

DOS Project 4 Part 2 - Twitter Clone

Rupayan Das - 63992113
Pirouz Naghavi - 30158432

Getting Started

Requirements

- .NET 5 SDK

Running the project

Run the server

- *dotnet run*

Run on client

- *index.html*

Packages Required

- DotNet 5.0
- Akka 1.4.21
- Akka.FSharp 1.4.21
- Newtonsoft.Json 13.0.1
- Suave" Version 2.6.0

What is working

Basic Requirements - ALL FINISHED

- Account Registration along with Login and Logout - Finished
- Sending Tweets along with mentions and hashtags - Finished
- Subscribing to user's tweets - Finished
- Re-tweets - Finished
- Querying for subscribed tweets, hashtags and user mentions - Finished
- Above requirement only for logged in users - Finished

Implementation

Websockets:

We use Suave's Websocket functionality to implement a handshake exchange between each client and the server. We store each client's WebSocket reference in a dictionary with the client's username as the key whenever he/she logs in.

By using Websockets, we are able to display all tweets posted by the user, the tweets where the user is mentioned and the tweets by his subscriptions in the twitter feed dynamically in real time. This also includes notifications when a user follows another user.

REST API/JSON:

The backend server has been implemented using the Suave package for F# and for JSON messages we have used Newtonsoft.Json. We have created different API Routes as required by each functionality listed down below.

POST REQUESTS: These results of these requests are reflected in the Live Twitter Feed due to websockets

- /register - registers user after receiving Username and Password from client
- /login - logins after comparing password with stored password
- /logout - removes user from loggedInUsers Dictionary
- /posttweet - sends a new tweet to be posted which then gets parsed by the parser actor into the respective mention dictionary and hashtag dictionary.
- /retweet - instructs the server to add the latest tweet from user's following tweet list to user's tweet list.
- /follow - Adds user to followee's follower list

GET REQUESTS:

- /fetchtweets/username - retrieves all the tweets and retweets posted by user and all the people the user is following and sends them to client
- /fetchmentions/username - retrieves all the tweets where the user is mentioned and sends them to client
- /fetchhashtags/username/hashtag - retrieves all the tweets with the specific hashtag asked for and sends them to the client.

JSON OBJECTS:

From Client (depends on request type):

- Register/Login {
 Username: string
 Password: string
}
- Logout {
 Username: string
}
- Tweet/Retweet {
 Username: string
 Tweet: string
}
- Follow {
 Follower: string
 Following: string
}

Server Response:

```
ResponseMessage {  
    UserID: string  
    Message: string       // message to client which is displayed in case of error  
    Content: list<string> // tweet list when fetch request is made by client  
    Check: int            // user status (loggedIn, notLoggedIn, notRegistered)  
    Error: bool           // true when error is detected  
}
```