

---

# **irclog Documentation**

***Release 0.2.0***

**Hong Minhee**

**Mar 24, 2017**



---

## Contents

---

<b>1</b>	<b>Getting started</b>	<b>3</b>
1.1	How to install . . . . .	3
1.2	How to run the web server . . . . .	3
1.3	How to use your WSGI server . . . . .	4
<b>2</b>	<b>irclog — IRC log viewer</b>	<b>5</b>
2.1	irclog.messages — Various message types . . . . .	5
2.2	irclog.parser — IRC log parser . . . . .	10
2.3	irclog.archive — IRC log archive . . . . .	11
2.4	irclog.web — Web IRC log view . . . . .	16
2.5	irclog.web.server — Built-in web server . . . . .	19
2.6	irclog.tests — Unit tests . . . . .	19
2.7	irclog.version — Version info . . . . .	20
<b>3</b>	<b>About distribution</b>	<b>21</b>
<b>4</b>	<b>Indices and tables</b>	<b>23</b>
	<b>Python Module Index</b>	<b>25</b>



**irclog** is a web viewer of logs [Irssi](#) generates. It has a WSGI interface and a built-in server both. If you don't know about WSGI and just want to use this log viewer, follow this: [Getting started](#).



# CHAPTER 1

---

## Getting started

---

### How to install

**irclog** is wholly written in **Python**, so you have to install Python on your system if it isn't installed yet.

It depends on some Python libraries also, but its install resolves the dependencies automatically. You can install **irclog** through **easy\_install** or **pip**:

```
$ pip install irclog
$ # ... or:
$ easy_install irclog
```

### How to run the web server

It has a built-in web server. What you have to do is just to use it. (Assume that IRC logs are stored in `.irssi/log/` of your home directory.)

```
$ python -m irclog.web.server \
    -p 80 \
    $HOME/.irssi/log/<server>/<channel>/<date:%Y-%m-%d>.log
```

---

**Note:** `<server>`, `<channel>` and `<date: ...>` are placeholders.

---

Then go <http://localhost/> in your web browser.

**See also:**

Module `irclog.web.server`

## How to use your WSGI server

---

**Note:** Ignore this if you don't understand what this is.

---

If there's your preferred WSGI server like `mod_wsgi` or `uWSGI`, you can server irclog on your WSGI server.

```
import os
import os.path
import irclog.web

path = os.path.join(os.environ["HOME"],
                    ".irssi/log/<server>/<channel>/<date:%Y-%m-%d>.log")
app = irclog.web.Application(path)
```

**See also:**

Package `irclog.web`



### irclog.messages — Various message types

#### Hierarchy

It provides a hierarchy of IRC log message types.

- *BaseMessage*
  - *Message*
  - *PublicMessage*
  - *ActionMessage*
  - *NoticeMessage*
  - *BaseNickMessage*
  - *NickMessage*
  - *SelfNickMessage*
  - *JoinMessage*
  - *ModeMessage*
  - *PartMessage*
  - *QuitMessage*
  - *KickMessage*
  - *BaseTopicMessage*
  - *TopicMessage*
  - *NoTopicMessage*

## Objects

**class** `irclog.messages.BaseMessage` (*messaged\_at*)

Bases: `object`

Abstract base type for all IRC log messages.

**Parameters** `messaged_at` (`datetime.datetime`) – a `datetime.datetime` logged

**messaged\_at**

The naive `datetime.datetime` logged.

**class** `irclog.messages.Message` (*messaged\_at, nick, line*)

Bases: `irclog.messages.BaseMessage`

Abstract base type for `PublicMessage`, `ActionMessage` and `NoticeMessage`.

**Parameters**

- **messaged\_at** (`datetime.datetime`) – a `datetime.datetime` logged
- **nick** (`basestring`) – a nickname
- **line** (`basestring`) – a message body

**nick**

The nickname.

**line**

The message body.

**class** `irclog.messages.PublicMessage` (*messaged\_at, nick, line*)

Bases: `irclog.messages.Message`

Most commonly used message type.

**Parameters**

- **messaged\_at** (`datetime.datetime`) – a `datetime.datetime` logged
- **nick** (`basestring`) – a nickname
- **line** (`basestring`) – a message body

**class** `irclog.messages.ActionMessage` (*messaged\_at, nick, line*)

Bases: `irclog.messages.Message`

ACTION message type.

**Parameters**

- **messaged\_at** (`datetime.datetime`) – a `datetime.datetime` logged
- **nick** (`basestring`) – a nickname
- **line** (`basestring`) – a message body

**class** `irclog.messages.NoticeMessage` (*messaged\_at, nick, line, channel*)

Bases: `irclog.messages.Message`

Notice message type.

