

syl

Generated by Doxygen 1.8.11

Contents

1	Namespace Index	1
1.1	Packages	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Namespace Documentation	9
5.1	app Namespace Reference	9
5.1.1	Function Documentation	9
5.1.1.1	client(_ip, _port)	9
5.1.1.2	server(_port)	9
5.1.2	Variable Documentation	10
5.1.2.1	ERROR_MSG	10
5.2	lib Namespace Reference	10
5.3	lib.client Namespace Reference	10
5.3.1	Function Documentation	11
5.3.1.1	end_client(_client)	11
5.4	lib.files_db Namespace Reference	11
5.4.1	Function Documentation	11
5.4.1.1	dict_factory(cursor, row)	11
5.5	lib.log Namespace Reference	12
5.6	lib.msg Namespace Reference	12
5.7	lib.server Namespace Reference	12
5.8	lib.user Namespace Reference	12
5.9	lib.window Namespace Reference	12

6	Class Documentation	13
6.1	lib.client.Client Class Reference	13
6.1.1	Detailed Description	13
6.1.2	Constructor & Destructor Documentation	14
6.1.2.1	__init__(self, _ip_addr, _port)	14
6.1.3	Member Function Documentation	14
6.1.3.1	listen_server(self)	14
6.1.3.2	send_server(self)	15
6.1.4	Member Data Documentation	15
6.1.4.1	ip_addr	15
6.1.4.2	msg	15
6.1.4.3	port	15
6.1.4.4	sock	15
6.1.4.5	thread_listen	16
6.1.4.6	thread_send	16
6.1.4.7	window	16
6.2	lib.files_db.Files_DB Class Reference	16
6.2.1	Detailed Description	16
6.2.2	Member Function Documentation	16
6.2.2.1	create(self, _db_name=""server"")	16
6.2.2.2	delete_files_by_group(self, _group, _db_name=""server"")	17
6.2.2.3	file_by_name_group(self, _file_name, _group, _db_name=""server"")	17
6.2.2.4	file_names_by_group(self, _group, _db_name=""server"")	17
6.2.2.5	insert_file(self, _group, _file_name, _file, _db_name=""server"")	18
6.3	lib.log.Log Class Reference	18
6.3.1	Detailed Description	19
6.3.2	Constructor & Destructor Documentation	19
6.3.2.1	__init__(self, _type)	19
6.3.3	Member Function Documentation	19
6.3.3.1	reg(self, _reg)	19

6.3.4	Member Data Documentation	19
6.3.4.1	time	19
6.3.4.2	type	20
6.4	lib.msg.Msg Class Reference	20
6.4.1	Detailed Description	20
6.4.2	Constructor & Destructor Documentation	20
6.4.2.1	__init__(self, _type, _log)	20
6.4.3	Member Function Documentation	21
6.4.3.1	create(self, _type, _from, _to, _group, _text, _file)	21
6.4.3.2	receive(self, _client)	21
6.4.3.3	receive_all(self, _client, _size)	22
6.4.3.4	send(self, _client, _msg)	22
6.4.4	Member Data Documentation	22
6.4.4.1	data	22
6.4.4.2	log	22
6.5	object Class Reference	23
6.6	lib.server.Server Class Reference	23
6.6.1	Detailed Description	24
6.6.2	Constructor & Destructor Documentation	24
6.6.2.1	__init__(self, _ip_addr, _port)	24
6.6.3	Member Function Documentation	25
6.6.3.1	com_away(self, _data, _user, _full_msg={})	25
6.6.3.2	com_ban(self, _data, _user, _full_msg={})	25
6.6.3.3	com_clear(self, _data, _user, _full_msg={})	25
6.6.3.4	com_create(self, _data, _user, _full_msg={})	26
6.6.3.5	com_delete(self, _data, _user, _full_msg={})	26
6.6.3.6	com_error(self, _data, _user, _full_msg={})	26
6.6.3.7	com_file(self, _data, _user, _full_msg)	27
6.6.3.8	com_get_file(self, _data, _user, _full_msg={})	27
6.6.3.9	com_help(self, _data, _user, _full_msg={})	27

6.6.3.10	<code>com_join(self, _data, _user, _full_msg={})</code>	28
6.6.3.11	<code>com_kick(self, _data, _user, _full_msg={})</code>	28
6.6.3.12	<code>com_leave(self, _data, _user, _full_msg={})</code>	29
6.6.3.13	<code>com_list(self, _data, _user, _full_msg={})</code>	29
6.6.3.14	<code>com_list_files(self, _data, _user, _full_msg={})</code>	29
6.6.3.15	<code>com_msg(self, _data, _user, _full_msg={})</code>	30
6.6.3.16	<code>com_nick(self, _data, _user, _full_msg={})</code>	30
6.6.3.17	<code>list_client(self, _user)</code>	30
6.6.3.18	<code>listen(self)</code>	31
6.6.3.19	<code>manage_coms(self, _data, _user)</code>	31
6.6.3.20	<code>msg_of_day(self, _user)</code>	31
6.6.3.21	<code>send_to_all(self, _user, _msg)</code>	32
6.6.3.22	<code>server_send(self, _user, _text)</code>	32
6.6.4	Member Data Documentation	32
6.6.4.1	<code>coms_list</code>	32
6.6.4.2	<code>groups_dict</code>	33
6.6.4.3	<code>ip_addr</code>	33
6.6.4.4	<code>last_user_key</code>	33
6.6.4.5	<code>msg</code>	33
6.6.4.6	<code>port</code>	33
6.6.4.7	<code>sock</code>	33
6.6.4.8	<code>trds_dict</code>	33
6.6.4.9	<code>users_list</code>	33
6.7	<code>lib.user.User</code> Class Reference	33
6.7.1	Detailed Description	34
6.7.2	Constructor & Destructor Documentation	34
6.7.2.1	<code>__init__(self, _key, _client, _addr, _nick, _group)</code>	34
6.7.3	Member Data Documentation	34
6.7.3.1	<code>addr</code>	34
6.7.3.2	<code>away</code>	35

