Project Milestone #3a

AWS Cognito w/ react-native

Rest API

API Endpoint

1. Sign up – amplify
2. Sign in – amplify

None of them are authenticated.

No request headers nor response headers.

Every endpoint use same url:

[https://vxmpyha4gzamvmf5xntbe5f5b4.appsync-api.us-west-2.amazonaws.com/](https://vxmpyha4gzamvmf5xntbe5f5b4.appsync-api.us-west-2.amazonaws.com/graphql)

Every error response is same

400: A JSON string containing error information

500: A JSON string containing error information

Example

post = async () => {

console.log('calling api');

const response = await API.post('myapi', '/items', {

body: {

id: '1',

name: 'hello amplify!'

}

});

alert(JSON.stringify(response, null, 2));

};

get = async () => {

console.log('calling api');

const response = await API.get('myapi', '/items/object/1');

alert(JSON.stringify(response, null, 2));

};

list = async () => {

console.log('calling api');

const response = await API.get('myapi', '/items/1');

alert(JSON.stringify(response, null, 2));

};

or

API.get('sampleCloudApi', '/items?q=test');

1. Follow a user

**Method**: put

**Path**: /users/{user.alias}/follower

**Description**: Returns nothing

Use user’s alias as a key to search and update user’s follow.

Append alias of following user.

**Response**:

200: A JSON string containing success information

1. Add followee

**Method**: put

**Path**: /users/{user.alias}/followee

**Description**: Returns nothing

Use follower’s alias as a key to search and update follower’s followee.

Append alias of user.

**Response:**

200: A JSON string containing success information

1. Get user’s follower

**Method**: get

**Path**: /users/{user.alias}/follower

**Description**: Returns a page of followers for a user

Use alias as a key to search and load user’s follower

**Response**:

200: A JSON string containing an array of followers with the following format: [ ... ]

1. Get user’s followee

**Method**: get

**Path**: /users/{user.alias}/followee

**Description**: Returns a page of followee for a user

Use alias as a key to search and load user’s followee

**Response**:

200: A JSON string containing an array of followees with the following format: [ ... ]

1. Post a status

**Method**: post

**Path**: /status

**Description**: Returns nothing

Add status with the user’s alias as a key in the status table

**Response:**

200: A JSON string containing success information

1. Get user’s story

**Method**: get

**Path**: / status?alias={user.alias}

**Description**: Returns all status belong to user

Use alias as a key to search and load user’s story

Find all the status that alias attribute is user.alias

**Response:**

200: A JSON string containing an array of status with the following format: [ ... ]

1. Get user’s Feed

**Method**: get

**Path**: /status?alias={user.alias}

**Description**: Returns all status user follows

Use alias as a key to search and load user’s feed

Find all the status that alias attribute is user.alias(follower’s alias)

**Response:**

200: A JSON string containing an array of status with the following format: [ ... ]

1. Get hash related status

**Method**: get

**Path**: /status?hashtag={hashtag}

**Description:** Returns all status that has same hashtag

Use hashtag as a key to search and load statuses

**Response:**

200: A JSON string containing an array of status with the following format: [ ... ]

1. Edit profile

**Method**: put

**Path**: /users/{user.alias}/picture

**Description**: Returns nothing

Use alias as a key to search user and upload picture

**Response:**

200: A JSON string containing success information