
Bo Documentation

Release 0.1-alpha

Bo developers

February 06, 2015

1	Getting started with Bo	3
1.1	Installation	3
1.2	Basic usage	3
2	Builtin Connectors	5
2.1	bo.dist.connectors.console	5
2.2	bo.dist.connectors.xmpp	5
3	Builtin Plugins	7
3.1	bo.dist.plugins.echo	7
3.2	bo.dist.plugins.uptime	7
3.3	bo.dist.plugins.rand	7
3.4	bo.dist.plugins.logs	7

Bo is a chatbot written in Python. You can run it as a standalone program or embed it in your program. Out of the box, Bo doesn't do much but you can teach it how to do stuff, like reacting to messages, by using *plugins*. Bo can also connect to any network provided the right *connector*.

Note: Although Bo is currently quite usable, it is in alpha and under development. Until a stable version is reached, features may be added and APIs may change.

The first Bo instance became operational on September 8, 2012.

Contents:

Getting started with Bo

1.1 Installation

Bo is best installed from `pypi` via `pip` (recommended) or `easy_install`. The use of `virtualenv` is highly recommended to avoid polluting the global Python environment. For example, to install Bo using `pip`:

```
$ pip install bo
```

1.1.1 Deploy on Heroku

Instead of running it local, you can deploy Bo on [Heroku](#):

```
$ git init bo-heroku
$ cd bo-heroku/
$ heroku create --stack cedar
$ wget -O config.yaml "https://raw.githubusercontent.com/konikos/bo/master/conf/xmpp.yaml"
$ echo "bo" >requirements.txt
$ echo "worker: bo -c config.yaml" >Procfile
```

Set your password in `config.yaml` and probably add some plugins.

Warning: If you use a GMail account, read [this](#) to avoid getting locked out of your account.

Now, run the following to get Bo working:

```
$ git commit -a -m 'init'
$ git push heroku master
$ heroku ps:scale worker=1
```

If you are having issues running Bo, use `heroku logs` to examine the output from Bo and `heroku restart` to restart it.

1.2 Basic usage

After installation you can test the `bo` command:

```
$ bo
usage: bo [-h] -c CONFIG [-l LOGFILE] [-q | -d]
bo: error: argument -c/--config is required
```

The `bo` commands requires a configuration file to set up a bot. Configuration files are written in [YAML](#). Here is an example to get you started:

```
name: Bo Example

plugins:
- bo.dist.plugins.rand
- bo.dist.plugins.uptime
- module: bo.dist.plugins.logs
  config:
    max_entries: 100
- bo.dist.plugins.sing

connectors:
- name: Console
  module: bo.dist.connectors.console
```

The configuration is pretty self-explanatory; it gives a name to the bot, and sets up some plugins as well as the console connector in order to interact with the bot via the console. The `bo.dist.plugins.logs` entry is different than the rest because configuration is needed for the plugin (in this case, the `max_entries` option).

Save the configuration as `console.yaml` and run `bo` to interact with Bo for the first time:

```
$ bo -c /path/to/console.yaml
```

Built-in Connectors

Connectors add to Bo the ability to connect to various networks. Described below is a list of all connectors that come with Bo.

2.1 bo.dist.connectors.console

A simple connector that lets you interact with Bo from the console. Write your message and hit Enter! This connector also supports “special” commands that emulate things like group invites. Enter `:help` to get a full list.

No parameters

2.2 bo.dist.connectors.xmpp

Connector for XMPP networks like Google Talk.

Parameters

- **username** – XMPP username
- **password** – XMPP password
- **gtalk** – True/False for whether this is a connector for Google Talk.
- **server** – The address of the XMPP server. Not needed if gtalk is True.
- **port** – The port of the XMPP server. Not needed if gtalk is True.

Warning: If you are using a Google Talk account and deploy your bot in a location than you are, make sure you are logged in the first time you run your bot in that location, it will deny access to Bo and send you an email asking you to confirm you are logged in to confirm it, you may get locked out of your account. It is possible to get back in but it is annoying. You have been warned!

Builtin Plugins

Bo comes packaged with some plugins that provide basic functionality and can be used as a template to make yours.

3.1 `bo.dist.plugins.echo`

Simply responds to any message with the same text.

3.2 `bo.dist.plugins.uptime`

Responds to the `uptime` command with info about the uptime, linux-style.

3.3 `bo.dist.plugins.rand`

Random operations like rolling a dice.

Commands:

- `roll` – roll a 6 sided dice
- `roll a N` – roll an N-sided dice

3.4 `bo.dist.plugins.logs`

Captures Bo log messages and displays them as chat messages. Currently only the last 25 messages are kept.

Commands:

- `logs` – Emit all captured log messages.
- `logs N` – Emit the last N captured messages.