6.7.3.3	client	35
6.7.3.4	group	35
6.7.3.5	key	35
6.7.3.6	nick	35
6.8	lib.window.Window Class Reference	35
6.8.1	Detailed Description	36
6.8.2	Constructor & Destructor Documentation	36
6.8.2.1	__init__(self)	36
6.8.3	Member Function Documentation	36
6.8.3.1	addstr_data(self, _str, _type)	36
6.8.3.2	addstr_header(self, _str)	37
6.8.3.3	clear_data(self)	37
6.8.3.4	clear_header(self)	37
6.8.3.5	close(self)	38
6.8.3.6	start(self, std)	38
6.8.4	Member Data Documentation	38
6.8.4.1	data	38
6.8.4.2	error_color	38
6.8.4.3	header	38
6.8.4.4	header_color	39
6.8.4.5	input_color	39
6.8.4.6	private_color	39
6.8.4.7	prompt	39
6.8.4.8	server_color	39
6.8.4.9	std	39
6.8.4.10	tex	39
6.8.4.11	text_color	39
7	File Documentation	41
7.1	app.py File Reference	41
7.2	lib/__init__.py File Reference	41
7.3	lib/client.py File Reference	41
7.4	lib/files_db.py File Reference	42
7.5	lib/log.py File Reference	42
7.6	lib/msg.py File Reference	42
7.7	lib/server.py File Reference	43
7.8	lib/user.py File Reference	43
7.9	lib/window.py File Reference	43
	Index	45

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

app	9
lib	10
lib.client	10
lib.files_db	11
lib.log	12
lib.msg	12
lib.server	12
lib.user	12
lib.window	12

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

object	23
lib.client.Client	13
lib.files_db.Files_DB	16
lib.log.Log	18
lib.msg.Msg	20
lib.server.Server	23
lib.user.User	33
lib.window.Window	35

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

lib.client.Client	
Class for client application mode	13
lib.files_db.Files_DB	
Class for files db	16
lib.log.Log	
Class for log	18
lib.msg.Msg	
Class for transit of messages	20
object	23
lib.server.Server	
Class for server application mode	23
lib.user.User	
Create to organize users information and apart this from server module	33
lib.window.Window	
Class for window control using Curses	35

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

app.py	41
lib/__init__.py	41
lib/client.py	41
lib/files_db.py	42
lib/log.py	42
lib/msg.py	42
lib/server.py	43
lib/user.py	43
lib/window.py	43

Chapter 5

Namespace Documentation

5.1 app Namespace Reference

Functions

- def `server` (`_port`)
Starts app as server.
- def `client` (`_ip`, `_port`)
Starts app as client.

Variables

- string `ERROR_MSG` = "Is necessary use arguments to inicialize this application\n"

5.1.1 Function Documentation

5.1.1.1 `def app.client (_ip, _port)`

Starts app as client.

Parameters

<code>_port</code>	The port
--------------------	----------

Returns

return one client

Definition at line 33 of file app.py.

5.1.1.2 `def app.server (_port)`

Starts app as server.

Parameters

<code>_port</code>	The port
--------------------	----------

Returns

return one server

Definition at line 20 of file app.py.

5.1.2 Variable Documentation

5.1.2.1 `string app.ERROR_MSG = "Is necessary use arguments to inicialize this application\n"`

Definition at line 44 of file app.py.

5.2 lib Namespace Reference

Namespaces

- [client](#)
- [files_db](#)
- [log](#)
- [msg](#)
- [server](#)
- [user](#)
- [window](#)

5.3 lib.client Namespace Reference

Classes

- class [Client](#)
Class for client application mode.

Functions

- def [end_client](#) (`_client`)
Ends a client.

5.3.1 Function Documentation

5.3.1.1 def lib.client.end_client (_client)

Ends a client.

Method isn't used.

```
@param      _client  The client
@return     void method.
```

Definition at line 164 of file client.py.

5.4 lib.files_db Namespace Reference

Classes

- class [Files_DB](#)
Class for files db.

Functions

- def [dict_factory](#) (cursor, row)
Create a fatory to convert result of a query in a dictionary.

5.4.1 Function Documentation

5.4.1.1 def lib.files_db.dict_factory (cursor, row)

Create a fatory to convert result of a query in a dictionary.

This is used in 'file_names_by_group', using row_factory attribute from a connection of sqlite3

Parameters

<i>cursor</i>	The cursor
<i>row</i>	The row

Returns

Return a dictionary from a query.

Definition at line 160 of file files_db.py.

5.5 lib.log Namespace Reference

Classes

- class [Log](#)
Class for log.

5.6 lib.msg Namespace Reference

Classes

- class [Msg](#)
Class for transit of messages.

5.7 lib.server Namespace Reference

Classes

- class [Server](#)
Class for server application mode.