**Parameters**

- **messaged\_at** (`datetime.datetime`) – a `datetime.datetime` logged
- **nick** (`basestring`) – a nickname

- **line** (basestring) – a message body
- **channel** (basestring) – a channel name

**channel**

The channel name.

**class** `irclog.messages.BaseNickMessage` (*messaged\_at, to*)

Bases: `irclog.messages.BaseMessage`

Abstract base type for nickname changing message types.

**Parameters**

- **messaged\_at** (`datetime.datetime`) – a `datetime.datetime` logged
- **to** (basestring) – a new nickname

**to**

The new nickname.

**class** `irclog.messages.NickMessage` (*messaged\_at, from\_, to*)

Bases: `irclog.messages.BaseNickMessage`

Nickname changing message type.

**Parameters**

- **messaged\_at** (`datetime.datetime`) – a `datetime.datetime` logged
- **from** (basestring) – a old nickname
- **to** (basestring) – a new nickname

**from\_**

The old nickname.

**class** `irclog.messages.SelfNickMessage` (*messaged\_at, to*)

Bases: `irclog.messages.BaseNickMessage`

My nickname changing message type.

**Parameters**

- **messaged\_at** (`datetime.datetime`) – a `datetime.datetime` logged
- **to** (basestring) – a new nickname

**class** `irclog.messages.JoinMessage` (*messaged\_at, nick, ident, channel*)

Bases: `irclog.messages.BaseMessage`

Join message type.

**Parameters**

- **messaged\_at** (`datetime.datetime`) – a `datetime.datetime` logged
- **nick** (basestring) – a nickname
- **ident** (basestring) – an ident
- **channel** (basestring) – a channel name

**nick**

The nickname.

**ident**

The ident.

**channel**

The channel name.

**class** `irclog.messages.ModeMessage` (*messaged\_at, server, channel, modelist, nick*)

Bases: `irclog.messages.BaseMessage`

MODE message type.

**Parameters**

- **messaged\_at** (`datetime.datetime`) – a `datetime.datetime` logged
- **server** (`basestring`) – a server
- **channel** (`basestring`) – a channel name
- **modelist** – a mode list
- **nick** (`basestring`) – a nickname

**server**

The server.

**channel**

The channel name.

**modelist**

The mode list.

**nick**

The nickname.

**class** `irclog.messages.PartMessage` (*messaged\_at, nick, ident, channel, reason*)

Bases: `irclog.messages.BaseMessage`

PART message type.

**Parameters**

- **nick** (`basestring`) – a nickname
- **ident** (`basestring`) – an ident
- **channel** (`basestring`) – a channel name
- **reason** (`basestring`) – a reason

**nick**

The nickname.

**ident**

The ident.

**channel**

The channel name.

**reason**

The reason.

**class** `irclog.messages.QuitMessage` (*messaged\_at, nick, ident, reason*)

Bases: `irclog.messages.BaseMessage`

Quitting message type.

**Parameters**

- **nick** (`basestring`) – a nickname

- **ident** (basestring) – an ident
- **reason** (basestring) – a reason

**nick**  
The nickname.

**ident**  
The ident.

**reason**  
The reason.

**class** `irclog.messages.KickMessage` (*messaged\_at, nick, channel, by, reason*)

Bases: `irclog.messages.BaseMessage`

Kick message type.

#### Parameters

- **nick** (basestring) – a nickname that was kicked
- **channel** (basestring) – a channel name
- **by** (basestring) – a nickname that kicks
- **reason** (basestring) – a reason

**nick**  
The nickname that was kicked.

**channel**  
The channel name.

**by**  
The nickname that kicks.

**reason**  
The reason.

**class** `irclog.messages.BaseTopicMessage` (*messaged\_at, nick, channel*)

Bases: `irclog.messages.BaseMessage`

Abstract base class for topic changing message types.

#### Parameters

- **messaged\_at** (`datetime.datetime`) – a `datetime.datetime` logged
- **nick** (basestring) – a nickname
- **channel** (basestring) – a channel name

**nick**  
The nickname.

**channel**  
The channel name.

**class** `irclog.messages.TopicMessage` (*messaged\_at, nick, channel, topic*)

Bases: `irclog.messages.BaseTopicMessage`

Topic changing message types.

