

BSRN Projekt: SLCP

**Fabian Matzollek, Michael Nguyen, Muhammed Can Sahin, Jin Huh, Burim
Rashica**

Generated by Doxygen 1.9.1

| | |
|--|-----------|
| 1 Namespace Index | 1 |
| 1.1 Namespace List | 1 |
| 2 Class Index | 3 |
| 2.1 Class List | 3 |
| 3 File Index | 5 |
| 3.1 File List | 5 |
| 4 Namespace Documentation | 7 |
| 4.1 broadcast_server Namespace Reference | 7 |
| 4.1.1 Detailed Description | 7 |
| 4.1.2 Function Documentation | 7 |
| 4.1.2.1 find_user_idx() | 8 |
| 4.1.2.2 run_broadcast_server() | 8 |
| 4.1.3 Variable Documentation | 8 |
| 4.1.3.1 BROADCAST_IP | 8 |
| 4.1.3.2 config | 8 |
| 4.1.3.3 USERS | 8 |
| 4.1.3.4 WHOIS_PORT | 8 |
| 4.2 cli Namespace Reference | 9 |
| 4.2.1 Detailed Description | 9 |
| 4.2.2 Function Documentation | 9 |
| 4.2.2.1 find_client_config() | 9 |
| 4.2.2.2 list_users() | 9 |
| 4.2.2.3 listen_to_client() | 10 |
| 4.2.2.4 load_config() | 10 |
| 4.2.2.5 main() | 10 |
| 4.2.2.6 print_help() | 10 |
| 4.3 client Namespace Reference | 10 |
| 4.3.1 Detailed Description | 11 |
| 4.3.2 Function Documentation | 11 |
| 4.3.2.1 load_config() | 11 |
| 4.3.2.2 main() | 11 |
| 5 Class Documentation | 13 |
| 5.1 broadcast_server.BroadcastServer Class Reference | 13 |
| 5.1.1 Detailed Description | 13 |
| 5.1.2 Constructor & Destructor Documentation | 13 |
| 5.1.2.1 __init__() | 13 |
| 5.1.3 Member Function Documentation | 14 |
| 5.1.3.1 start() | 14 |
| 5.1.3.2 stop() | 14 |
| 5.1.4 Member Data Documentation | 14 |

| | |
|--|----|
| 5.1.4.1 broadcast_ip | 14 |
| 5.1.4.2 online_users | 14 |
| 5.1.4.3 port | 14 |
| 5.1.4.4 running | 15 |
| 5.1.4.5 sock | 15 |
| 5.2 client.Client Class Reference | 15 |
| 5.2.1 Detailed Description | 16 |
| 5.2.2 Constructor & Destructor Documentation | 16 |
| 5.2.2.1 __init__() | 16 |
| 5.2.3 Member Function Documentation | 16 |
| 5.2.3.1 _get_image_port_for() | 16 |
| 5.2.3.2 _handle_incoming_image() | 17 |
| 5.2.3.3 _ipc_accept_loop() | 17 |
| 5.2.3.4 _ipc_handler() | 17 |
| 5.2.3.5 _parse_knowusers() | 17 |
| 5.2.3.6 _udp_listener_thread() | 18 |
| 5.2.3.7 broadcast() | 18 |
| 5.2.3.8 get_online_users() | 18 |
| 5.2.3.9 handle_cli_command() | 18 |
| 5.2.3.10 handle_udp_message() | 19 |
| 5.2.3.11 join_network() | 19 |
| 5.2.3.12 periodic_who_sync() | 19 |
| 5.2.3.13 send_image() | 19 |
| 5.2.3.14 send_message() | 20 |
| 5.2.3.15 send_to_cli() | 20 |
| 5.2.3.16 start_ipc_server() | 20 |
| 5.2.3.17 start_tcp_server() | 20 |
| 5.2.3.18 start_udp_listener() | 20 |
| 5.2.4 Member Data Documentation | 21 |
| 5.2.4.1 autoreply | 21 |
| 5.2.4.2 away_message | 21 |
| 5.2.4.3 bilder_ordner | 21 |
| 5.2.4.4 broadcast_ip | 21 |
| 5.2.4.5 cli_sockets | 21 |
| 5.2.4.6 config | 21 |
| 5.2.4.7 handle | 21 |
| 5.2.4.8 handle_has_been_set | 22 |
| 5.2.4.9 has_joined | 22 |
| 5.2.4.10 host_ip | 22 |
| 5.2.4.11 idx | 22 |
| 5.2.4.12 image_ipc_port | 22 |
| 5.2.4.13 ipc_port | 22 |

