



Blue Bird Mockery

Lohit Bhambri(*lohit.bhambri@ufl.edu*)

Imthiaz Hussain (*imthiazh.hussain@ufl.edu*)

COP5615 The goal of this project is to develop a Twitter *"like"* simple functions in Erlang

How to run the project

We are using distributed erlang functionality to achieve a client-server model. To store the information we are using `mnesia` which is an `object store` that store information in `key-value record pair`.

Currently, we are using a single system to perform the simulation. We can use multiple system as well, we just need to set cookie when we simulate both client and server using command
`erlang:set_cookie(<<cookie-name>>)`

Initialize a server

We have a single server node

Server@127.0.0.1

```
erl -name Server@127.0.0.1
```

This will start the server node, now we have to start the server

```
c(simpleServer).  
simpleServer:start().
```

Initialize a client

Here we have simulated four client nodes

Client-abc@127.0.0.1

```
erl -name Client-abc@127.0.0.1
```

This will start the client node, now we have to start the client

```
c(simpleClient).  
simpleClient:sendMessage().
```

Client-pqr@127.0.0.1

```
erl -name Client-pqr@127.0.0.1
```

This will start the client node, now we have to start the client

```
c(simpleClient).  
simpleClient:sendMessage().
```

Client-stu@127.0.0.1

```
erl -name Client-stu@127.0.0.1
```

This will start the client node, now we have to start the client

```
c(simpleClient).  
simpleClient:sendMessage().
```

Client-vwx@127.0.0.1

```
erl -name Client-vwx@127.0.0.1
```

This will start the client node, now we have to start the client

```
c(simpleClient).  
simpleClient:sendMessage().
```

Here is the output on our server once the handshake is successfully established between client and server

```
Eshell V13.0.4 (abort with ^G)  
(Server@127.0.0.1)1> c(simpleServer).  
{ok,simpleServer}  
(Server@127.0.0.1)2> simpleServer:start().  
true  
(Server@127.0.0.1)3> Main Loop Listening Scope  
  
(Server@127.0.0.1)3> Message is "Hello"  
(Server@127.0.0.1)3> Main Loop Listening Scope  
  
(Server@127.0.0.1)3> Message is "Hello"  
(Server@127.0.0.1)3> Main Loop Listening Scope  
  
(Server@127.0.0.1)3> Message is "Hello"  
(Server@127.0.0.1)3> Main Loop Listening Scope  
  
(Server@127.0.0.1)3> Message is "Hello"  
(Server@127.0.0.1)3> Main Loop Listening Scope
```

Register the clients

Once server and client relationship is established between the client and server nodes, our first step will be to register client.

Users will be prompted to provide preferred `username` and `password`.

Here is the result for the registering user `abc`

```
Eshell V13.0.4 (abort with ^G)
(Client-abc@127.0.0.1)1> c(simpleClient).
{ok,simpleClient}
(Client-abc@127.0.0.1)2> simpleClient:sendMessage().
Welcome to Twitter
1. Register
2. Login
Enter Register or Login: "Register".
You entered "Register"
Enter UserId: "abc".
Enter Password: "abc123".
Entered UserId is "abc"
Entered Password is "abc123"
User is successfully registered
User Logged In Successfully
Inside Twitter
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets:
```