5.8 lib.user Namespace Reference

Classes

- class [User](#)
Create to organize users information and apart this from server module.

5.9 lib.window Namespace Reference

Classes

- class [Window](#)
Class for window control using Curses.

Chapter 6

Class Documentation

6.1 lib.client.Client Class Reference

Class for client application mode.

Inherits [object](#).

Public Member Functions

- def [__init__](#) (self, _ip_addr, _port)
- def [listen_server](#) (self)
method started as thread to read/listen server.
- def [send_server](#) (self)
method started as thread to send data to server.

Public Attributes

- [ip_addr](#)
- [port](#)
- [sock](#)
- [window](#)
- [msg](#)
- [thread_listen](#)
- [thread_send](#)

6.1.1 Detailed Description

Class for client application mode.

Definition at line 21 of file client.py.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 `def lib.client.Client.__init__(self, _ip_addr, _port)`

Definition at line 23 of file client.py.

6.1.3 Member Function Documentation

6.1.3.1 `def lib.client.Client.listen_server (self)`

method started as thread to read/listen server.

Parameters

<i>self</i>	The object
-------------	------------

Returns

void method.

Definition at line 47 of file client.py.

6.1.3.2 def lib.client.Client.send_server (*self*)

method started as thread to send data to server.

Parameters

<i>self</i>	The object
-------------	------------

Returns

void method.

Definition at line 112 of file client.py.

6.1.4 Member Data Documentation**6.1.4.1 lib.client.Client.ip_addr**

Definition at line 25 of file client.py.

6.1.4.2 lib.client.Client.msg

Definition at line 32 of file client.py.

6.1.4.3 lib.client.Client.port

Definition at line 26 of file client.py.

6.1.4.4 lib.client.Client.sock

Definition at line 27 of file client.py.

6.1.4.5 lib.client.Client.thread_listen

Definition at line 33 of file client.py.

6.1.4.6 lib.client.Client.thread_send

Definition at line 34 of file client.py.

6.1.4.7 lib.client.Client.window

Definition at line 29 of file client.py.

The documentation for this class was generated from the following file:

- lib/[client.py](#)

6.2 lib.files_db.Files_DB Class Reference

Class for files db.

Inherits [object](#).

Public Member Functions

- def [create](#) (self, _db_name="server")
Create a database, if already exist data from another old server it's excluded.
- def [insert_file](#) (self, _group, _file_name, _file, _db_name="server")
Insert a file in database, the file is a Base 64 encoded text.
- def [file_names_by_group](#) (self, _group, _db_name="server")
List files stored in a group.
- def [file_by_name_group](#) (self, _file_name, _group, _db_name="server")
Find file by name and group, this is a key for a file.
- def [delete_files_by_group](#) (self, _group, _db_name="server")
delete files stored in a group

6.2.1 Detailed Description

Class for files db.

Definition at line 12 of file files_db.py.

6.2.2 Member Function Documentation

6.2.2.1 def lib.files_db.Files_DB.create (self, _db_name = "server")

Create a database, if already exist data from another old server it's excluded.

Parameters

<i>self</i>	The object
<i>_db_name</i>	The database name

Returns

return True if success.

Definition at line 24 of file files_db.py.

```
6.2.2.2 def lib.files_db.Files_DB.delete_files_by_group ( self, _group, _db_name = "server" )
```

delete files stored in a group

Parameters

<i>self</i>	The object
<i>_group</i>	The group
<i>_db_name</i>	The database name

Returns

Return True if files was deleted

Definition at line 134 of file files_db.py.

```
6.2.2.3 def lib.files_db.Files_DB.file_by_name_group ( self, _file_name, _group, _db_name = "server" )
```

Find file by name and group, this is a key for a file.

Parameters

<i>self</i>	The object
<i>_file_name</i>	The file name
<i>_group</i>	The group
<i>_db_name</i>	The database name

Returns

return a Base 64 file.

Definition at line 108 of file files_db.py.

```
6.2.2.4 def lib.files_db.Files_DB.file_names_by_group ( self, _group, _db_name = "server" )
```

List files stored in a group.

Parameters

<i>self</i>	The object
<i>_group</i>	The group
<i>_db_name</i>	The database name

Returns

Return a list of dictionaries with file names.

Definition at line 81 of file files_db.py.

6.2.2.5 `def lib.files_db.Files_DB.insert_file (self, _group, _file_name, _file, _db_name = "server")`

Insert a file in database, the file is a Base 64 encoded text.

Parameters

<i>self</i>	The object
<i>_group</i>	The group
<i>_file_name</i>	The file name
<i>_file</i>	The file
<i>_db_name</i>	The database name

Returns

Return True if file was inserted on database

Definition at line 54 of file files_db.py.

The documentation for this class was generated from the following file:

- [lib/files_db.py](#)

6.3 lib.log.Log Class Reference

Class for log.

Inherits [object](#).

Public Member Functions

- `def __init__ (self, _type)`
Constructs the object.
- `def reg (self, _reg)`
Method to register each entry of log.

Public Attributes

- [type](#)
- [time](#)

6.3.1 Detailed Description

Class for log.

Definition at line 13 of file log.py.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 def lib.log.Log.__init__(self, _type)

Constructs the object.

Receive _type to use in file name.

```
@param      self    The object
@param      _type   The type
```

Definition at line 22 of file log.py.

6.3.3 Member Function Documentation

6.3.3.1 def lib.log.Log.reg (self, _reg)

Method to register each entry of log.