| | |
|--|-----------|
| 5.2.4.14 known_users | 22 |
| 5.2.4.15 name | 22 |
| 5.2.4.16 sock | 23 |
| 5.2.4.17 udp_port | 23 |
| 5.2.4.18 whois_port | 23 |
| 6 File Documentation | 25 |
| 6.1 broadcast_server.py File Reference | 25 |
| 6.2 cli.py File Reference | 25 |
| 6.3 client.py File Reference | 26 |
| Index | 27 |

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

| | |
|----------------------------------|----|
| broadcast_server | 7 |
| cli | 9 |
| client | 10 |

Chapter 2

Class Index

2.1 Class List

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

| | |
|--|----|
| broadcast_server.BroadcastServer | 13 |
| client.Client | 15 |

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

| | |
|-------------------------------------|----|
| broadcast_server.py | 25 |
| cli.py | 25 |
| client.py | 26 |

Chapter 4

Namespace Documentation

4.1 broadcast_server Namespace Reference

Classes

- class [BroadcastServer](#)

Functions

- def [find_user_idx](#) (ip, port)
- def [run_broadcast_server](#) ()

Variables

- [config](#) = toml.load("config.toml")
- [BROADCAST_IP](#) = [config](#)["network"]["broadcast_ip"]
- [WHOIS_PORT](#) = [config](#)["network"]["whoisport"]
- [USERS](#) = [config](#)["users"]

4.1.1 Detailed Description

@file broadcast_server.py

@brief Implementierung des Discovery-Servers für das Simple Local Chat Protocol (SLCP).

Dieses Modul stellt den Discovery-Dienst bereit, der Broadcast-Nachrichten (z. B. JOIN, WHO, LEAVE) empfängt u

4.1.2 Function Documentation

4.1.2.1 find_user_idx()

```
def broadcast_server.find_user_idx (
    ip,
    port )
```

@brief Findet den Benutzerindex anhand von IP und Port.
@param ip String, die IP-Adresse des Benutzers.
@param port Integer, der Port des Benutzers.
@return Integer, der Index des Benutzers in der Konfiguration, oder None, wenn nicht gefunden.

4.1.2.2 run_broadcast_server()

```
def broadcast_server.run_broadcast_server ( )
```

@brief Hauptfunktion zum Starten des Broadcast-Servers.

4.1.3 Variable Documentation

4.1.3.1 BROADCAST_IP

```
string broadcast_server.BROADCAST_IP = config["network"]["broadcast_ip"]
```

4.1.3.2 config

```
broadcast_server.config = toml.load("config.toml")
```

4.1.3.3 USERS

```
list broadcast_server.USERS = config["users"]
```

4.1.3.4 WHOIS_PORT

```
int broadcast_server.WHOIS_PORT = config["network"]["whoisport"]
```

4.2 cli Namespace Reference

Functions

- def `load_config` ()
- def `find_client_config` (config, idx)
- def `list_users` (config)
- def `listen_to_client` (sock)
- def `print_help` ()
- def `main` ()

4.2.1 Detailed Description

@file cli.py

@brief Kommandozeilenschnittstelle (CLI) für das Simple Local Chat Protocol (SLCP).

Dieses Modul stellt die Benutzerschnittstelle für den Chat-Client bereit, über die Benutzer Befehle wie JOIN,

4.2.2 Function Documentation

4.2.2.1 `find_client_config()`

```
def cli.find_client_config (  
    config,  
    idx )
```

@brief Findet die Konfiguration für einen Benutzer anhand seines Indexes.

@param config Dictionary mit den Konfigurationsdaten.

@param idx Integer, der den Index des Benutzers angibt (1-basiert).

@return Dictionary mit den Benutzerkonfigurationsdaten.

@throws Exception wenn kein Benutzer mit dem angegebenen Index gefunden wird.