Output once our users are registered on the server

```
(Server@127.0.0.1)3> Registering the user
(Server@127.0.0.1)3> Entry is []
(Server@127.0.0.1)3> Successfully Registerd UserId  "abc"
(Server@127.0.0.1)3> Main Loop Listening Scope

(Server@127.0.0.1)3> Entry is [{users,"abc","abc123"}]
(Server@127.0.0.1)3> UserId "abc" logged in
(Server@127.0.0.1)3> Upserting the pair in userActor list
(Server@127.0.0.1)3> UserActor Entry is [{userActor,"abc",<16234.85.0>}]
(Server@127.0.0.1)3> Main Loop Listening Scope

(Server@127.0.0.1)3> Registering the user
(Server@127.0.0.1)3> Entry is []
(Server@127.0.0.1)3> Successfully Registerd UserId  "pqr"
(Server@127.0.0.1)3> Main Loop Listening Scope

(Server@127.0.0.1)3> Entry is [{users,"pqr","pqr123"}]
(Server@127.0.0.1)3> UserId "pqr" logged in
(Server@127.0.0.1)3> Upserting the pair in userActor list
(Server@127.0.0.1)3> UserActor Entry is [{userActor,"pqr",<16235.85.0>}]
(Server@127.0.0.1)3> Main Loop Listening Scope

(Server@127.0.0.1)3> Registering the user
(Server@127.0.0.1)3> Entry is []
(Server@127.0.0.1)3> Successfully Registerd UserId  "stu"
(Server@127.0.0.1)3> Main Loop Listening Scope

(Server@127.0.0.1)3> Entry is [{users,"stu","stu123"}]
(Server@127.0.0.1)3> UserId "stu" logged in
(Server@127.0.0.1)3> Upserting the pair in userActor list
(Server@127.0.0.1)3> UserActor Entry is [{userActor,"stu",<16236.85.0>}]
(Server@127.0.0.1)3> Main Loop Listening Scope

(Server@127.0.0.1)3> Registering the user
(Server@127.0.0.1)3> Entry is []
(Server@127.0.0.1)3> Successfully Registerd UserId  "vwx"
(Server@127.0.0.1)3> Main Loop Listening Scope

(Server@127.0.0.1)3> Entry is [{users,"vwx","vwx123"}]
(Server@127.0.0.1)3> UserId "vwx" logged in
(Server@127.0.0.1)3> Upserting the pair in userActor list
(Server@127.0.0.1)3> UserActor Entry is [{userActor,"vwx",<16237.85.0>}]
(Server@127.0.0.1)3> Main Loop Listening Scope
```

Follow the user

Here users `pqr` and client `stu` will follow user `abc` . User `vwx` will not follow anyone right now.

Here is an example of user `pqr` requesting to follow user `abc`

```
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets: "Follow".
You entered "Follow"
Follow the users
Enter the follow-Id: "abc".
Follow the users
Enter the follow-Id: "Done".
Inside Twitter
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets:
```

Output once our user `pqr` and `stu` is following user `abc`

```
(Server@127.0.0.1)3> Inside addFollow list
(Server@127.0.0.1)3> New List is ["pqr"]
(Server@127.0.0.1)3> Updated Entry is [{follow,"abc",["pqr"]}]]
(Server@127.0.0.1)3> Main Loop Listening Scope

(Server@127.0.0.1)3> Inside addFollow list
(Server@127.0.0.1)3> New List is ["stu","pqr"]
(Server@127.0.0.1)3> Updated Entry is [{follow,"abc",["stu","pqr"]}]]
(Server@127.0.0.1)3> Main Loop Listening Scope
```

Post a tweet

Here user `abc` will post a tweet, it will be visible to both user `pqr` and user `stu` once they will check their tweet

Client-abc@127.0.0.1

This client will post two tweets

```
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets: "Tweets".
You entered "Tweets"
What do you want to-do?
1. Feed
2. Post
Your Input: "Post".
You entered "Post"
Write your thoughts here: "Hello World from abc user".
Message Posted
Your post is : "Hello World from abc user"
Inside Twitter
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets: "Tweets".
You entered "Tweets"
What do you want to-do?
1. Feed
2. Post
Your Input: "Post".
You entered "Post"
Write your thoughts here: "I hope you are alright".
Message Posted
Your post is : "I hope you are alright"
Inside Twitter
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets:
```