#### Parameters

- **messaged\_at** (`datetime.datetime`) – a `datetime.datetime` logged

- **nick** (basestring) – a nickname
- **channel** (basestring) – a channel name
- **topic** (basestring) – a new topic

**topic**

The new topic

**class** `irclog.messages.NoTopicMessage` (*messaged\_at*, *nick*, *channel*)

Bases: `irclog.messages.BaseTopicMessage`

Empty topic message types.

**Parameters**

- **messaged\_at** (`datetime.datetime`) – a `datetime.datetime` logged
- **nick** (basestring) – a nickname
- **channel** (basestring) – a channel name

## irclog.parser — IRC log parser

This module provides a function which takes lines of log then transforms it to message objects in `irclog.messages` module.

`irclog.parser.PATTERN`

The `re` pattern matches to a line of IRC log message.

---

**Note:** This regular expression is originally written by [Kang Seonghoon](#) aka *lifthrasiir*. This pattern is posted in [an article of LangDev](#).

---

`irclog.parser.parse` (*lines*, *date=None*, *encoding='utf-8'*)

Transforms lines of log to message objects in `irclog.messages` module.

**Parameters**

- **lines** (*iterable object*, *file object*) – lines of code
- **date** (`datetime.date`) – a date of the log. default is today
- **encoding** – a text encoding. default is "utf-8"

**Returns** a list of `irclog.messages.BaseMessage` instances

---

**Note:** This is exactly a generator function.

---

`irclog.parser.parser` (*function*)

Registers a parser function.

**Parameters** **function** (*callable object*) – a function parses to register

**Returns** passed function

`irclog.parser.nickmsg` (*when*, *nickfrom*, *nickto*, *\*\*\_*)

Parses `irclog.messages.NickMessage`.

`irclog.parser.selfnickmsg` (*when*, *selfnickto*, *\*\*\_*)

Parses `irclog.messages.SelfNickMessage`.

```
irclog.parser.joinmsg(when, joinnick, joinident, joinchan, **_)
    Parses irclog.messages.JoinMessage.

irclog.parser.modemsg(when, modeserver, modechan, modelist, modenick, **_)
    Parses irclog.messages.ModeMessage.

irclog.parser.partmsg(when, partnick, partident, partchan, partreason, **_)
    Parses irclog.messages.PartMessage.

irclog.parser.quitmsg(when, quitnick, quitident, quitreason, **_)
    Parses irclog.messages.QuitMessage.

irclog.parser.kickmsg(when, kicknick, kickchan, kickby, kickreason, **_)
    Parses irclog.messages.KickMessage.

irclog.parser.topicmsg(when, topicnick, topicchan, topicline, **_)
    Parses irclog.messages.TopicMessage.

irclog.parser.notopicmsg(when, notopicnick, notopicchan, **_)
    Parses irclog.messages.NoTopicMessage.

irclog.parser.pubmsg(when, pubnick, publine, **_)
    Parses irclog.messages.PublicMessage.

irclog.parser.actmsg(when, actnick, actline, **_)
    Parses irclog.messages.ActionMessage.

irclog.parser.noticemsg(when, noticenick, noticechan, noticeline, **_)
    Parses irclog.messages.ActionMessage.
```

## irclog.archive — IRC log archive

**irclog.archive.STRPTIME\_TO\_GLOB**  
The table for translating `strptime()` directives to `glob` patterns.

**irclog.archive.STRPTIME\_DIRECTIVE\_PATTERN**  
The `re` pattern object that matches to `strptime()` directives.

```
>>> fmtstr = "%Y-%m-%d"
>>> STRPTIME_DIRECTIVE_PATTERN.sub(lambda m: STRPTIME_TO_GLOB[m.group(0)],
...                               fmtstr)
'[0-9][0-9][0-9][0-9]-[01][0-9]-[0-3][0-9]'
```

**class** `irclog.archive.FilenamePattern` (*pattern*)

Bases: `object`

The glob-like, but specialized to IRC log files, filename pattern matcher.

```
>>> pattern = FilenamePattern("/logs/<server>"
...                          "</channel>.<date:%Y-%m-%d>.log")
>>> pattern.glob_pattern_string()
'/logs/*/*.[0-9][0-9][0-9][0-9]-[01][0-9]-[0-3][0-9].log'
>>> pattern.glob_pattern_string(server="Freenode")
'/logs/Freenode/*.[0-9][0-9][0-9][0-9]-[01][0-9]-[0-3][0-9].log'
```

**REPLACER\_PATTERN**

The pattern of replacer e.g. `<date:%Y-%m-%d>`, `<channel>`.

```
replacer ::= `<' name [ ':' format ] `>'  
name      ::= /[A-Za-z_]+/  
format    ::= /^[^>]*/
```