4.2.2.2 `list_users()`

```
def cli.list_users (  
    config )
```

@brief Listet alle verfügbaren Benutzer aus der Konfiguration auf.

@param config Dictionary mit den Konfigurationsdaten.

4.2.2.3 listen_to_client()

```
def cli.listen_to_client (
    sock )
```

@brief Lauscht auf eingehende Nachrichten vom Client-Prozess und gibt diese aus.
@param sock Socket-Objekt für die Kommunikation mit dem Client.

4.2.2.4 load_config()

```
def cli.load_config ( )
```

@brief Lädt die Konfigurationsdatei config.toml.
@return Dictionary mit den geladenen Konfigurationsdaten.

4.2.2.5 main()

```
def cli.main ( )
```

@brief Hauptfunktion der CLI, initialisiert die Verbindung und verarbeitet Benutzereingaben.

4.2.2.6 print_help()

```
def cli.print_help ( )
```

@brief Gibt eine Hilfeübersicht mit verfügbaren Befehlen aus.

4.3 client Namespace Reference

Classes

- class [Client](#)

Functions

- def [load_config](#) ()
- def [main](#) ()

4.3.1 Detailed Description

```
@file client.py
```

```
@brief Client-Implementierung für das Simple Local Chat Protocol (SLCP).
```

Dieses Modul implementiert die Netzwerkkommunikation und die Logik für einen Chat-Client, einschließlich UDP-

4.3.2 Function Documentation

4.3.2.1 load_config()

```
def client.load_config ( )
```

```
@brief Lädt die Konfigurationsdatei config.toml.
```

```
@return Dictionary mit den geladenen Konfigurationsdaten.
```

4.3.2.2 main()

```
def client.main ( )
```

```
@brief Hauptfunktion des Clients, initialisiert den Client-Prozess.
```


Chapter 5

Class Documentation

5.1 broadcast_server.BroadcastServer Class Reference

Public Member Functions

- def `__init__` (self, port=`WHOIS_PORT`, broadcast_ip=`BROADCAST_IP`)
- def `start` (self)
- def `stop` (self)

Public Attributes

- `port`
- `broadcast_ip`
- `online_users`
- `running`
- `sock`

5.1.1 Detailed Description

```
@class BroadcastServer
@brief Verwaltet den Discovery-Dienst für SLCP, empfängt und verarbeitet Broadcast-Nachrichten.
```

5.1.2 Constructor & Destructor Documentation

5.1.2.1 `__init__()`

```
def broadcast_server.BroadcastServer.__init__ (
    self,
    port = WHOIS_PORT,
    broadcast_ip = BROADCAST_IP )

@brief Initialisiert den Broadcast-Server.
@param port Integer, der Port für Broadcast-Nachrichten (Standard: WHOIS_PORT).
@param broadcast_ip String, die Broadcast-IP-Adresse (Standard: BROADCAST_IP).
```

5.1.3 Member Function Documentation

5.1.3.1 start()

```
def broadcast_server.BroadcastServer.start (
    self )
```

@brief Startet den Broadcast-Server und lauscht auf eingehende Nachrichten.

5.1.3.2 stop()

```
def broadcast_server.BroadcastServer.stop (
    self )
```

@brief Stoppt den Broadcast-Server und schließt den Socket.

5.1.4 Member Data Documentation

5.1.4.1 broadcast_ip

```
broadcast_server.BroadcastServer.broadcast_ip
```

5.1.4.2 online_users

```
broadcast_server.BroadcastServer.online_users
```

5.1.4.3 port

```
broadcast_server.BroadcastServer.port
```

5.1.4.4 running

`broadcast_server.BroadcastServer.running`

5.1.4.5 sock

`broadcast_server.BroadcastServer.sock`

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

- [broadcast_server.py](#)

5.2 client.Client Class Reference

Public Member Functions

- `def __init__ (self, idx, config)`
- `def periodic_who_sync (self)`
- `def start_udp_listener (self)`
- `def handle_udp_message (self, msg, addr)`
- `def get_online_users (self)`
- `def send_to_cli (self, text)`
- `def start_ipc_server (self)`
- `def handle_cli_command (self, msg)`
- `def join_network (self)`
- `def send_message (self, target, text)`
- `def broadcast (self, message)`
- `def send_image (self, target, path)`
- `def start_tcp_server (self)`

