Company Name: Upao
CS 326
11/30/2016
Team Members: Shengbo Lou, Weijie Shi, Yucheng Lu, Siyu Peng, Robert Liu, Wei Xie

Adding a Web Server

HTTP Requests / Routes

Here list all the HTTP routes implemented in the server.

GET /user/:userId/feed

Purpose: Retrieve the trend post feed data

Mock Server Method: getPostFeedData(userId)

Authorization: All users can access trend post feed data of others on the profile page of a specific user.

POST /postItem [postdata], where [postdata] is a JSON object describing a trend post item.

Purpose: Create a new /postItem/:postItemId

Mock Server Method: postStatus(user, text, cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

PUT /postItem/:postItemId/likelist/:userId userId

Purpose: Add the userId to the likelist of the postItem with postItemId

Mock Server Method: likePost(feedItemId, user, cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

DELETE /postItem/:postItemId/likelist/:userId

Purpose: Remove the userId from the likelist of the postItem with postItemId

Mock Server Method: unLikePost(feedItemId, user, cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

GET /user/:userId

Purpose: Retrieve the user data

Mock Server Method: getUserData(user,cb)

Authorization: All users can access data of others on the profile page of a specific user.

POST /postItem/:postItemId/commentThread/comment comment

Purpose: Create a new /postItem/:postItemId/commentThread/comment

Mock Server Method: adpostComment(activityId, author, comment, cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

PUT /settings/user/:userId [userdata], where [userdata] is a JSON object describing a user.

Purpose: Update /settings/user/:userId and /user/:userId

Mock Server Method: changeUserInfo(data, cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

PUT /settings/emailChange/user/:userId emailaddress

Purpose: Update /settings/user/:userId/email and /user/:userId/email

Mock Server Method: changeEmail(data,cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

PUT /settings/location/user/:userId location

Purpose: Update /settings/user/:userId/location and /user/:userId/location

Mock Server Method: setlocation(userId,location)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

GET /user/:userid/activity

Purpose: Retrieve the activity feed data

Mock Server Method: getActivityFeedData(user,cb)

Authorization: All users can access activity feed data of others on the profile page of a specific user.

POST /postActivity [activitydata], where [activitydata] is a JSON object describing an activity post.

Purpose: Create a new /postActivity

Mock Server Method: createActivity(data,cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

GET /activityItem/:activityId

Purpose: Retrieve the activity detail data

Mock Server Method: getActivityDetail(id,cb)

Authorization: All users can access activity detail data through the link of the activity.

PUT /activityItem/:activityId/likelist/:userId userId

Purpose: Add the userId to the likelist of the activityItem with activityId

Mock Server Method: likeActivity(activityId, user, cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

DELETE /activityItem/:activityId/likelist/:userId

Purpose: Remove the userId from the likelist of the activityItem with activityId

Mock Server Method: unLikeActivity(activityId, user, cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

POST /activityItem/:activityId/commentThread/comment comment

Purpose: Create a new /activityItem/:activityId/commentThread/comment

Mock Server Method: adpostComment(activityId, author, comment, cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

GET /user/:userId/notification

Purpose: Retrieve the notification data

Mock Server Method: getNotificationData(user, cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

PUT /notification/:notificationId/:userId userId, where the first userId is of the receiver and the later one is of the sender

Purpose: Add the later user to the friend list of the former user

Mock Server Method: acceptRequest(id,user,cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

DELETE /notification/:notificationId/:userId

Purpose: Remove the notification of notificationId from the notification list of the user with userId

Mock Server Method: deleteNotification(id, user ,cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

GET /user/:userId/chatsession/:id

Purpose: Retrieve the chat message

Mock Server Method: getMessages(userid,id,cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

POST /user/:userid/chatsession/:id [message], where [message] is is a JSON object describing a chat message.

Purpose: Update the chat message between the users with userid and id

Mock Server Method: postMessage(sessionId,sender,target, text, cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

GET /getsession/:userid/:targetid

Purpose: Retrieve the chat session between users with userid and targetid

Mock Server Method: getSessionId(userid, targetid, cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

GET /search/userid/:userid/querytext/:querytext

Purpose: Retrieve the search results of querytext

Mock Server Method: searchquery(userid, querytext, cb)

Authorization: The body of the HTTP request contains an ”author" field containing a user ID. The requester must have the same user ID.

Special Server Setup Procedure

N/A. There are no additional setup besides `npm install` and `node src/server.js`.

Contribution

Here list all the startup founders with their contributions on product features, and each feature corresponds to its own routes.

Robert Liu:

Profile

Shengbo Lou:

Trend

Settings

Siyu Peng:

Notification

Yuchen Lu:

Chat

Activity Detail

Wei Xie:

Search

Weijie Shi:

Post Activity

Lingering Bugs / Issues / Dropped Features

1. Unable to add new friends
2. Unable to join/sign up an activity
3. Unable to upload image while creating activity

4. Unable to log out

The bugs are mainly due to the problem of mock database and authentication. Once the real database is set and the authentication is applied, we can fix the problems.