### replacers

The list of replacers.

```
>>> pattern = FilenamePattern("/logs/<server>"  
...                               "/<channel>.<date:%Y-%m-%d>.log")  
>>> list(pattern.replacers)  
['server', 'channel', 'date']
```

### replacer\_pairs

The list of replacers' (name, format). (For replacers have no format, (name, None).)

```
>>> pattern = FilenamePattern("/logs/<server>"  
...                               "/<channel>.<date:%Y-%m-%d>.log")  
>>> list(pattern.replacer_pairs)  
[('server', None), ('channel', None), ('date', '%Y-%m-%d')]
```

### replacer\_dict

The dict of replacers.

```
>>> pattern = FilenamePattern("/logs/<server>"  
...                               "/<channel>.<date:%Y-%m-%d>.log")  
>>> pattern.replacer_dict["server"]  
>>> pattern.replacer_dict["date"]  
'%Y-%m-%d'  
>>> pattern.replacer_dict["xyz"]  
Traceback (most recent call last):  
...  
KeyError: 'xyz'
```

### fill\_replacers (replacers, escape=None)

Fills replacers with given values.

```
>>> pattern = FilenamePattern("/logs/<server>"  
...                               "/<channel>.<date:%Y-%m-%d>.log")  
>>> import datetime  
>>> pattern.fill_replacers({"server": "Freenode",  
...                        "channel": "#hongminhee",  
...                        "date": datetime.date(2010, 8, 4)})  
'/logs/Freenode/#hongminhee.2010-08-04.log'
```

When escape function has given, non-replacers are applied into the function:

```
>>> pattern.fill_replacers({"server": "Freenode",  
...                        "channel": "#hongminhee",  
...                        "date": datetime.date(2010, 8, 4)},  
...                        escape=lambda x: x.upper())  
'/LOGS/Freenode/#hongminhee.2010-08-04.LOG'
```

It may raise `KeyError` when too few replacers has given:

```
>>> pattern.fill_replacers({"server": "Freenode"})  
Traceback (most recent call last):
```



```
...
KeyError: 'channel'
```

**Parameters** **replacers** (`dict`, mapping object) – a mapping object of (`replacer_name`, `str_to_replace`)

**Returns** a filled filename

**glob\_pattern\_string** (`**replacers`)

Generates a `glob` pattern string. It takes keyword arguments of `replacers` to fill also.

```
>>> pattern = FilenamePattern("/logs/<server>"
...                               "</channel>.<date:%Y-%m-%d>.log")
>>> pattern.glob_pattern_string()
'/logs/*/*. [0-9] [0-9] [0-9] [0-9]-[01] [0-9]-[0-3] [0-9].log'
>>> pattern.glob_pattern_string(server="Freenode")
'/logs/Freenode/*. [0-9] [0-9] [0-9] [0-9]-[01] [0-9]-[0-3] [0-9].log'
>>> pattern.glob_pattern_string(channel="#hongminhee")
'/logs/*/#hongminhee. [0-9] [0-9] [0-9] [0-9]-[01] [0-9]-[0-3] [0-9].log'
>>> import datetime
>>> pattern.glob_pattern_string(date=datetime.date(2010, 8, 4))
'/logs/*/*.2010-08-04.log'
```

**Parameters** **\*\*replacers** – replacers to fill. keywords go replacer names and values fills them

**Returns** a glob pattern string

**glob** (`**replacers`)

Globs with the pattern.

**Parameters** **\*\*replacers** – replacers to fill. keywords go replacer names and values fills them

**Returns** a `list` of all matched paths

**See also:**

- Module `glob`
- Function `glob.glob()`

**iglob** (`**replacers`)

Case-insensitive version of `glob()`.

**Parameters** **\*\*replacers** – replacers to fill. keywords go replacer names and values fills them

**Returns** a `list` of all matched paths

**See also:**

- Module `glob`
- Function `glob.iglob()`

**re\_pattern\_string** (`**replacers`)

Generates a `re` pattern string. It takes keyword arguments of `replacers` to fill also.

```
>>> import datetime
>>> pattern = FilenamePattern("/<server>/<channel>.<date:%Y%m%d>")
>>> pattern.re_pattern_string(date=datetime.date(2010, 8, 4))
'^\\/(?P<server>.+?)\\/(?P<channel>.+?)\\.20100804$'
```

**Parameters** **\*\*replacers** – replacers to fill. keywords go replacer names and values fills them

**Returns** a `re` pattern string

**re\_pattern** (**\*\*replacers**)

Generates a `re` pattern object. It takes keyword arguments of replacers to fill also.