Public Attributes

- `has_joined`
- `config`
- `idx`
- `name`
- `handle`
- `host_ip`
- `udp_port`
- `ipc_port`
- `image_ipc_port`
- `broadcast_ip`
- `whois_port`
- `known_users`
- `away_message`
- `autoreply`
- `bilder_ordner`
- `cli_sockets`
- `handle_has_been_set`
- `sock`

Private Member Functions

- `def _udp_listener_thread` (self)
- `def _parse_knowusers` (self, users_str)
- `def _ipc_accept_loop` (self, server)
- `def _ipc_handler` (self, conn)
- `def _get_image_port_for` (self, handle)
- `def _handle_incoming_image` (self, conn)

5.2.1 Detailed Description

```
@class Client
@brief Repräsentiert einen Chat-Client im SLCP, der Nachrichten und Bilder sendet/empfängt.
```

5.2.2 Constructor & Destructor Documentation

5.2.2.1 __init__()

```
def client.Client.__init__ (
    self,
    idx,
    config )

@brief Initialisiert den Client mit Konfigurationsdaten.
@param idx Integer, der den Index des Benutzers in der Konfiguration angibt.
@param config Dictionary mit den Konfigurationsdaten aus config.toml.
```

5.2.3 Member Function Documentation

5.2.3.1 _get_image_port_for()

```
def client.Client._get_image_port_for (
    self,
    handle ) [private]

@brief Ermittelt den Bild-IPC-Port für einen bestimmten Benutzer.
@param handle String, der Handle des Benutzers.
@return Integer, der Bild-IPC-Port, oder None, wenn nicht gefunden.
```

5.2.3.2 `_handle_incoming_image()`

```
def client.Client._handle_incoming_image (
    self,
    conn ) [private]

@brief Verarbeitet eingehende Bilddaten und speichert diese lokal.
@param conn Socket-Objekt der TCP-Verbindung.
```

5.2.3.3 `_ipc_accept_loop()`

```
def client.Client._ipc_accept_loop (
    self,
    server ) [private]

@brief Akzeptiert eingehende IPC-Verbindungen von der CLI.
@param server Socket-Objekt des IPC-Servers.
```

5.2.3.4 `_ipc_handler()`

```
def client.Client._ipc_handler (
    self,
    conn ) [private]

@brief Verarbeitet eingehende Befehle von der CLI.
@param conn Socket-Objekt der IPC-Verbindung.
```

5.2.3.5 `_parse_knowusers()`

```
def client.Client._parse_knowusers (
    self,
    users_str ) [private]

@brief Parst die KNOWUSERS-Nachricht und aktualisiert die Liste bekannter Benutzer.
@param users_str String mit der Liste der Benutzer (Handle, IP, Port).
```

5.2.3.6 `_udp_listener_thread()`

```
def client.Client._udp_listener_thread (
    self ) [private]
```

@brief Lauscht auf eingehende UDP-Nachrichten und verarbeitet sie.

5.2.3.7 `broadcast()`

```
def client.Client.broadcast (
    self,
    message )
```

@brief Sendet eine Broadcast-Nachricht an den Discovery-Server.
@param message String, die zu sendende Nachricht.

5.2.3.8 `get_online_users()`

```
def client.Client.get_online_users (
    self )
```

@brief Sendet eine WHO-Anfrage, um die Liste der online-Benutzer zu aktualisieren.

5.2.3.9 `handle_cli_command()`

```
def client.Client.handle_cli_command (
    self,
    msg )
```

@brief Verarbeitet Befehle von der CLI gemäß SLCP.
@param msg String, der empfangene Befehl.

5.2.3.10 handle_udp_message()

```
def client.Client.handle_udp_message (
    self,
    msg,
    addr )

@brief Verarbeitet eingehende UDP-Nachrichten gemäß SLCP.
@param msg String, die empfangene Nachricht.
@param addr Tuple mit IP-Adresse und Port des Absenders.
```

5.2.3.11 join_network()

```
def client.Client.join_network (
    self )

@brief Sendet eine JOIN-Nachricht, um dem Chat beizutreten.
```

5.2.3.12 periodic_who_sync()

```
def client.Client.periodic_who_sync (
    self )

@brief Sendet periodisch WHO-Anfragen, um die Liste bekannter Benutzer zu aktualisieren.
```

5.2.3.13 send_image()

```
def client.Client.send_image (
    self,
    target,
    path )

@brief Sendet ein Bild an einen bestimmten Benutzer.
@param target String, der Handle des Empfängers.
@param path String, der Pfad zur Bilddatei.
```

5.2.3.14 send_message()

```
def client.Client.send_message (
    self,
    target,
    text )
```

@brief Sendet eine Textnachricht an einen bestimmten Benutzer.
@param target String, der Handle des Empfängers.
@param text String, der Text der Nachricht.

