

---

# **lala Documentation**

***Release 0.4***

**Wieland Hoffmann**

August 09, 2015



<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	PyPI . . . . .	3
1.2	Git . . . . .	3
1.3	Setup . . . . .	3
1.4	Starting the bot . . . . .	4
<b>2</b>	<b>Updating</b>	<b>5</b>
<b>3</b>	<b>Configuration file options</b>	<b>7</b>
<b>4</b>	<b>Plugins</b>	<b>9</b>
<b>5</b>	<b>Lala IRC Bot</b>	<b>11</b>
<b>6</b>	<b>Lala Config Module</b>	<b>13</b>
<b>7</b>	<b>Lala utils</b>	<b>15</b>
<b>8</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>



Contents:



---

## Installation

---

### 1.1 PyPI

Lala is available on the Python Package Index. This makes installing it with [pip](#) as easy as:

```
pip install lala
```

### 1.2 Git

If you want the latest code or even feel like contributing, the code is available on [Github](#).

You can easily clone the code with git:

```
git clone git://github.com/mineo/lala.git
```

Now you can start hacking on the code or install it system-wide:

```
python2 setup.py install
```

### 1.3 Setup

Setting up the bot is relatively easy: Simply copy the supplied example config file from:

```
/usr/share/lala/config.example
```

to either:

```
/etc/lala.config
```

or:

```
$XDG_CONFIG_HOME/lala/config
```

or (if the latter is not set):

```
$HOME/.lala/config
```

After that, you can modify the config file as you wish.

## 1.4 Starting the bot

Starting the bot is very easy as it installs a `twistd` plugin:

```
twistd lala
```

For now, only a `--verbose` option is provided which makes the output written to the log file more verbose.

Additionally, `twistd` itself offers a few options, use:

```
twistd --help
```

to view them.



---

## Updating

---

Some commits require a manual update of the SQLite database used for the quotes.

The following is a list of those commits and the commands to run the update, where `<sqlite-file>` is the path to the quote database file. Newer commits appear first.

- 410ccde58bf6f635b121c0a5349dce5fd643679a (part of release xx.xx)  
`sqlite3 <sqlite-file> misc/2-add-votes-to-quotes.sql`
- a602b32dfa8aeb0d89b18275fe9e371199f3fd7c (part of release xx.xx)  
`sqlite3 <sqlite-file> misc/1-add-id-columns.sql`
- 44f8b476a1e28af60b5e21704b5c1eafd2f9bb54 (part of release 0.3)  
`sqlite3 <sqlite-file> misc/0-add-author-to-quotes.sql`



---

## Configuration file options

---

Both the basic configuration and the plugin configuration of Lala takes place in a central configuration file.

The basic options, which can be set in the *base* section, are:

- **server** The server to connect to.
- **port** The port on which the server listens.
- **nick** The nick of the bot.
- **admins** A comma-separated list of people allowed to issue admin-only commands (like reconnect or quit) to the bot.
- **nickserv\_password** Optional.  
The password which is used to identify with Nickserv.
- **channels** Optional.  
A comma-separated list of channels which are joined automatically after connection to the server.
- **fallback\_encoding** Optional.  
Lala assumes all messages received are UTF-8 encoded. If that's not the case, this is the second encoding to try.
- **plugins** Optional.  
A comma-separated list of plugins to load at startup. The plugin "base" is always loaded, it contains basic commands like "help" and "reconnect".
- **debug** Optional.  
if "yes", this has the same effect as specifying `-verbose` on the command line.



---

## Plugins

---

- birthday
- calendar
- decide
- down
- fortune
- httptitle
- last
- log
- quotes
- roulette

API:



---

## Lala IRC Bot

---

```
class lala.bot.Lala(*args, **kwargs)
    Bases: twisted.words.protocols.irc.IRCClient

    action(user, channel, data)
        Called when a user performs an ACTION on a channel.

    irc_RPL_WHOSREGNICK(prefix, params)

    joined(channel)
        Called after joining a channel.

    lineRate = 1

    modeChanged(user, channel, set, modes, args)
        The mode of a user has been changed. If it was added by Chanserv and the user is in the admin list,
        append him to identified_admins.

    msg(channel, message, log, length=None)
        Sends message to channel.

        Depending on log, the message will be logged or not.

        Do not use this method from plugins, use lala.util.msg() instead.

    nickname

    noticed(user, channel, message)
        Same as lala.bot.Lala.privmsg() for NOTICES.

    privmsg(user, channel, message)
        Handles received messages.

    signedOn()
        Called after a connection to the server has been established.

        Joins all configured channels and identifies with Nickserv.

    userJoined(user, channel)
        Handles join events.

    userKicked(user, message)

    userLeft(user, channel)

    userQuit(user, message)

    versionName = 'lala'

    versionNum = '0.4'
```





---

## Lala Config Module

---

### Config module

`lala.config.get(key, converter=None)`

Returns the value of a config option. The section is the name of the calling file.

Default values for all keys can be set with `set_default_options()`.

**Parameters** `key` – The key to lookup

`lala.config.get_int(*args)`

Returns the value of a config option as an int.

**Parameters** `*args` – See `lala.config.get()`

**Return type** `int`

`lala.config.get_list(*args)`

Gets a list option.