**Parameters** **\*\*replacers** – replacers to fill. keywords go replacer names and values fills them

**Returns** a `re` pattern

---

**Note:** This method is the same to following code:

```
re.compile(file_pattern.re_pattern_string(**replacers))
```

---

**class** `irclog.archive.BaseArchive`

Bases: `object`

Abstract base class for *Archive*, *Server* and *Channel*.

**ELEMENT\_CLASS**

Should be implemented in the subclass.

**ELEMMENT\_TAG**

Should be implemented in the subclass.

**pattern\_replacers**

Should be implemented in the subclass.

**decode\_element\_key** (*element\_key*)

Decodes an element key string to element key.

**Parameters** **element\_key** (basestring) – an element key string

**Returns** an element key

**encode\_element\_key** (*element\_key*)

Encodes an element key to element key string.

**Parameters** **element\_key** – an element key

**Returns** an element key string

**class** `irclog.archive.Archive` (*pattern*)

Bases: `irclog.archive.BaseArchive`

Log archive.

**Parameters** **pattern** (*FilenamePattern*, basestring) – logs filename pattern e.g. `"/logs/<server>/<channel>.<date:%Y-%m-%d>.log"`

**ELEMENT\_CLASS**

alias of *Server*

**decode\_element\_key** (*element\_key*)

Decodes an element key string to element key.

**Parameters** **element\_key** (basestring) – an element key string

**Returns** an element key

**encode\_element\_key** (*element\_key*)

Encodes an element key to element key string.

**Parameters** **element\_key** – an element key

**Returns** an element key string

**class** `irclog.archive.Server` (*archive, server*)

Bases: `irclog.archive.BaseArchive`

IRC server.

**Parameters**

- **archive** (*Archive*) – an archive
- **server** (basestring) – a server name

**ELEMENT\_CLASS**

alias of *Channel*

**decode\_element\_key** (*element\_key*)

Decodes an element key string to element key.

**Parameters** **element\_key** (basestring) – an element key string

**Returns** an element key

**encode\_element\_key** (*element\_key*)

Encodes an element key to element key string.

**Parameters** **element\_key** – an element key

**Returns** an element key string

**class** `irclog.archive.Channel` (*server, channel*)

Bases: `irclog.archive.BaseArchive`

IRC channel or nick.

**Parameters**

- **server** (*Server*) – a server
- **channel** (basestring) – a channel name or a nick

**ELEMENT\_CLASS**

alias of *Log*

**decode\_element\_key** (*element\_key*)

Decodes an element key string to element key.

**Parameters** **element\_key** (basestring) – an element key string

**Returns** an element key

**encode\_element\_key** (*element\_key*)

Encodes an element key to element key string.

**Parameters** **element\_key** – an element key

**Returns** an element key string

**class** `irclog.archive.Log(channel, date)`

Bases: `object`

IRC log.

**Parameters**

- **channel** (*Channel*) – a channel logged
- **date** (`datetime.date`) – a date logged

**server**

The server.

**archive**

The archive.

**pattern**

The filename pattern.

**yesterday\_log**

The yesterday log of the same channel.

**tomorrow\_log**

The tomorrow log of the same channel.

**filename**

The filename of the log. If not exists, it could be None.

**file\_size**

The size of the log file in bytes.

**is\_logged()**

Returns `True` if the channel of the day has logged.

**Returns** `True` or `False`

## irclog.web — Web IRC log view

**class** `irclog.web.Application(archive, template_path, path_prefix='', encoding='utf-8', debug=False)`

Bases: `object`

WSGI application.

```
import os, os.path
path = os.path.join(
    os.environ["HOME"],
    ".irssi/logs/<server>/<channel>/<date:%Y-%m-%d>.log"
)
app = Application(path)
```

**Parameters**

- **archive** (*Archive*, *FilenamePattern*, basestring) – an archive object or path
- **template\_path** (basestring) – a path of template files
- **path\_prefix** (basestring) – a prefix for the path. default is an empty string

- **encoding** (basestring) – an encoding of response text. default is "utf-8"
- **debug** (bool) – a debug flag. default is False

**STATIC\_PATH\_PATTERN****PATH\_MAP**

The `list:` of tuple contains template name, routing pattern and object getting function.

Form is like following:

```
[("template.html", re.compile("url regex"), lambda app, ...: ...)]
```

**archive**

The *Archive* object.

**template\_path**

The path of template files.