5.2.3.15 send_to_cli()

```
def client.Client.send_to_cli (
    self,
    text )
```

@brief Sendet eine Nachricht an alle verbundenen CLI-Instanzen.
@param text String, die an die CLI gesendet werden soll.

5.2.3.16 start_ipc_server()

```
def client.Client.start_ipc_server (
    self )
```

@brief Startet den IPC-Server für die Kommunikation mit der CLI.

5.2.3.17 start_tcp_server()

```
def client.Client.start_tcp_server (
    self )
```

@brief Startet den TCP-Server für den Empfang von Bildern.

5.2.3.18 start_udp_listener()

```
def client.Client.start_udp_listener (
    self )
```

@brief Startet den UDP-Listener-Thread für eingehende Nachrichten.

5.2.4 Member Data Documentation

5.2.4.1 autoreply

`client.Client.autoreply`

5.2.4.2 away_message

`client.Client.away_message`

5.2.4.3 bilder_ordner

`client.Client.bilder_ordner`

5.2.4.4 broadcast_ip

`client.Client.broadcast_ip`

5.2.4.5 cli_sockets

`client.Client.cli_sockets`

5.2.4.6 config

`client.Client.config`

5.2.4.7 handle

`client.Client.handle`

5.2.4.8 handle_has_been_set

```
client.Client.handle_has_been_set
```

5.2.4.9 has_joined

```
client.Client.has_joined
```

5.2.4.10 host_ip

```
client.Client.host_ip
```

5.2.4.11 idx

```
client.Client.idx
```

5.2.4.12 image_ipc_port

```
client.Client.image_ipc_port
```

5.2.4.13 ipc_port

```
client.Client.ipc_port
```

5.2.4.14 known_users

```
client.Client.known_users
```

5.2.4.15 name

```
client.Client.name
```

5.2.4.16 sock

```
client.Client.sock
```

5.2.4.17 udp_port

```
client.Client.udp_port
```

5.2.4.18 whois_port

```
client.Client.whois_port
```

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

- [client.py](#)

Chapter 6

File Documentation

6.1 broadcast_server.py File Reference

Classes

- class [broadcast_server.BroadcastServer](#)

Namespaces

- [broadcast_server](#)

Functions

- def [broadcast_server.find_user_idx](#) (ip, port)
- def [broadcast_server.run_broadcast_server](#) ()

Variables

- [broadcast_server.config](#) = `toml.load("config.toml")`
- [broadcast_server.BROADCAST_IP](#) = `config["network"]["broadcast_ip"]`
- [broadcast_server.WHOIS_PORT](#) = `config["network"]["whoisport"]`
- [broadcast_server.USERS](#) = `config["users"]`

6.2 cli.py File Reference

Namespaces

- [cli](#)

Functions

- def [cli.load_config](#) ()
- def [cli.find_client_config](#) (config, idx)
- def [cli.list_users](#) (config)
- def [cli.listen_to_client](#) (sock)
- def [cli.print_help](#) ()
- def [cli.main](#) ()

6.3 client.py File Reference

Classes

- class [client.Client](#)

Namespaces

- [client](#)

Functions

- def [client.load_config](#) ()
- def [client.main](#) ()