Parameters

<i>self</i>	The object
<i>_reg</i>	The register

Returns

This is a void method, only writes a entry in log file.

Definition at line 38 of file log.py.

6.3.4 Member Data Documentation

6.3.4.1 lib.log.Log.time

Definition at line 24 of file log.py.

6.3.4.2 lib.log.Log.type

Definition at line 23 of file log.py.

The documentation for this class was generated from the following file:

- [lib/log.py](#)

6.4 lib.msg.Msg Class Reference

Class for transit of messages.

Inherits [object](#).

Public Member Functions

- `def __init__(self, _type, _log)`
Constructs the object.
- `def create(self, _type, _from, _to, _group, _text, _file)`
Un used method, can be used to create message with time attribute.
- `def send(self, _client, _msg)`
This method receives a message and a socket.
- `def receive(self, _client)`
This method receives a socket.
- `def receive_all(self, _client, _size)`
Support method for 'receive' method, works receiving from socket a message by size.

Public Attributes

- [log](#)
- [data](#)

6.4.1 Detailed Description

Class for transit of messages.

This class structs each message in packs and make log call.

Definition at line 17 of file msg.py.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 `def lib.msg.Msg.__init__(self, _type, _log)`

Constructs the object.

Parameters

<i>self</i>	The object
<i>_type</i>	The type
<i>_log</i>	The log

Definition at line 27 of file msg.py.

6.4.3 Member Function Documentation

6.4.3.1 `def lib.msg.Msg.create (self, _type, _from, _to, _group, _text, _file)`

Un used method, can be used to create message with time attribute.

Parameters

<i>self</i>	The object
<i>_type</i>	The type
<i>_from</i>	The from
<i>_to</i>	The to (message receiver)
<i>_group</i>	The group
<i>_text</i>	The text
<i>_file</i>	The file

Returns

changes self.data dictionary of class.

Definition at line 49 of file msg.py.

6.4.3.2 `def lib.msg.Msg.receive (self, _client)`

This method receives a socket.

Read 4 first bytes with size of message. After that receive the real message.

Parameters

<i>self</i>	The object
<i>_client</i>	The client (socket)

Returns

return a dictionary with message attributes

Definition at line 97 of file msg.py.

6.4.3.3 `def lib.msg.Msg.receive_all (self, _client, _size)`

Support method for 'receive' method, works receiving from socket a message by size.

Parameters

<i>self</i>	The object
<i>_client</i>	The client (socket)
<i>_size</i>	The size

Returns

return received data.

Definition at line 123 of file msg.py.

6.4.3.4 `def lib.msg.Msg.send (self, _client, _msg)`

This method receives a message and a socket.

Structs this message with a 4 bytes in benning, this bytes contains the size of message.

Parameters

<i>self</i>	The object
<i>_client</i>	The client (socket)
<i>_msg</i>	The message

Returns

void method, only sends a message.

Definition at line 74 of file msg.py.

6.4.4 Member Data Documentation

6.4.4.1 `lib.msg.Msg.data`

Definition at line 32 of file msg.py.

6.4.4.2 `lib.msg.Msg.log`

Definition at line 29 of file msg.py.

The documentation for this class was generated from the following file:

- [lib/msg.py](#)

6.5 object Class Reference

Inherited by [lib.client.Client](#), [lib.files_db.Files_DB](#), [lib.log.Log](#), [lib.msg.Msg](#), [lib.server.Server](#), [lib.user.User](#), and [lib.window.Window](#).

The documentation for this class was generated from the following file:

- [lib/user.py](#)

6.6 lib.server.Server Class Reference

Class for server application mode.

Inherits [object](#).

Public Member Functions

- `def __init__(self, _ip_addr, _port)`
Constructs the object.
- `def listen(self)`
When [app.py](#) instances this class this method is called, this method maintain an infinity loop listen socket.
- `def list_client(self, _user)`
When listen method find new user this method is call as a thread, this method manage the communication with users.
- `def send_to_all(self, _user, _msg)`
Sends message to all users in same group as sender.
- `def server_send(self, _user, _text)`
Support method to send message as server for one user.
- `def msg_of_day(self, _user)`
Send to user a server random initial message from a text file.
- `def manage_coms(self, _data, _user)`
works like a switch case for commands in server.
- `def com_error(self, _data, _user, _full_msg={})`
in case of users call a nonexistent function this method returns a message informing user.
- `def com_away(self, _data, _user, _full_msg={})`
set user status to away, in this status user can't receive private messages.
- `def com_ban(self, _data, _user, _full_msg={})`
This method banish a user from a group.
- `def com_clear(self, _data, _user, _full_msg={})`
clear client window, this could be implemented only on clien side but to store a log this is managed by server.
- `def com_create(self, _data, _user, _full_msg={})`
create a group
- `def com_delete(self, _data, _user, _full_msg={})`
Delete group from server.
- `def com_file(self, _data, _user, _full_msg)`
receives a file send by user and store in database
- `def com_get_file(self, _data, _user, _full_msg={})`
get a file stored in user group.
- `def com_help(self, _data, _user, _full_msg={})`

- send a list of commands and description to user.*
- def `com_join` (self, _data, _user, _full_msg={})
join a group.
- def `com_kick` (self, _data, _user, _full_msg={})
kick user from group.
- def `com_leave` (self, _data, _user, _full_msg={})
leave a group or a chat if user not in a group.
- def `com_list` (self, _data, _user, _full_msg={})
list server groups if user isn't in a group or list users of user group.
- def `com_list_files` (self, _data, _user, _full_msg={})
Lists files stored on user group.
- def `com_msg` (self, _data, _user, _full_msg={})
send a private message from one user to another.
- def `com_nick` (self, _data, _user, _full_msg={})
changes user nickname.

