Botibal Documentation

Release 0.8.0

VirtualTam

Oct 01, 2018

User Documentation

A silly, quizzical XMPP bot based on the SliXMPP (3.5+) library.

CHAPTER 1

Installation

Botibal is currently compatible with Python 3.5, 3.6 and 3.7, and has been tested on Linux.

1.1 With pip (system-wide)

The simplest way to install Botibal is with pip:

```
$ pip install botibal
```

1.2 With pip & virtualenv (recommended)

To install Botibal in a Python virtualenv (here with Python 3.5 as the default interpreter):

```
# create a new virtualenv
$ virtualenv /path/to/envs/botibal

# activate the virtualenv
$ source /path/to/envs/botibal/bin/activate

# install botibal
(botibal) $ pip install botibal

# check which packages have been installed
(botibal) $ pip freeze
aiodns==1.0.1
botibal==0.7.5
pyasn1==0.1.9
pyasn1-modules==0.0.8
slixmpp==1.1
```

To specify a Python interpreter:

```
# create a new Python 3 virtualenv
$ virtualenv -p /usr/bin/python3.5 /path/to/envs/botibal3

# activate the virtualenv
$ source /path/to/envs/botibal3/bin/activate

# install botibal
(botibal3) $ pip install botibal

# check which packages have been installed
(botibal3) $ pip freeze
aiodns==1.0.0
botibal==0.7.5
pyasn1==0.1.9
pyasn1-modules==0.0.8
pycares==1.0.0
slixmpp==1.1
wheel==0.24.0
```

1.3 From the sources

To install Botibal from the latest source revision and install it in a dedicated virtualenv:

```
# fetch the sources
$ git clone https://github.com/virtualtam/botibal
$ cd botibal

# create and activate a new virtualenv
$ virtualenv /path/to/botibal-src
$ source /path/to/botibal-src/bin/activate

# upgrade pip (required to read requirements attributes)
(botibal-src) $ pip install -U pip

# build and install
(botibal-src) $ make install

# check which packages have been installed
(botibal3) $ pip freeze

aiodns==1.0.0
botibal==0.7.5
pyasn1==0.1.9
pyasn1-modules==0.0.8
pycares==1.0.0
slixmpp==1.1
wheel==0.24.0
```

CHAPTER 2

Configuration

Botibal relies on a simple INI file to hold its configuration. An example configuration file is provided in the sources.

2.1 Example configuration file

For the bot to work properly, a configuration file must be created:

- it can be located anywhere on the disk,
- it will be passed as a mandatory argument when running a bot (see *Usage*),
- it holds connection and administration information (JIDs, MUC).

Listing 1: `config.example.ini`

```
[auth]
jid=imahbot@lulzse.cx
password=OhGoshWhereAreTheKeys?

[muc]
room=1408@skyr.im
admin_jid=thecat@saw.it

[nick]
botibal=Botibal
minibal=Minibal
quizzibal=Quizzibal
```

CHAPTER 3

XMPP bots

3.1 MiniBal, the minimalist bot

- connects to a groupchat,
- answers to plopping: `<user>: plop <bot>` => `<bot>: plop <user>`,
- allows to taunt users!
- gives the current date and time.

3.2 Botibal, the silly bot

- all of MiniBal's features,
- knows basic text ciphering (currently, ROT13).

3.3 Quizzibal, the quizzical bot

- all of MiniBal's features,
- quizz sessions!
- quizz handling: start/stop, display scores, manage questions.

CHAPTER 4

Usage

4.1 Start the bot

Run with no arguments to get the help string:

```
$ botibal [-h] [-d] (-b | -m | -q) config_file database_file

positional arguments:
  config_file      configuration file
  database_file    data storage file

optional arguments:
  -h, --help       show this help message and exit
  -d, --debug      set logging to DEBUG
  -b, --botibal    BotiBal, the silly bot
  -m, --minibal    MiniBal, the minimalist bot
  -q, --quizzibal  QuizziBal, the quizzical bot
```

To start MiniBal:

```
$ botibal -m ~/botibal/config.ini ~/botibal/botibal.db
```

4.2 Chat commands

The available commands depend on which bot is running, and are split as follows:

- *MUC* - command execution (say/display something);
- *PM* - bot admin:
  - add, delete and list elements,
  - change bot status,

  - send instructions that will result in the bot sending messages to the MUC.

The bots' internal command interface is derived from argparse, interaction is thus very similar to running programs from the command-line (see the *Examples* section).

Once the bot is online and connected to a groupchat, you can:

- get help for MUC commands
  - list all available commands: `<bot_nick>: -h`
  - get help for a given command: `<bot_nick>: <command> -h`
- execute MUC commands: `<bot_nick>: command [args]`
- get help for PM commands:
  - list all available commands: `-h`
  - get help for a given command: `<command> -h`
- execute PM commands: `command [args]`

CHAPTER 5

Examples

5.1 MUC interaction (MiniBal)

```
<Hans> plop Minibal
<Minibal> plop Hans
[...]
<Hans> Minibal: -h
<Minibal>
usage: Minibal:  [-h] {say,time} ...

positional arguments:
  {say,time,taunt}
    say             say something
    time
[...]
<Hans> Minibal: say -h
<Minibal>
usage: Minibal: say [-h] text [text ...]

positional arguments:
  text
[...]
<Hans> Minibal: say hello, world!
<Minibal> hello, world!
```

5.2 PM interaction (MiniBal)

```
<Hans> -h
<Minibal>
usage: Minibal:  [-h] {say,time,quit,taunt} ...
```

(continues on next page)

(continued from previous page)

```
positional arguments:
  {say,time,quit,taunt}
    say                 say something
    time
    quit                tells the bot to stop
    taunt               manage taunts
[...]
<Hans> taunt add programming, do you speak it, mofo?
<Hans> taunt --list
<Minibal>
1 - programming, do you speak it, mofo? (Hans)
```

CHAPTER 6

Developer resources

See also:

- *Installation*
- *Configuration*

6.1 Requirements

To install *production* and *test* dependencies:

```
(botibal) $ pip install -r requirements/tests.txt
```

To install *production*, *test* and *development* dependencies:

```
(botibal) $ pip install -r requirements/development.txt
```

6.2 Tools

Static analysis:

- isort: check imports ordering & formatting;
- pycodestyle: miscellaneous language checks (formerly PEP8);
- pydocstyle: check doctsring formatting (formerly PEP257);
- Pylint: all-in-one syntax checker.

Tests:

- coverage;
- pytest;

- unittest.

6.2.1 Tox

Alternatively, if you have Tox installed, as well as a Python 3.5+ interpreter available:

```
# yup, it's that simple ;-)
$ tox
```