**template\_environment**

The Jinja2 template environment. A `jinja2.Environment` instance.

**path\_prefix**

The prefix of the path.

**encoding**

The encoding of response text.

**debug**

The debug flag.

**classmethod route** (*template\_name, pattern\_string*)

Registers a function as request handler.

```
>>> __tmp__ = Application.PATH_MAP
>>> Application.PATH_MAP = []
>>> Application.PATH_MAP
[]
>>> @Application.route("myhandler.html", r"^/(?P<a>.)/(?P<b>.)/?$")
... def myhandler(app, a, b):
...     return {"a": a, "b": b}
...
>>> Application.PATH_MAP
[('myhandler.html', <_sre.SRE_Pattern object at ...>,
  <function myhandler at ...>)]
>>> Application.PATH_MAP = __tmp__
```

**Parameters**

- **template\_name** (basestring) – a template filename
- **pattern\_string** (basestring) – a pattern of URL to route

**classmethod redirect** (*pattern\_string, status\_code=307*)

Registers a redirection handler.

```
>>> __tmp__ = Application.PATH_MAP
>>> Application.PATH_MAP = []
>>> Application.PATH_MAP
[]
>>> @Application.redirect(r"^/(?P<a>.)/(?P<b>.)/?$", 301)
```

```
... def redirect(app, a, b):
...     return "/url/{0}/{1}".format(a, b)
...
>>> Application.PATH_MAP
[(301, <_sre.SRE_Pattern object at ...>,
  <function redirect at ...>)]
>>> Application.PATH_MAP = __tmp__
```

**Parameters** `pattern_string` (basestring) – a pattern of URL to route

**class** `irclog.web.Environment` (*application*, *environment*)

Bases: `dict`

HTTP request environment. It contains WSGI environment values, and is a subtype of `dict`.

**Parameters**

- **application** (*Application*) – an application
- **environment** (`dict`) – a WSGI environment dictionary

**application**

The *Application* instance.

**app**

Alias of *application*.

**authorization**

The tuple that contains credential pair: user and password.

```
>>> assert isinstance(env, Environment)
>>> env['HTTP_AUTHORIZATION']
'Basic dXNlcjpwYXNz'
>>> env.authorization
('user', 'pass')
```

It is `None` when there's no 'HTTP\_AUTHORIZATION' key:

```
>>> assert isinstance(env2, Environment)
>>> 'HTTP_AUTHORIZATION' in env2
False
>>> print env2.auth
None
```

`irclog.web.setup_template_environment` (*environment*)

Sets up the Jinja2 environment.

**Parameters** `environment` (`jinja2.Environment`) – a Jinja2 environment

`irclog.web.quote_url` (*object*)

Quotes as URL.

```
>>> quote_url(1)
'1'
>>> quote_url("hello world")
'hello%20world'
```

**Parameters** `object` – an object to be quoted

**Returns** a quoted string

`irclog.web.url_for(object)`  
 Makes an URL for the object.

**Parameters** `object` – an object to create URL

**Returns** an URL

## irclog.web.server — Built-in web server

### Command line interface

This module provides a command line interface for running the web server.

```
$ python -m irclog.web.server
Usage: python -m irclog.web.server [options] archive-path

python -m irclog.web.server: error: archive path is required
```

It takes exactly one argument, the path of log archive, and several options following explained:

- H** <host>, **--host** <host>  
 Host to listen. Default: 0.0.0.0.
- p** <port>, **--port** <port>  
 Port to bind. Default: 8080.
- T** <path>, **--template** <path>  
 HTML template path. Default: basic.
- P** <prefix>, **--path-prefix** <prefix>  
 Prefix of the URL path.
- E** <key>=<value>, **--wsgi-environ** <key>=<value>  
 Set WSGI environment. It can be multiple e.g. `-Ea=1 -Eb=2` or `-Ea=b=c=123` (which is equivalent to `-Ea=123 -Eb=123 -Ec=123`).
- d**, **--debug**  
 Debug mode.

### Internals

`irclog.web.server.serve(host, port, app)`  
 Serves a WSGI application.

**Parameters**

- **host** (basestring) – a host to listen
- **port** (int) – a port to listen

## irclog.tests — Unit tests

`irclog.tests.MODULES`  
 The modules to test.

`irclog.tests.suite()`

Adapts doctests to `unittest` interface.

**Returns** an adopted `unittest.TestSuite` object

## irclog.version — Version info

`irclog.version.VERSION_INFO = (0, 2, 0)`

(tuple) The version numbers in tuple e.g. (1, 2, 3).