Public Attributes

- `ip_addr`
- `port`
- `sock`
- `coms_list`
- `users_list`
- `groups_dict`
- `last_user_key`
- `msg`
- `trds_dict`

6.6.1 Detailed Description

Class for server application mode.

Definition at line 18 of file server.py.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 def lib.server.Server.__init__(self, _ip_addr, _port)

Constructs the object.

Parameters

<i>self</i>	The object
<i>_ip_addr</i>	The ip address
<i>_port</i>	The port

Definition at line 28 of file server.py.

6.6.3 Member Function Documentation

6.6.3.1 `def lib.server.Server.com_away (self, _data, _user, _full_msg = { })`

set user status to away, in this status user can't receive private messages.

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user
<i>_full_msg</i>	The full message

Returns

void method.

Definition at line 266 of file server.py.

6.6.3.2 `def lib.server.Server.com_ban (self, _data, _user, _full_msg = { })`

This method banish a user from a group.

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user
<i>_full_msg</i>	The full message

Returns

void method.

Definition at line 317 of file server.py.

6.6.3.3 `def lib.server.Server.com_clear (self, _data, _user, _full_msg = { })`

clear client window, this could be implemented only on clien side but to store a log this is managed by server.

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user
<i>_full_msg</i>	The full message

Returns

void method.

Definition at line 390 of file server.py.

6.6.3.4 `def lib.server.Server.com_create (self, _data, _user, _full_msg = { })`

create a group

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user
<i>_full_msg</i>	The full message

Returns

void method.

Definition at line 421 of file server.py.

6.6.3.5 `def lib.server.Server.com_delete (self, _data, _user, _full_msg = { })`

Delete group from server.

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user
<i>_full_msg</i>	The full message

Returns

void method.

Definition at line 480 of file server.py.

6.6.3.6 `def lib.server.Server.com_error (self, _data, _user, _full_msg = { })`

in case of users call a nonexistent function this method returns a message informing user.

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user
<i>_full_msg</i>	The full message

Returns

void method.

Definition at line 250 of file server.py.

6.6.3.7 `def lib.server.Server.com_file (self, _data, _user, _full_msg)`

receives a file send by user and store in database

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user
<i>_full_msg</i>	The full message

Returns

void method.

Definition at line 530 of file server.py.

6.6.3.8 `def lib.server.Server.com_get_file (self, _data, _user, _full_msg = { })`

get a file stored in user group.

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user
<i>_full_msg</i>	The full message

Returns

void method.

Definition at line 583 of file server.py.

6.6.3.9 `def lib.server.Server.com_help (self, _data, _user, _full_msg = { })`

send a list of commands and description to user.

Every method in 'coms_list' if receives 'description' as '_data' returns a string with command description.

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user
<i>_full_msg</i>	The full message

Returns

void method.

Definition at line 630 of file server.py.

```
6.6.3.10 def lib.server.Server.com_join ( self, _data, _user, _full_msg = { } )
```

join a group.

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user
<i>_full_msg</i>	The full message

Returns

void method.

Definition at line 664 of file server.py.

```
6.6.3.11 def lib.server.Server.com_kick ( self, _data, _user, _full_msg = { } )
```

kick user from group.

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user
<i>_full_msg</i>	The full message

Returns

void method.

Definition at line 715 of file server.py.

6.6.3.12 `def lib.server.Server.com_leave (self, _data, _user, _full_msg = { })`

leave a group or a chat if user not in a group.

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user
<i>_full_msg</i>	The full message

Returns

void method.

Definition at line 776 of file server.py.

6.6.3.13 `def lib.server.Server.com_list (self, _data, _user, _full_msg = { })`

list server groups if user isn't in a group or list users of user group.

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user
<i>_full_msg</i>	The full message

Returns

void method.

Definition at line 831 of file server.py.

6.6.3.14 `def lib.server.Server.com_list_files (self, _data, _user, _full_msg = { })`

Lists files stored on user group.

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user
<i>_full_msg</i>	The full message

Returns

void method.

Definition at line 875 of file server.py.

```
6.6.3.15 def lib.server.Server.com_msg ( self, _data, _user, _full_msg = { } )
```

send a private message from one user to another.

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user
<i>_full_msg</i>	The full message

Returns

void method.

Definition at line 912 of file server.py.

```
6.6.3.16 def lib.server.Server.com_nick ( self, _data, _user, _full_msg = { } )
```

changes user nickname.

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user
<i>_full_msg</i>	The full message

Returns

void method.

Definition at line 964 of file server.py.

```
6.6.3.17 def lib.server.Server.list_client ( self, _user )
```

When listen method find new user this method is call as a thread, this method manage the communication with users.

Parameters

<i>self</i>	The object
<i>_user</i>	The user

Returns

void method.

Definition at line 103 of file server.py.

6.6.3.18 `def lib.server.Server.listen (self)`

When [app.py](#) instances this class this method is called, this method maintain an infinity loop listen socket.

Parameters

<i>self</i>	The object
-------------	------------

Returns

void method.

Definition at line 68 of file server.py.

6.6.3.19 `def lib.server.Server.manage_coms (self, _data, _user)`

works like a switch case for commands in server.