**Parameters** `*args` – See `lala.config.get()`

**Return type** list of strings

`lala.config.set(key, value, plugin=None)`

Sets the value of key. The section is the name of the calling file.

`lala.config.set_default_options(**kwargs)`

Sets the default options for a plugin.

The names of the arguments in `kwargs` will be used as the option names, the values as the values of the options.

`lala.config.set_list(key, value, *args)`

Sets option key to value where value is a list of values.

None of the values in value are allowed to contain `lala.config._LIST_SEPARATOR`.

This method does *not* preserve the type of the items in the list, they're all passed through `str()`.

**Parameters**

- **key** – See `lala.config.set()`
- **value** – A list of values for key.

`lala.config._LIST_SEPARATOR = ','`

Used as a separator when storing lists of values in the config file



---

## Lala utils

---

Helpers to be used with plugins

**class** `lala.util.command` (*command=None, admin\_only=False, aliases=None*)  
 Bases: `object`

Decorator to register a command. The name of the command is the `__name__` attribute of the decorated function.  
 Example:

```
@command
def heyiamacommand(user, channel, text):
    pass
```

You can also pass a `command` parameter to overwrite the name of the command:

```
@command(command="yetanothercommand")
def command_with_a_really_stupid_or_insanelly_long_name(user,
channel, text):
    pass
```

`aliases` can be a list of names under which the function will be available in addition to its primary name.

An additional argument, `admin_only` can be used to make a function available to admins only:

```
@command(admin_only=True)
def give_me_the_one_ring(user, channel, text):
    pass
```

`lala.util.msg` (*target, message, log=True*)

Send a message to a target.

### Parameters

- **target** (*str*) – Target to send the message to. Can be a channel or user
- **message** (*str or [str]*) – One or more messages to send
- **log** (*bool*) – Whether or not to log the message

`lala.util.on_join` (*f*)

Decorator for functions reacting to joins

**Parameters** **f** – The function which should be called on joins.

**class** `lala.util.regex` (*regex*)

Bases: `object`

Decorator to register a regex. Example:

```
@regex("(https?://.+)\s?")
def somefunc(user, channel, text, match_obj):
    pass
```

`match_obj` is a `re.MatchObject`.

**Parameters** `regex` – A `re.RegexObject` or a string representing a regular expression.

`lala.pluginmanager.disable(trigger)`

Disables *trigger*.

**Parameters** `trigger` (*str*) – The trigger to disable. Can be a key for a callback or a regular expression

`lala.pluginmanager.enable(trigger)`

Enables *trigger*.

**Parameters** `trigger` (*str*) – The trigger to enable. Can be a key for a callback or a regular expression

`lala.pluginmanager.is_admin(user)`

Check whether `user` is an admin.

If a nickserv password is set, this will work by checking an internal list of identified admins.

If no nickserv password is set, this simply checks if `user` is in the “admins” option of the “base” section.

**class** `lala.pluginmanager.PluginFunc` (*func, enabled=True, admin\_only=False, aliases=None*)

Bases: `object`

`lala.pluginmanager.load_plugin(name)`

---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



I

`lala.config`, [13](#)  
`lala.pluginmanager`, [16](#)  
`lala.util`, [15](#)





## Symbols

`_LIST_SEPARATOR` (in module `lala.config`), 13

## A

`action()` (`lala.bot.Lala` method), 11

## C

`command` (class in `lala.util`), 15

## D

`disable()` (in module `lala.pluginmanager`), 16

## E

`enable()` (in module `lala.pluginmanager`), 16

## G

`get()` (in module `lala.config`), 13

`get_int()` (in module `lala.config`), 13

`get_list()` (in module `lala.config`), 13

## I

`irc_RPL_WHOSREGNICK()` (`lala.bot.Lala` method), 11

`is_admin()` (in module `lala.pluginmanager`), 16

## J

`joined()` (`lala.bot.Lala` method), 11

## L

`Lala` (class in `lala.bot`), 11

`lala.config` (module), 13

`lala.pluginmanager` (module), 16

`lala.util` (module), 15

`lineRate` (`lala.bot.Lala` attribute), 11

`load_plugin()` (in module `lala.pluginmanager`), 16

## M

`modeChanged()` (`lala.bot.Lala` method), 11

`msg()` (in module `lala.util`), 15

`msg()` (`lala.bot.Lala` method), 11

## N

`nickname` (`lala.bot.Lala` attribute), 11

`noticed()` (`lala.bot.Lala` method), 11

## O

`on_join()` (in module `lala.util`), 15

## P

`PluginFunc` (class in `lala.pluginmanager`), 16

`privmsg()` (`lala.bot.Lala` method), 11

## R

`regex` (class in `lala.util`), 15

## S

`set()` (in module `lala.config`), 13

`set_default_options()` (in module `lala.config`), 13

`set_list()` (in module `lala.config`), 13

`signedOn()` (`lala.bot.Lala` method), 11

## U

`userJoined()` (`lala.bot.Lala` method), 11

`userKicked()` (`lala.bot.Lala` method), 11

`userLeft()` (`lala.bot.Lala` method), 11

`userQuit()` (`lala.bot.Lala` method), 11

## V

`versionName` (`lala.bot.Lala` attribute), 11

`versionNum` (`lala.bot.Lala` attribute), 11