun. Some examples include a command to select and display the name of an anime character, commands to replicate an 8ball, and to replicate a "spin-the-bottle"-like query, choosing one random user in the channel.

Obviously, how you use your bot and whether you choose to include/write these 'fun' commands really depends on your channel. Some more light-hearted channels can easily take a bot and create some sort of 'character' around it, providing all sorts of fun, light-hearted commands, whereas some more professional channels can just take the useful features and ignore the rest.

Isn't Python too slow for something like this? Why not use <language>?

Surprisingly not!

Python makes it very easy to create new modules, so long as you're familiar with Python scripting. It also lets us code new functionality into the core without much trouble, as well as do all this in a small, concise package.

Python's not everyone's cup of tea, but it's worked out fairly well for this project. If you are familiar with some other language and want an IRC bot you can write modules and plugins for, it could be best to look into another project. It's always best to go with something you can use – we won't be mad!

If I write a cool module, do I *need* to release it to everyone?

We would love it if you could, or if you would even try to contribute back to the primary Goshu repository so that everyone running the software can get the great new features!

However, you don't need to. Because Goshu is licensed under the ISC license, you have the flexibility to keep any changes to yourself if you want to.