Verify wich function user want and call right method 'com_*' to manage this.

Parameters

<i>self</i>	The object
<i>_data</i>	The data
<i>_user</i>	The user

Returns

{ description_of_the_return_value }

Definition at line 229 of file server.py.

6.6.3.20 `def lib.server.Server.msg_of_day (self, _user)`

Send to user a server random initial message from a text file.

This call 'com_help' method too.

```
@param      self      The object
@param      _user     The user

@return     void method.
```

Definition at line 198 of file server.py.

6.6.3.21 `def lib.server.Server.send_to_all (self, _user, _msg)`

Sends message to all users in same group as sender.

Parameters

<i>self</i>	The object
<i>_user</i>	The user
<i>_msg</i>	The message

Returns

void method.

Definition at line 157 of file server.py.

6.6.3.22 `def lib.server.Server.server_send (self, _user, _text)`

Support method to send message as server for one user.

Parameters

<i>self</i>	The object
<i>_user</i>	The user
<i>_text</i>	The text

Returns

void method.

Definition at line 179 of file server.py.

6.6.4 Member Data Documentation

6.6.4.1 `lib.server.Server.coms_list`

Definition at line 35 of file server.py.

6.6.4.2 lib.server.Server.groups_dict

Definition at line 53 of file server.py.

6.6.4.3 lib.server.Server.ip_addr

Definition at line 29 of file server.py.

6.6.4.4 lib.server.Server.last_user_key

Definition at line 54 of file server.py.

6.6.4.5 lib.server.Server.msg

Definition at line 55 of file server.py.

6.6.4.6 lib.server.Server.port

Definition at line 30 of file server.py.

6.6.4.7 lib.server.Server.sock

Definition at line 31 of file server.py.

6.6.4.8 lib.server.Server.trds_dict

Definition at line 57 of file server.py.

6.6.4.9 lib.server.Server.users_list

Definition at line 52 of file server.py.

The documentation for this class was generated from the following file:

- [lib/server.py](#)

6.7 lib.user.User Class Reference

Create to organize users information and apart this from server module.

Inherits [object](#).

Public Member Functions

- `def __init__(self, _key, _client, _addr, _nick, _group)`
Constructs the object.

Public Attributes

- `key`
- `client`
- `addr`
- `nick`
- `group`
- `away`

6.7.1 Detailed Description

Create to organize users information and apart this from server module.

Classe to store information of server users

Definition at line 10 of file user.py.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 `def lib.user.User.__init__(self, _key, _client, _addr, _nick, _group)`

Constructs the object.

Parameters

<code>self</code>	The object
<code>_key</code>	The key
<code>_client</code>	The client
<code>_addr</code>	The address
<code>_nick</code>	The nick
<code>_group</code>	The group

Returns

return a instance of `User` class

Definition at line 25 of file user.py.

6.7.3 Member Data Documentation

6.7.3.1 `lib.user.User.addr`

Definition at line 29 of file user.py.

6.7.3.2 lib.user.User.away

Definition at line 32 of file user.py.

6.7.3.3 lib.user.User.client

Definition at line 28 of file user.py.

6.7.3.4 lib.user.User.group

Definition at line 31 of file user.py.

6.7.3.5 lib.user.User.key

Definition at line 27 of file user.py.

6.7.3.6 lib.user.User.nick

Definition at line 30 of file user.py.

The documentation for this class was generated from the following file:

- lib/[user.py](#)

6.8 lib.window.Window Class Reference

Class for window control using Curses.

Inherits [object](#).

Public Member Functions

- def [__init__](#) (self)
Constructs the object.
- def [start](#) (self, [std](#))
Start all Curses windows: a standard window, a header window, a data window and a input window.
- def [close](#) (self)
{ function_description }
- def [addstr_header](#) (self, [_str](#))
Add text to header window.
- def [addstr_data](#) (self, [_str](#), [_type](#))
Add text to data window, depending of type, the color is different.
- def [clear_header](#) (self)
Clear header window.
- def [clear_data](#) (self)
clear data window.

Public Attributes

- [std](#)
- [text_color](#)
- [header_color](#)
- [input_color](#)
- [server_color](#)
- [private_color](#)
- [error_color](#)
- [header](#)
- [data](#)
- [prompt](#)
- [tex](#)

6.8.1 Detailed Description

Class for window control using Curses.

Definition at line 15 of file window.py.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 `def lib.window.Window.__init__(self)`

Constructs the object.

Parameters

<i>self</i>	The object
-------------	------------

Definition at line 22 of file window.py.

6.8.3 Member Function Documentation

6.8.3.1 `def lib.window.Window.addstr_data (self, _str, _type)`

Add text to data window, depending of type, the color is different.

Parameters

<i>self</i>	The object
<i>_str</i>	The string
<i>_type</i>	The type

Returns

void method.

Definition at line 137 of file window.py.

6.8.3.2 `def lib.window.Window.addstr_header (self, _str)`

Add text to header window.

Parameters

<i>self</i>	The object
<i>_str</i>	The string

Returns

void method.

Definition at line 122 of file window.py.

6.8.3.3 `def lib.window.Window.clear_data (self)`

clear data window.

Parameters

<i>self</i>	The object
-------------	------------

Returns

void method.

Definition at line 168 of file window.py.

6.8.3.4 `def lib.window.Window.clear_header (self)`

Clear header window.

