Prepared for:
Module : Network Programming
Project : NWP_myteams_2019
School : Epitech Paris
Prom : 2023

By:
-Antoine Poisson
-Quentin Maillard
-Michel Kusy

May 2020 Internet Protocol

Table of content:

Introduction

Motivation :

The objectives of myteams are to promote of sharing messages, create new users in a database, create new team with users, create new channels in our teams, or send private messages to other users. Basically the goal of this project is to recreate the performance of the real Team.

Scope :

The myteams is intended to provide a reliable process-to-process communication service in a multi network environment. The myteams is intended to be a host-to-host protocol in common use in multiple networks.

Overview

Relation to other protocol :

Our myteams is base on the FTP protocol, which the other project of our module Network Programming. We did something simple with code going from 200 to 599, and also lines ending with “\n”. The message the server sends us looks like this : “000|message|\n” so it’s easier for the client to check and parse.

Function description :

As explained before we have code going from 200 to 599, and our reply on client side is based on what code server send us. From 200 to 299, is a success code.

From 300 to 399, is a success data for client.
From 400 to 499, is a failed data for client.
From 500 to 599, is a server error.

Then we have a list of commands we can do:

- ”/help”, it shows us every commands we can execute. and respond with 200.
- “/login [“username”]”, it log in a client on a new account or an old one if he already logged in with this username. And respond with 300 in case of success and 400 in case of fail. It call also functions from Epitech’s lib :
  server_event_user_created
  server_event_user_logged_in
  client_event_loggedin
  client_error_already_exist
- “/logout”, disconnect the client from the server. And respond with 301. It call also functions from Epitech’s lib :
  server_event_user_logged_out
  client_event_loggedout
- “/users”, get the list of all users that exist on the domain. And respond with 309. It call also functions from Epitech’s lib :
  client_print_users
- “/user [“user_uuid”]”, it get information about a user. And respond with 302 in case of success and 401 in case of fail. It call also functions from Epitech’s lib :
  client_print_user
  client error_unknown_user
- “/send [“user_uuid”] [“message_body”]”, it sends a message to a user. And respond with 303 in case of success and 401 in case of fail. It call also functions from Epitech’s lib :
  server_event_private_message_sended
  client_event_private_message_received
  client_error_unknown_user

- “/messages [“user_uuid”]”, it list all messages exchanged with a user. And respond with 304 in case of success and 401 in case of fail. It call also functions from Epitech’s lib :
  client_private_message_print_messages
  client_error_unknown_user
- “/subscribe [“team_uuid”]”, subscribe to the event of a team and its sub directories (enable reception of all events from a team). And respond with 305 in case of success and 402 in case of fail. It call also functions from Epitech’s lib :
  server_event_user_join_a_team client_error_unknown_team
  client_print_subscribed
- “/subscribed ?[“team_uuid”]”, list all subscribed teams or list all users subscribed to a team. And respond with 402. It call also functions from Epitech’s lib :
  client_error_unknown_team
  client_print_team
  client_print_user
- “/unsubscribe [“team_uuid”]”, unsubscribe from a team. And respond with 306 in case of success and 402 in case of fail. It call also functions from Epitech’s lib :
  server_event_user_leave_a_team client_error_unknown_team
  client_print_unsubscribed
- “/use ?[“team_uuid”] ?[“channel_uuid”] ?[“thread_uuid”]”, use specify a context : team/channel/thread. And respond with 402 or 403 or 404. It call also functions from Epitech’s lib :
  client_error_unknown_team
  client_error_unknown_channel
  client_error_unknown_thread
- “/use”, in term of use for teams.
- “/create [“team_name”] [“team_description”]”, create a new team. And respond with 307 or 308 in case of success and 400 in case of fail. It call also functions from Epitech’s lib :
  server_event_team_created client_event_team_created
  client_error_already_exist
  client_print_team_created

- “/list ”, list all existing teams. And respond with 310. It call also functions from Epitech’s lib :
  client_print_teams
- “/info”, display currently logged user infos. And respond with 302. It call also functions from Epitech’s lib :
  client_print_user

- “/use “team_uuid””, in term of use for channels.
- “/create [“team_name”] [“team_description”]”, create new channel. And respond with 311 or 312 in case of success and 400 in case of fail. It call also functions from Epitech’s lib :
  server_event_channel_created
  client_event_channel_created
  client_error_already_exist
  client_print_channel_created
- “/list”, list all existing channels. And respond with 313. It call also functions from Epitech’s lib :
  client_team_print_channels
- “/info”, display currently selected team infos. And respond with 314. It call also functions from Epitech’s lib :
  client_print_team

- “/use “team_id” “channel_uuid””, in term of use for threads.
- “/create [“team_name”] [“team_description”]”, create new thread. And respond with 315 or 316 in case of success and 400 in case of fail. It call also functions from Epitech’s lib :
  server_event_thread_created
  client_event_thread_created
  client_error_already_exist
  client_print_thread_created
- “/list”, list all existing threads. And respond with 317. It call also functions from Epitech’s lib :
  client_channel_print_threads
- “/info”, display currently selected channel infos. And respond with 318. It call also functions from Epitech’s lib :

client_print_channel

- “/use “team_uuid” “channel_uuid” “thread_uuid””, in term of use for replies.
- “/create [“team_name”] [“team_description”]”, create new reply. And respond with 319 or 320 in case of success and 400 in case of fail. It call also functions from Epitech’s lib :
  server_event_thread_new_message
  client_event_thread_message_received
  client_error_already_exist
  client_print_reply_created
- “/list”, list all existing replies. And respond with 321. It call also functions from Epitech’s lib :
  client_thread_print_replies
- “/info”, display currently selected thread infos. And respond with 322. It call also functions from Epitech’s lib :
  client_thread_print_replies

And other code for differents errors:

- 500 for command not found
- 503 for invalid parameter syntax
- 504 for invalid parameter