Company Name: Upao

CS 326

12/12/2016

Team Members: Shengbo Lou, Weijie Shi, Yucheng Lu, Siyu Peng, Robert Liu, Wei Xie

Adding a Database

HTTP route and detail:

GET /user/:userId/feed

Purpose:

Retrieve the trend post feed data from mongoDB

Client.server.method:

getPostFeedData(user,cb)

Server.server:

app.get('/user/:userId/feed',cb)

getPostFeedData(user,cb)

getPostFeedItem(feedItemId,cb)

MongoDB:

db.collection('users').findOne()

db.collection('postFeeds').findOne()

db.collection('postFeedItems').findOne()

Authorization:

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

Contributor:

Shengbo Lou,Robert Liu

POST /postItem [postdata], where [postdata] is a JSON object describing a

trend post item.

Purpose:

Createa new postItem into mongoDB

Client.server.method:

postStatus(user,text,cb)

Server.server:

app.get('/postItem',{postdata},cb)

postStatus(user,text,location,img,cb)

MongoDB:

db.collection('postFeedItems').insertOne()

Authorization:

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

Contributor:

Shengbo Lou

PUT /postItem/:postItemId/likelist/:userId

Purpose:

Add the userId to the likelist of the postItem with postItemId

Client.server.method:

likePost(feedItemId, user, cb)

Server.server:

app.put('/postItem/:postItemId/likelist/:userId', cb)

MongoDB:

db.collection('postFeedItems').updateOne()

Authorization:

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

Contributor:

Shengbo Lou

DELETE /postItem/:postItemId/likelist/:userId

Purpose:

Remove the userId from the likelist of the postItem with postItemId

Client.server.method:

unLikePost(feedItemId, user, cb)

Server.server:

app.delete(‘/postItem/:postItemId/likelist/:userId’,cb)

MongoDB:

db.collection(‘postFeedItems’).updateOne()

Authorization:

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

Contributor:

Shengbo Lou

GET /user/:userId

Purpose:

Retrieve the user data

Client.server.method:

getUserData(user,cb)

Server.server:

app.get('/user/:userId',cb)

getUserData(userId,cb)

resolveUserObjects(userData.friends,cb)

resolveSessionObject(userData.sessions,cb)

MongoDB:

db.collection('users').findOne()

db.collection('users').find()

db.collection('messageSession').find()

Authorization:

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

Contributor:

Shengbo Lou,Robert Liu

POST /postItem/:postItemId/commentThread/comment comment

Purpose:

post a new comment to a postitem

Client.server.method:

adpostComment(activityId, author, comment, cb)

Server.server:

app.post('/postItem/:postItemId/commentThread/comment',cb)

getPostFeedItem(PostItemId,cb)

MongoDB:

db.collection('postFeedItems').updateOne()

db.collection('postFeedItems').findOne()

Authorization:

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

Contributor:

Shengbo Lou

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

describing a user.

Purpose:

change user Infomation

Client.server.method:

changeUserInfo(data, cb)

Server.server:

app.put('/settings/user/:userId',{userdata},cb)

getUserData(userId,cb)

MongoDB:

db.collection('users').updateOne()

db.collection('users').findOne()

Authorization:

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

Contributor:

Shengbo Lou

PUT /settings/emailChange/user/:userId,{emailaddress}

Purpose:

change User email address

Client.server.method:

changeEmail(data,cb)

Server.server:

app.put('/settings/emailChange/user/:userId emailaddress',{emailaddress},cb)

getUserData(userId,cb)

MongoDB:

db.collection('users',cb).updateOne()

Authorization:

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

Contributor:

Shengbo Lou

PUT /settings/avatar/user/:userId

Purpose:

change User photo

Client.server.method:

ChangeAvatar(user,img,cb)

Server.server:

app.put('/settings/avatar/user/:userId', cb)

MongoDB:

db.collection('users').findAndModify()

Authorization:

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

Contributor:

Shengbo Lou

PUT /settings/location/user/:userId location

Purpose:

update user current location

Client.server.method:

getlocation(cb)

use Google geolocation library

setlocation(userId,location)

Server.server:

app.put('/settings/location/user/:userId',cb)

MongoDB:

db.collection('users').updateOne()

Authorization:

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

Contributor:

Shengbo Lou

GET /user/:userid/activity

Purpose:

Retrieve the activity feed data

Client.server.method:

getActivityFeedData(user,cb)

Server.server:

app.get('/user/:userid/activity',cb)

getActivityFeedData(userId,cb)

getActivityFeedItem(activityId, cb)

MongoDB:

db.collection('users').findOne()

db.collection('activities').findOne()

db.collection('activityItems').findOne()

Authorization:

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

Contributor:

Yuchen Lu,Robert Liu

POST /postActivity [activitydata] , where [activitydata] is a JSON object

describing an activity post.

