Project 4 part II

Readme

1. Team Members

Kanika Gupta, UFID - 96977046, kanikagupta@ufl.edu

Nikhil Chopra, UFID - 98973831, nikhilchopra60@ufl.edu

2. How to run

a. Start the server

Go to - Project4part2/Server/twitterserver

Run Command – mix phx.server

b. Start the client

Go to - Project4part2/Client/twitterclient

Command - escript twitterclient

3. Working

a. Server:

We have created a web socket listening at ws://localhost:4000/socket/websocket. Clients will be joining different topics, where each topic represents followed user's name. So, whenever a tweet event (tweet request) is sent from a user, that tweet is broadcasted to all the users present in the topic bound to that user. So, all the live users will be getting the tweet directly whereas for offline users the tweet would be saved at the server.

The server architecture and internal functioning is same as the architecture described in Project4Part1. The major difference between servers in Project4Part1 and Project4Part2 is that, in this project, the requests to the server are coming on the websockets via Json.

b. Client:

We have simulated each user as a GenSocketClient, which is used as a client to test websockets on Phoenix server. This GenSocketClient will be sending various types of requests to the Phoenix server, such as – register, login, tweet, follow, retweet, logout, query hash tag, query tweets in which the user is mentioned etc. The server is handling all these requests based on the architecture described in Project4Part1.

4. Largest Number of users simulated

As shown in the video, the largest number of users simulated with Phoenix is 8100, while without Phoenix the largest number of users simulated were 10,000 as reported in the Project4Part1 report.

5. Youtube Link for the demonstration video –

https://youtu.be/9-V_aTs0o_Q