The tweets will be added to the message-box of user `pqr` and user `stu`

Server@127.0.0.1

```

(Server@127.0.0.1)3> Inside the getFollow function
(Server@127.0.0.1)3> Follower Id is "stu"
(Server@127.0.0.1)3> Entry is {atomic,[{twitterWall,"stu",[]}]}}
(Server@127.0.0.1)3> Message Box []
(Server@127.0.0.1)3> Updated Message-Box is ["Hello World from abc user"]
(Server@127.0.0.1)3> Upserted Message is {atomic,
                                     {twitterWall,"stu",["Hello World from abc user"]}}
(Server@127.0.0.1)3> Feed Check is {atomic,[{twitterWall,"stu",["Hello World from abc user"]}]}}
(Server@127.0.0.1)3> Follower Id is "pqr"
(Server@127.0.0.1)3> Entry is {atomic,[{twitterWall,"pqr",[]}]}}
(Server@127.0.0.1)3> Message Box []
(Server@127.0.0.1)3> Updated Message-Box is ["Hello World from abc user"]
(Server@127.0.0.1)3> Upserted Message is {atomic,
                                     {twitterWall,"pqr",["Hello World from abc user"]}}
(Server@127.0.0.1)3> Feed Check is {atomic,[{twitterWall,"pqr",["Hello World from abc user"]}]}}
(Server@127.0.0.1)3> Main Loop Listening Scope

(Server@127.0.0.1)3> Inside the getFollow function
(Server@127.0.0.1)3> Follower Id is "stu"
(Server@127.0.0.1)3> Entry is {atomic,[{twitterWall,"stu",["Hello World from abc user"]}]}}
(Server@127.0.0.1)3> Message Box ["Hello World from abc user"]
(Server@127.0.0.1)3> Updated Message-Box is ["I hope you are alright","Hello World from abc user"]
(Server@127.0.0.1)3> Upserted Message is {atomic,
                                     {twitterWall,"stu",
                                      ["I hope you are alright",
                                       "Hello World from abc user"]}}
(Server@127.0.0.1)3> Feed Check is {atomic,[{twitterWall,"stu",
                                      ["I hope you are alright",
                                       "Hello World from abc user"]}]}}
(Server@127.0.0.1)3> Follower Id is "pqr"
(Server@127.0.0.1)3> Entry is {atomic,[{twitterWall,"pqr",["Hello World from abc user"]}]}}
(Server@127.0.0.1)3> Message Box ["Hello World from abc user"]
(Server@127.0.0.1)3> Updated Message-Box is ["I hope you are alright","Hello World from abc user"]
(Server@127.0.0.1)3> Upserted Message is {atomic,
                                     {twitterWall,"pqr",
                                      ["I hope you are alright",
                                       "Hello World from abc user"]}}
(Server@127.0.0.1)3> Feed Check is {atomic,[{twitterWall,"pqr",
                                      ["I hope you are alright",
                                       "Hello World from abc user"]}]}}
(Server@127.0.0.1)3> Main Loop Listening Scope

(Server@127.0.0.1)3>

```

Client-pqr@127.0.0.1

This client will check the tweets available in the message-box.

```
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets: "Tweets".
You entered "Tweets"
What do you want to-do?
1. Feed
2. Post
Your Input: "Feed".
You entered "Feed"
The users message box is ["I hope you are alright",
                           "Hello World from abc user"]
Feed is : "I hope you are alright"
Do you want to retweet this message (Yes/No): "No".
Feed is : "Hello World from abc user"
Do you want to retweet this message (Yes/No): "No".
Retweet Box is []
Inside Twitter
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets:
```

Client-stu@127.0.0.1

This client will check the tweets available in the message-box.

```
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets: "Tweets".
You entered "Tweets"
What do you want to-do?
1. Feed
2. Post
Your Input: "Feed".
You entered "Feed"
The users message box is ["I hope you are alright",
                           "Hello World from abc user"]
Feed is : "I hope you are alright"
Do you want to retweet this message (Yes/No): "No".
Feed is : "Hello World from abc user"
Do you want to retweet this message (Yes/No): "No".
Retweet Box is []
Inside Twitter
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets:
```


Post Tweet to the user who is not following using @ annotation

Our user-id is a three character user-id, so in order to post tweet to non-following user, we need to use following annotation in our tweets @<user-id>

User vwx will post a tweet and use annotation @abc . This tweet will be available inside the message box of user abc .