Parameters

<i>self</i>	The object
-------------	------------

Returns

void method.

Definition at line 156 of file window.py.

6.8.3.5 `def lib.window.Window.close (self)`

{ function_description }

Parameters

<i>self</i>	The object
-------------	------------

Returns

{ description_of_the_return_value }

Definition at line 106 of file window.py.

6.8.3.6 `def lib.window.Window.start (self, std)`

Start all Curses windows: a standard window, a header window, a data window and a input window.

Parameters

<i>self</i>	The object
<i>std</i>	The standard window

Returns

Only create windows to client.

Definition at line 35 of file window.py.

6.8.4 Member Data Documentation

6.8.4.1 `lib.window.Window.data`

Definition at line 75 of file window.py.

6.8.4.2 `lib.window.Window.error_color`

Definition at line 64 of file window.py.

6.8.4.3 `lib.window.Window.header`

Definition at line 70 of file window.py.

6.8.4.4 lib.window.Window.header_color

Definition at line 52 of file window.py.

6.8.4.5 lib.window.Window.input_color

Definition at line 55 of file window.py.

6.8.4.6 lib.window.Window.private_color

Definition at line 61 of file window.py.

6.8.4.7 lib.window.Window.prompt

Definition at line 78 of file window.py.

6.8.4.8 lib.window.Window.server_color

Definition at line 58 of file window.py.

6.8.4.9 lib.window.Window.std

Definition at line 37 of file window.py.

6.8.4.10 lib.window.Window.tex

Definition at line 88 of file window.py.

6.8.4.11 lib.window.Window.text_color

Definition at line 49 of file window.py.

The documentation for this class was generated from the following file:

- [lib/window.py](#)

Chapter 7

File Documentation

7.1 app.py File Reference

Namespaces

- `app`

Functions

- `def app.server (_port)`
Starts app as server.
- `def app.client (_ip, _port)`
Starts app as client.

Variables

- string `app.ERROR_MSG` = "Is necessary use arguments to initialize this application\n"

7.2 lib/__init__.py File Reference

Namespaces

- `lib`

7.3 lib/client.py File Reference

Classes

- class `lib.client.Client`
Class for client application mode.

Namespaces

- [lib.client](#)

Functions

- def [lib.client.end_client](#) (_client)
Ends a client.

7.4 lib/files_db.py File Reference

Classes

- class [lib.files_db.Files_DB](#)
Class for files db.

Namespaces

- [lib.files_db](#)

Functions

- def [lib.files_db.dict_factory](#) (cursor, row)
Create a factory to convert result of a query in a dictionary.

7.5 lib/log.py File Reference

Classes

- class [lib.log.Log](#)
Class for log.

Namespaces

- [lib.log](#)

7.6 lib/msg.py File Reference

Classes

- class [lib.msg.Msg](#)
Class for transit of messages.

Namespaces

- [lib.msg](#)

7.7 lib/server.py File Reference

Classes

- class [lib.server.Server](#)
Class for server application mode.

Namespaces

- [lib.server](#)

7.8 lib/user.py File Reference

Classes

- class [lib.user.User](#)
Create to organize users information and apart this from server module.

Namespaces

- [lib.user](#)

7.9 lib/window.py File Reference

Classes

- class [lib.window.Window](#)
Class for window control using Curses.

Namespaces

- [lib.window](#)

Index

- `__init__`
 - `lib::client::Client`, 14
 - `lib::log::Log`, 19
 - `lib::msg::Msg`, 20
 - `lib::server::Server`, 24
 - `lib::user::User`, 34
 - `lib::window::Window`, 36
- `addr`
 - `lib::user::User`, 34
- `addstr_data`
 - `lib::window::Window`, 36
- `addstr_header`
 - `lib::window::Window`, 37
- `app`, 9
 - `client`, 9
 - `ERROR_MSG`, 10
 - `server`, 9
- `app.py`, 41
- `away`
 - `lib::user::User`, 34
- `clear_data`
 - `lib::window::Window`, 37
- `clear_header`
 - `lib::window::Window`, 37
- `client`
 - `app`, 9
 - `lib::user::User`, 35
- `close`
 - `lib::window::Window`, 37
- `com_away`
 - `lib::server::Server`, 25
- `com_ban`
 - `lib::server::Server`, 25
- `com_clear`
 - `lib::server::Server`, 25
- `com_create`
 - `lib::server::Server`, 26
- `com_delete`
 - `lib::server::Server`, 26
- `com_error`
 - `lib::server::Server`, 26
- `com_file`
 - `lib::server::Server`, 27
- `com_get_file`
 - `lib::server::Server`, 27
- `com_help`
 - `lib::server::Server`, 27
- `com_join`
 - `lib::server::Server`, 28
- `com_kick`
 - `lib::server::Server`, 28
- `com_leave`
 - `lib::server::Server`, 28
- `com_list`
 - `lib::server::Server`, 29
- `com_list_files`
 - `lib::server::Server`, 29
- `com_msg`
 - `lib::server::Server`, 30
- `com_nick`
 - `lib::server::Server`, 30
- `coms_list`
 - `lib::server::Server`, 32
- `create`
 - `lib::files_db::Files_DB`, 16
 - `lib::msg::Msg`, 21
- `data`
 - `lib::msg::Msg`, 22
 - `lib::window::Window`, 38
- `delete_files_by_group`
 - `lib::files_db::Files_DB`, 17
- `dict_factory`
 - `lib::files_db`, 11
- `ERROR_MSG`
 - `app`, 10
- `end_client`
 - `lib::client`, 11
- `error_color`
 - `lib::window::Window`, 38
- `file_by_name_group`
 - `lib::files_db::Files_DB`, 17
- `file_names_by_group`
 - `lib::files_db::Files_DB`, 17
- `group`
 - `lib::user::User`, 35
- `groups_dict`
 - `lib::server::Server`, 32
- `header`
 - `lib::window::Window`, 38
- `header_color`
 - `lib::window::Window`, 38
- `input_color`
 - `lib::window::Window`, 39