Index

- `__init__`
 - `broadcast_server.BroadcastServer`, 13
 - `client.Client`, 16
 - `_get_image_port_for`
 - `client.Client`, 16
 - `_handle_incoming_image`
 - `client.Client`, 16
 - `_ipc_accept_loop`
 - `client.Client`, 17
 - `_ipc_handler`
 - `client.Client`, 17
 - `_parse_knownusers`
 - `client.Client`, 17
 - `_udp_listener_thread`
 - `client.Client`, 17
- `autoreply`
 - `client.Client`, 21
- `away_message`
 - `client.Client`, 21
- `bilder_ordner`
 - `client.Client`, 21
- `broadcast`
 - `client.Client`, 18
- `BROADCAST_IP`
 - `broadcast_server`, 8
- `broadcast_ip`
 - `broadcast_server.BroadcastServer`, 14
 - `client.Client`, 21
- `broadcast_server`, 7
 - `BROADCAST_IP`, 8
 - `config`, 8
 - `find_user_idx`, 7
 - `run_broadcast_server`, 8
 - `USERS`, 8
 - `WHOIS_PORT`, 8
- `broadcast_server.BroadcastServer`, 13
 - `__init__`, 13
 - `broadcast_ip`, 14
 - `online_users`, 14
 - `port`, 14
 - `running`, 14
 - `sock`, 15
 - `start`, 14
 - `stop`, 14
- `broadcast_server.py`, 25
- `cli`, 9
 - `find_client_config`, 9
 - `list_users`, 9
 - `listen_to_client`, 9
 - `load_config`, 10
 - `main`, 10
 - `print_help`, 10
- `cli.py`, 25
- `cli_sockets`
 - `client.Client`, 21
- `client`, 10
 - `load_config`, 11
 - `main`, 11
- `client.Client`, 15
 - `__init__`, 16
 - `_get_image_port_for`, 16
 - `_handle_incoming_image`, 16
 - `_ipc_accept_loop`, 17
 - `_ipc_handler`, 17
 - `_parse_knownusers`, 17
 - `_udp_listener_thread`, 17
 - `autoreply`, 21
 - `away_message`, 21
 - `bilder_ordner`, 21
 - `broadcast`, 18
 - `broadcast_ip`, 21
 - `cli_sockets`, 21
 - `config`, 21
 - `get_online_users`, 18
 - `handle`, 21
 - `handle_cli_command`, 18
 - `handle_has_been_set`, 21
 - `handle_udp_message`, 18
 - `has_joined`, 22
 - `host_ip`, 22
 - `idx`, 22
 - `image_ipc_port`, 22
 - `ipc_port`, 22
 - `join_network`, 19
 - `known_users`, 22
 - `name`, 22
 - `periodic_who_sync`, 19
 - `send_image`, 19
 - `send_message`, 19
 - `send_to_cli`, 20
 - `sock`, 22
 - `start_ipc_server`, 20
 - `start_tcp_server`, 20
 - `start_udp_listener`, 20
 - `udp_port`, 23
 - `whois_port`, 23

client.py, 26
config
 broadcast_server, 8
 client.Client, 21
find_client_config
 cli, 9
find_user_idx
 broadcast_server, 7
get_online_users
 client.Client, 18
handle
 client.Client, 21
handle_cli_command
 client.Client, 18
handle_has_been_set
 client.Client, 21
handle_udp_message
 client.Client, 18
has_joined
 client.Client, 22
host_ip
 client.Client, 22
idx
 client.Client, 22
image_ipc_port
 client.Client, 22
ipc_port
 client.Client, 22
join_network
 client.Client, 19
known_users
 client.Client, 22
list_users
 cli, 9
listen_to_client
 cli, 9
load_config
 cli, 10
 client, 11
main
 cli, 10
 client, 11
name
 client.Client, 22
online_users
 broadcast_server.BroadcastServer, 14
periodic_who_sync
 client.Client, 19
port
 broadcast_server.BroadcastServer, 14
print_help
 cli, 10
run_broadcast_server
 broadcast_server, 8
running
 broadcast_server.BroadcastServer, 14
send_image
 client.Client, 19
send_message
 client.Client, 19
send_to_cli
 client.Client, 20
sock
 broadcast_server.BroadcastServer, 15
 client.Client, 22
start
 broadcast_server.BroadcastServer, 14
start_ipc_server
 client.Client, 20
start_tcp_server
 client.Client, 20
start_udp_listener
 client.Client, 20
stop
 broadcast_server.BroadcastServer, 14
udp_port
 client.Client, 23
USERS
 broadcast_server, 8
WHOIS_PORT
 broadcast_server, 8
whois_port
 client.Client, 23