Purpose:

Create a new postActivity

Client.server.method:

createActivity(data,cb)

Server.server:

app.post('/postActivity',{activitydata},cb)

postActivity(body,cb)

MongoDB:

db.collection('activityItems').insertOne()

db.collection('users').findOne()

db.collection('activities').updateOne()

Authorization:

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

Contributor:

Weijie Shi

GET /activityItem/:activityId

Purpose:

Retrieve the activity detail data

Client.server.method:

getActivityDetail(id,cb)

Server.server:

app.get('/activityItem/:activityId',cb)

getActivityFeedItem(activityId,cb)

MongoDB:

db.collection('activityItems').findOne()

Authorization:

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

Contributor:

Yuchen Lu

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

Purpose:

Add the userId to the likelist of the activityItem with activityId

Client.server.method:

likeActivity(activityId, user, cb)

Server.server:

app.put('/activityItem/:activityId/likelist/:userId',cb)

MongoDB:

db.collection('activityItems').findAndModify()

Authorization:

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

Contributor:

Yuchen Lu

DELETE /activityItem/:activityId/likelist/:userId

Purpose:

Remove the userId from the likelist of the activityItem with activityId

Client.server.method:

unLikeActivity(activityId, user, cb)

Server.server:

app.delete('/activityItem/:activityId/likelist/:userId',cb)

MongoDB:

db.collection('activityItems').findAndModify()

Authorization:

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

Contributor:

Yuchen Lu

POST /activityItem/:activityId/commentThread/comment comment

Purpose:

post a new activity comment

Client.server.method:

adpostComment(activityId, author, comment, cb)

Server.server:

getActivityFeedItem(activityItemId,cb)

MongoDB:

db.collection('activityItems').updateOne()

db.collection('activityItems').findOne()

Authorization:

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

Contributor:

Yuchen Lu

GET /user/:userId/notification

Purpose:

Retrieve the notification data

Client.server.method:

getNotificationData(user, cb)

Server.server:

app.get('/user/:userId/notification',cb)

getNotificationData(notificationId,cb)

getNotificationItem(notificationId,cb)

MongoDB:

db.collection('notifications').findOne()

db.collection('notificationItems').findOne()

Authorization:

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

Contributor:

Siyu Peng

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

Client.server.method:

acceptRequest(id,user,cb)

Server.server:

app.put('/notification/:notificationId/:userId',cb)

getNotificationItem(notificationId,cb)

MongoDB:

db.collection('users').updateOne()

db.collection('notificationItems').findOne()

Authorization:

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

Contributor:

Siyu Peng

DELETE /notification/:notificationId/:userId

Purpose:

Remove the notification of notificationId from the notification list

of the user with userId

Client.server.method:

deleteNotification(id, user ,cb)

Server.server:

app.delete('/notification/:notificationId/:userId',cb)

getNotificationData()

getNotificationData(notificationId,cb)

getNotificationItem(notificationId,cb)

MongoDB:

db.collection('users').findOne()

db.collection('notifications').updateOne()

db.collection('notificationItems').remove()

db.collection('notifications').findOne()

db.collection('notificationItems').findOne()

Authorization:

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

Contributor:

Siyu Peng

GET /user/:userId/chatsession/:id

Purpose:

Retrieve the chat message

Client.server.method:

getMessages(userid,id,cb)

Server.server:

app.get('/user/:userId/chatsession/:id',cb)

getMessage(sessionId, cb)

MongoDB:

db.collection('messageSession').findOne()

db.collection('message').findOne()

Authorization:

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

Contributor:

Yuchen Lu

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

Client.server.method:

postMessage(sessionId,sender,target, text, cb)

Server.server:

app.post('/user/:userid/chatsession/:id',[message],cb)

getMessage(sessionId, cb)

MongoDB:

db.collection('message').updateOne()

db.collection('message').findOne()

Authorization:

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

Contributor:

Yuchen Lu

GET /getsession/:userid/:targetid

Purpose:

Retrieve the chat session between users with userid and targetid

Client.server.method:

getSessionId(userid, targetid, cb)

Server.server:

app.get('/getsession/:userid/:targetid',cb)

getSessionId(userid, targetid, cb)

MongoDB:

db.collection("messageSession").findOne()

Authorization:

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

Contributor:

Yuchen Lu

POST /resetdb

Purpose:

resetdatabase

Server.server:

app.post('/resetdb',cb)

ResetDatabase(db,cb)

resetdatabase in mongoDB

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

Purpose:

Search users{ firstname,lastname } activityItems{description}

postFeedItems{text}. Use regular expression to match the query text

Client.server.method:

searchquery(userid,querytext,cd)

Server.server:

app.get('/search/userid/:userid/querytext/:querytext',cb)

MongoDB:

db.collection('users').find()

db.collection('activityItems').find()

db.collection('postFeedItems').find()

Authorization:

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

Contributor:

Wei Xie

We implemented most of the features including authentication and web socket, please refer to branch “socket” for more detail