**User**

<https://ah-jeez.herokuapp.com/users>

id: DataTypes.UUIDV4 //primary key

roomId: DataTypes.UUIDV4, //foreign key - room the user is in

fbId: DataTypes.STRING, //facebook id (unique)

fbToken: DataTypes.STRING, //facebook auth token (unique)

name: DataTypes.STRING, //username

pic: DataTypes.STRING, // profile pic url

createdAt: DataTypes.DATE,

updatedAt: DataTypes.DATE

**Sending auth tokens**

Just 2 headers - Content-Type and Authorization. Example below, note Authorization’s value always has the word ‘Token’, then a space, and finally the token.

Content-Type: application/json

Authorization: Token QURTOkFERg==

**Login**

Put in the body what returns from Facebook. If we already have this Facebook id associated with a user, we will update the token. Otherwise, we will create a new user and add whatever is in the body. TODO: validate that the token and Facebook id on the server side.

required: fbToken, fbId

|  |  |
| --- | --- |
| Route: | post: ‘/users/login’ |
| token header? | no |
| Request1 (**creating user**) | {"user":  {  "fbToken": "token1",  "fbId": "fb123",  "pic": "www.sic pics.com",  "name": "name1"  }  } |
| Response1 | {  "success": true,  "user": {  "id": "07c2a12c-aa0a-41db-9309-4bd61e9bec70",  "fbToken": "token1",  "fbId": "fb123",  "pic": "www.sic pics.com",  "name": "name1",  "updatedAt": "2015-02-05T03:10:22.542Z",  "createdAt": "2015-02-05T03:10:22.542Z",  "roomId": null  }  } |
| Request 2 (**updating user**) | {  "user":{  "fbToken": "token2",  "fbId": "fb123"  }  } |
| Response 2 | {  "success": true,  "user": {  "id": "07c2a12c-aa0a-41db-9309-4bd61e9bec70",  "fbId": "fb123",  "fbToken": "token2",  "name": "name1",  "pic": "www.sic pics.com",  "createdAt": "2015-02-05T03:10:22.542Z",  "updatedAt": "2015-02-05T03:11:11.779Z",  "roomId": null  }  } |
| Error response | {  "success": false,  "errors": {  "name": "SequelizeValidationError",  "message": "Validation error",  "errors": [  {  "message": "fbToken cannot be null",  "type": "notNull Violation",  "path": "fbToken",  "value": null  }  ]  }  } |

**List**

Just return a list of all users and their public info.

|  |  |
| --- | --- |
| Route: | get: ‘/users’ |
| token header? | no |
| Response | {  "users": [  {  "id": "07c2a12c-aa0a-41db-9309-4bd61e9bec70",  "roomId": null,  "name": "name1",  "pic": "www.sic pics.com",  "fbId": "fb123"  }  ]  } |

**Read**

Get a users public info with an id

|  |  |
| --- | --- |
| Route: | get: ‘/users/:id’ |
| token header? | no |
| Response | {  "success": true,  "user": {  "id": "07c2a12c-aa0a-41db-9309-4bd61e9bec70",  "roomId": null,  "name": "name1",  "pic": "www.sic pics.com",  "fbId": "fb123"  }  } |

**Delete**

|  |  |
| --- | --- |
| Route: | delete: ‘/users/:id’ |
| token header? | yes |
| Response | {  "success": true  } |