`irclog.version.VERSION = '0.2.0'`

(basestring) The version string e.g. '1.2.3'.



## CHAPTER 3

---

### About distribution

---

It is written by [Hong Minhee](#). See more other contributors from CONTRIBUTORS file of the source code.

You can check out recent version of the source code from its [Mercurial repository](#):

```
$ hg clone https://bitbucket.org/dahlia/irclog
```

This program is distributed under MIT license.



## CHAPTER 4

---

### Indices and tables

---

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



### i

- `irclog`, [4](#)
- `irclog.archive`, [11](#)
- `irclog.messages`, [5](#)
- `irclog.parser`, [10](#)
- `irclog.tests`, [19](#)
- `irclog.version`, [20](#)
- `irclog.web`, [16](#)
- `irclog.web.server`, [19](#)



## Symbols

- E <key>=<value>, -wsgi-environ <key>=<value>  
python-m-irclog.web.server command line option,  
19
- H <host>, -host <host>  
python-m-irclog.web.server command line option,  
19
- P <prefix>, -path-prefix <prefix>  
python-m-irclog.web.server command line option,  
19
- T <path>, -template <path>  
python-m-irclog.web.server command line option,  
19
- d, -debug  
python-m-irclog.web.server command line option,  
19
- p <port>, -port <port>  
python-m-irclog.web.server command line option,  
19

## A

- ActionMessage (class in irclog.messages), 6
- actmsg() (in module irclog.parser), 11
- app (irclog.web.Environment attribute), 18
- Application (class in irclog.web), 16
- application (irclog.web.Environment attribute), 18
- Application.PATH\_MAP (in module irclog.web), 17
- Application.STATIC\_PATH\_PATTERN (in module irclog.web), 17
- Archive (class in irclog.archive), 14
- archive (irclog.archive.Log attribute), 16
- archive (irclog.web.Application attribute), 17
- authorization (irclog.web.Environment attribute), 18

## B

- BaseArchive (class in irclog.archive), 14
- BaseArchive.ELEMENT\_CLASS (in module irclog.archive), 14
- BaseArchive.ELEMENT\_TAG (in module irclog.archive), 14

- BaseMessage (class in irclog.messages), 6
- BaseNickMessage (class in irclog.messages), 7
- BaseTopicMessage (class in irclog.messages), 9
- by (irclog.messages.KickMessage attribute), 9

## C

- Channel (class in irclog.archive), 15
- channel (irclog.messages.BaseTopicMessage attribute), 9
- channel (irclog.messages.JoinMessage attribute), 7
- channel (irclog.messages.KickMessage attribute), 9
- channel (irclog.messages.ModeMessage attribute), 8
- channel (irclog.messages.NoticeMessage attribute), 7
- channel (irclog.messages.PartMessage attribute), 8

## D

- debug (irclog.web.Application attribute), 17
- decode\_element\_key() (irclog.archive.Archive method), 14
- decode\_element\_key() (irclog.archive.BaseArchive method), 14
- decode\_element\_key() (irclog.archive.Channel method), 15
- decode\_element\_key() (irclog.archive.Server method), 15

## E

- ELEMENT\_CLASS (irclog.archive.Archive attribute), 14
- ELEMENT\_CLASS (irclog.archive.Channel attribute), 15
- ELEMENT\_CLASS (irclog.archive.Server attribute), 15
- encode\_element\_key() (irclog.archive.Archive method), 15
- encode\_element\_key() (irclog.archive.BaseArchive method), 14
- encode\_element\_key() (irclog.archive.Channel method), 15
- encode\_element\_key() (irclog.archive.Server method), 15
- encoding (irclog.web.Application attribute), 17
- Environment (class in irclog.web), 18

## F

file\_size (irclog.archive.Log attribute), 16  
filename (irclog.archive.Log attribute), 16  
FilenamePattern (class in irclog.archive), 11  
FilenamePattern.REPLACER\_PATTERN (in module irclog.archive), 11  
fill\_replacers() (irclog.archive FilenamePattern method), 12  
from\_ (irclog.messages.NickMessage attribute), 7

## G

glob() (irclog.archive FilenamePattern method), 13  
glob\_pattern\_string() (irclog.archive FilenamePattern method), 13

## I

ident (irclog.messages.JoinMessage attribute), 7  
ident (irclog.messages.PartMessage attribute), 8  
ident (irclog.messages.QuitMessage attribute), 9  
iglob() (irclog.archive FilenamePattern method), 13  
irclog (module), 4  
irclog.archive (module), 11  
irclog.messages (module), 5  
irclog.parser (module), 10  
irclog.tests (module), 19  
irclog.version (module), 20  
irclog.web (module), 16  
irclog.web.server (module), 19  
is\_logged() (irclog.archive.Log method), 16