- insert_file
 - lib::files_db::Files_DB, 18
- ip_addr
 - lib::client::Client, 15
 - lib::server::Server, 33
- key
 - lib::user::User, 35
- last_user_key
 - lib::server::Server, 33
- lib, 10
- lib.client, 10
- lib.client.Client, 13
- lib.files_db, 11
- lib.files_db.Files_DB, 16
- lib.log, 12
- lib.log.Log, 18
- lib.msg, 12
- lib.msg.Msg, 20
- lib.server, 12
- lib.server.Server, 23
- lib.user, 12
- lib.user.User, 33
- lib.window, 12
- lib.window.Window, 35
- lib/__init__.py, 41
- lib/client.py, 41
- lib/files_db.py, 42
- lib/log.py, 42
- lib/msg.py, 42
- lib/server.py, 43
- lib/user.py, 43
- lib/window.py, 43
- lib::client
 - end_client, 11
- lib::client::Client
 - __init__, 14
 - ip_addr, 15
 - listen_server, 14
 - msg, 15
 - port, 15
 - send_server, 15
 - sock, 15
 - thread_listen, 15
 - thread_send, 16
 - window, 16
- lib::files_db
 - dict_factory, 11
- lib::files_db::Files_DB
 - create, 16
 - delete_files_by_group, 17
 - file_by_name_group, 17
 - file_names_by_group, 17
 - insert_file, 18
- lib::log::Log
 - __init__, 19
 - reg, 19
 - time, 19
 - type, 19
- lib::msg::Msg
 - __init__, 20
 - create, 21
 - data, 22
 - log, 22
 - receive, 21
 - receive_all, 21
 - send, 22
- lib::server::Server
 - __init__, 24
 - com_away, 25
 - com_ban, 25
 - com_clear, 25
 - com_create, 26
 - com_delete, 26
 - com_error, 26
 - com_file, 27
 - com_get_file, 27
 - com_help, 27
 - com_join, 28
 - com_kick, 28
 - com_leave, 28
 - com_list, 29
 - com_list_files, 29
 - com_msg, 30
 - com_nick, 30
 - coms_list, 32
 - groups_dict, 32
 - ip_addr, 33
 - last_user_key, 33
 - list_client, 30
 - listen, 31
 - manage_coms, 31
 - msg, 33
 - msg_of_day, 31
 - port, 33
 - send_to_all, 32
 - server_send, 32
 - sock, 33
 - trds_dict, 33
 - users_list, 33
- lib::user::User
 - __init__, 34
 - addr, 34
 - away, 34
 - client, 35
 - group, 35
 - key, 35
 - nick, 35
- lib::window::Window
 - __init__, 36
 - addstr_data, 36
 - addstr_header, 37
 - clear_data, 37
 - clear_header, 37
 - close, 37
 - data, 38

- error_color, [38](#)
- header, [38](#)
- header_color, [38](#)
- input_color, [39](#)
- private_color, [39](#)
- prompt, [39](#)
- server_color, [39](#)
- start, [38](#)
- std, [39](#)
- tex, [39](#)
- text_color, [39](#)
- list_client
 - lib::server::Server, [30](#)
- listen
 - lib::server::Server, [31](#)
- listen_server
 - lib::client::Client, [14](#)
- log
 - lib::msg::Msg, [22](#)
- manage_coms
 - lib::server::Server, [31](#)
- msg
 - lib::client::Client, [15](#)
 - lib::server::Server, [33](#)
- msg_of_day
 - lib::server::Server, [31](#)
- nick
 - lib::user::User, [35](#)
- object, [23](#)
- port
 - lib::client::Client, [15](#)
 - lib::server::Server, [33](#)
- private_color
 - lib::window::Window, [39](#)
- prompt
 - lib::window::Window, [39](#)
- receive
 - lib::msg::Msg, [21](#)
- receive_all
 - lib::msg::Msg, [21](#)
- reg
 - lib::log::Log, [19](#)
- send
 - lib::msg::Msg, [22](#)
- send_server
 - lib::client::Client, [15](#)
- send_to_all
 - lib::server::Server, [32](#)
- server
 - app, [9](#)
- server_color
 - lib::window::Window, [39](#)
- server_send
 - lib::server::Server, [32](#)
- sock
 - lib::client::Client, [15](#)
 - lib::server::Server, [33](#)
- start
 - lib::window::Window, [38](#)
- std
 - lib::window::Window, [39](#)
- tex
 - lib::window::Window, [39](#)
- text_color
 - lib::window::Window, [39](#)
- thread_listen
 - lib::client::Client, [15](#)
- thread_send
 - lib::client::Client, [16](#)
- time
 - lib::log::Log, [19](#)
- trds_dict
 - lib::server::Server, [33](#)
- type
 - lib::log::Log, [19](#)
- users_list
 - lib::server::Server, [33](#)
- window
 - lib::client::Client, [16](#)