Client-vwx@127.0.0.1

```
Inside Twitter
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets: "Tweets".
You entered "Tweets"
What do you want to-do?
1. Feed
2. Post
Your Input: "Post".
You entered "Post"
Write your thoughts here: "Hello @abc".
Message Posted
Your post is : "Hello @abc"
Inside Twitter
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets:
```

The message will be available inside the message box of user abc

Server@127.0.0.1

```
(Server@127.0.0.1)3> Annotated SubStrng is "abc"
(Server@127.0.0.1)3> Follower Id is "abc"
(Server@127.0.0.1)3> Entry is {atomic,[{twitterWall,"abc",[]}]}}
(Server@127.0.0.1)3> Message Box []
(Server@127.0.0.1)3> Updated Message-Box is ["Hello @abc"]
(Server@127.0.0.1)3> Upserted Message is {atomic,{twitterWall,"abc",["Hello @abc"]}}
(Server@127.0.0.1)3> Feed Check is {atomic,[{twitterWall,"abc",["Hello @abc"]}]}}
(Server@127.0.0.1)3> Inside the getFollow function
(Server@127.0.0.1)3> Nothing to post
(Server@127.0.0.1)3> Main Loop Listening Scope
```

Retweet the tweet

Now user abc will retweet the message available inside it's message queue. This message will be added inside the message-box of user pqr and user stu , who are the followers of user abc

Client-abc@127.0.0.1

```

Enter Follow or Tweets: "Tweets".
You entered "Tweets"
What do you want to-do?
1. Feed
2. Post
Your Input: "Feed".
You entered "Feed"
The users message box is ["Hello @abc"]
Feed is : "Hello @abc"
Do you want to retweet this message (Yes/No): "Yes".
Retweet Box is ["Hello @abc"]
Inside Twitter
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets:

```

The re-tweeted message will be available inside the message box of user `pqr` and user `stu`

Server@127.0.0.1

```

(Server@127.0.0.1)3> Inside the getFollow function
(Server@127.0.0.1)3> Follower Id is "stu"
(Server@127.0.0.1)3> Entry is {atomic,[{twitterWall,"stu",[]}]}}
(Server@127.0.0.1)3> Message Box []
(Server@127.0.0.1)3> Updated Message-Box is ["Hello @abc"]
(Server@127.0.0.1)3> Upserted Message is {atomic,{twitterWall,"stu",["Hello @abc"]}}
(Server@127.0.0.1)3> Feed Check is {atomic,[{twitterWall,"stu",["Hello @abc"]}]}}
(Server@127.0.0.1)3> Follower Id is "pqr"
(Server@127.0.0.1)3> Entry is {atomic,[{twitterWall,"pqr",[]}]}}
(Server@127.0.0.1)3> Message Box []
(Server@127.0.0.1)3> Updated Message-Box is ["Hello @abc"]
(Server@127.0.0.1)3> Upserted Message is {atomic,{twitterWall,"pqr",["Hello @abc"]}}
(Server@127.0.0.1)3> Feed Check is {atomic,[{twitterWall,"pqr",["Hello @abc"]}]}}
(Server@127.0.0.1)3> Main Loop Listening Scope

(Server@127.0.0.1)3> ***Chk if the message box is empty now {atomic,[]}
(Server@127.0.0.1)3> Main Loop Listening Scope

```

This retweet will be added inside the message box of user `pqr` and user `stu` and will be visible to both users respectively

Client-pqr@127.0.0.1

```
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets: "Tweets".
You entered "Tweets"
What do you want to-do?
1. Feed
2. Post
Your Input: "Feed".
You entered "Feed"
The users message box is ["Hello @abc"]
Feed is : "Hello @abc"
Do you want to retweet this message (Yes/No): "No".
Retweet Box is []
Inside Twitter
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets:
```

Client-stu@127.0.0.1

```
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets: "Tweets".
You entered "Tweets"
What do you want to-do?
1. Feed
2. Post
Your Input: "Feed".
You entered "Feed"
The users message box is ["Hello @abc"]
Feed is : "Hello @abc"
Do you want to retweet this message (Yes/No): "No".
Retweet Box is []
Inside Twitter
What do you want to do?
1. Follow
2. Tweets
Enter Follow or Tweets:
```