## J

JoinMessage (class in irclog.messages), 7  
joinmsg() (in module irclog.parser), 10

## K

KickMessage (class in irclog.messages), 9  
kickmsg() (in module irclog.parser), 11

## L

line (irclog.messages.Message attribute), 6  
Log (class in irclog.archive), 16

## M

Message (class in irclog.messages), 6  
messaged\_at (irclog.messages.BaseMessage attribute), 6  
modelist (irclog.messages.ModeMessage attribute), 8  
ModeMessage (class in irclog.messages), 8  
modemsg() (in module irclog.parser), 11  
MODULES (in module irclog.tests), 19

## N

nick (irclog.messages.BaseTopicMessage attribute), 9  
nick (irclog.messages.JoinMessage attribute), 7

nick (irclog.messages.KickMessage attribute), 9  
nick (irclog.messages.Message attribute), 6  
nick (irclog.messages.ModeMessage attribute), 8  
nick (irclog.messages.PartMessage attribute), 8  
nick (irclog.messages.QuitMessage attribute), 9  
NickMessage (class in irclog.messages), 7  
nickmsg() (in module irclog.parser), 10  
NoticeMessage (class in irclog.messages), 6  
noticemsg() (in module irclog.parser), 11  
NoTopicMessage (class in irclog.messages), 10  
notopicmsg() (in module irclog.parser), 11

## P

parse() (in module irclog.parser), 10  
parser() (in module irclog.parser), 10  
PartMessage (class in irclog.messages), 8  
partmsg() (in module irclog.parser), 11  
path\_prefix (irclog.web.Application attribute), 17  
PATTERN (in module irclog.parser), 10  
pattern (irclog.archive.Log attribute), 16  
pattern\_replacers (irclog.archive.BaseArchive attribute), 14  
PublicMessage (class in irclog.messages), 6  
pubmsg() (in module irclog.parser), 11  
python-m-irclog.web.server command line option  
-E <key>=<value>, --wsgi-environ <key>=<value>, 19  
-H <host>, --host <host>, 19  
-P <prefix>, --path-prefix <prefix>, 19  
-T <path>, --template <path>, 19  
-d, --debug, 19  
-p <port>, --port <port>, 19

## Q

QuitMessage (class in irclog.messages), 8  
quitmsg() (in module irclog.parser), 11  
quote\_url() (in module irclog.web), 18

## R

re\_pattern() (irclog.archive FilenamePattern method), 14  
re\_pattern\_string() (irclog.archive FilenamePattern method), 13  
reason (irclog.messages.KickMessage attribute), 9  
reason (irclog.messages.PartMessage attribute), 8  
reason (irclog.messages.QuitMessage attribute), 9  
redirect() (irclog.web.Application class method), 17  
replacer\_dict (irclog.archive FilenamePattern attribute), 12  
replacer\_pairs (irclog.archive FilenamePattern attribute), 12  
replacers (irclog.archive FilenamePattern attribute), 12  
route() (irclog.web.Application class method), 17



## S

SelfNickMessage (class in `irclog.messages`), 7  
selfnickmsg() (in module `irclog.parser`), 10  
serve() (in module `irclog.web.server`), 19  
Server (class in `irclog.archive`), 15  
server (`irclog.archive.Log` attribute), 16  
server (`irclog.messages.ModeMessage` attribute), 8  
setup\_template\_environment() (in module `irclog.web`), 18  
STRPTIME\_DIRECTIVE\_PATTERN (in module `irclog.archive`), 11  
STRPTIME\_TO\_GLOB (in module `irclog.archive`), 11  
suite() (in module `irclog.tests`), 19

## T

template\_environment (`irclog.web.Application` attribute), 17  
template\_path (`irclog.web.Application` attribute), 17  
to (`irclog.messages.BaseNickMessage` attribute), 7  
tomorrow\_log (`irclog.archive.Log` attribute), 16  
topic (`irclog.messages.TopicMessage` attribute), 10  
TopicMessage (class in `irclog.messages`), 9  
topicmsg() (in module `irclog.parser`), 11

## U

url\_for() (in module `irclog.web`), 19

## V

VERSION (in module `irclog.version`), 20  
VERSION\_INFO (in module `irclog.version`), 20

## Y

yesterday\_log (`irclog.archive.Log` attribute), 16