FastChat Documentation

A.K.A

Nov 25, 2022

CONTENTS:

CHAPTER

ONE

CLIENT

1.1 client module

client.**Send_group_key**(*recipient*, *g_id*)
Sending the group key to someone. Used when we need to add someone or kick a person.

client.**add**()
Used for adding a member to the group.

client.**add_friend**()
When we want to make someone our friend. We can send messages to friends only.

client.**connect_balancer**(*balancerIP*, *balancerPort*)
This function is used to connect the client to the load balancer

Parameters

- **balancerIP** (*str*) – IP of balancer
- **balancerPort** (*int*) – Port of balancer

Returns socket of balancer

Return type socket

client.**connect_server**(*server_json*)
For connecting to a single server

Parameters **server_json** (*list*) – contains information about the server

client.**connect_servers**()
Used to connect client with all the servers

client.**create_group**()
Creates a group with us as the only member. We can keep adding more people.

client.**execute_command**(*Command_type*)
The main execute function which is used when the user wants us to do something.

Parameters **Command_type** (*str*) – The command the user want to perform.

Returns True for valid command else False

Return type bool

client.**get_name_locally**(*ID*)
Gets the name of a person who is already our friend

Parameters **ID** (*int*) – The ID of the person whose name we want

Returns Returns the name of the person if he is our friend else returns empty string

Return type str

client.**get_server**(*balancer*)
Asks the balancer for the server it should connect to

Parameters **balancer** (*socket*) – balancer socket

Returns Returns the server ID

Return type int

client.**group_image**()
For sending an image in a group. It is encrypted using a symmetric key which is only with the members of the group.

client.**group_message**()
For messaging in a group. It is encrypted using a symmetric key which is only with the members of the group.

client.**handle_New_group**(*message*)
This is used for getting the group_ID of the group the client created from the server

Parameters **message** (*dict*) – The message the server sent in dict format with all fields

client.**handle_accept_friend**(*message*)
Handles when someone sends a friend request. It generates a symmetric key and sends the friend back using RSA

Parameters **message** (*dict*) – The message the person sent in dict format with all fields

client.**handle_add**(*message*)
The case when someone is added to the group.

Parameters **message** (*dict*) – The message the person/server sent in dict format with all fields.

client.**handle_friend_key**(*message*)
Handles the case when a friend send us a key for communication

Parameters **message** (*dict*) – The message the person/server sent in dict format with all fields

client.**handle_group_image**(*message*)
When we recieve an image in a group. It is stored in our system as received_(image title) format

Parameters **message** (*dict*) – The message the person/server sent in dict format with all fields

client.**handle_group_key**(*message*)
When we are new to a group the admin send us the key through RSA. This function stores that key in our local database.

Parameters **message** (*dict*) – The message the person/server sent in dict format with all fields

client.**handle_group_message**(*message*)
When we recieve a message in a group

Parameters **message** (*dict*) – The message the person/server sent in dict format with all fields

client.**handle_kick**(*message*)
When someone is kicked from a group. It can be us or someone else. It is also ensured that new key will be generated for the group.

Parameters **message** (*dict*) – The message the person/server sent in dict format with all fields.

client.**handle_server**(*server*)
The main handle function when we recieve something from the server. Different actions are taken depending on the type of message recieved.

Parameters **server** (*socket*) – the server socket

client.**handle_single_image**(*message*)
Handles the case when a friend sends us an image. It is stored in our system as received_(image title)

Parameters **message** (*dict*) – The message the person/server sent in dict format with all fields

client.**handle_single_message**(*message*)
Handles the case when a friend sends us a message

Parameters **message** (*dict*) – The message the person/server sent in dict format with all fields

client.**help**()
Prints the basic commands for the help of the user

client.**kick**()
For kicking someone from a group. It is also ensured that we generate a new key and share to all members except the removed one through RSA for each member. Also only admin can kick from a group.

client.**q**()
Used for exiting the user

client.**register**(*server_json*, *pub_str*)
This is called when the client joins for the first time. For storing his credentials

Parameters

- **server_json** (*list*) – Contains the information of the server
- **pub_str** (*str*) – The public key of the client in str format

Returns ID assigned to the client by the server

Return type int

client.**request_pending_msgs**()
Used for requesting pending messages from the server

client.**single_image**()
When we want to send an image to a friend

client.**single_message**()
When we want to send a text message to a friend.

client.**write**(*balancer_sock*)
Gets a server from the balancer and takes a command from the user and executes the command through the server the balancer gave.

Parameters **balancer_sock** (*socket*) – The socket of the balancer

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

C

INDEX