

Milestone 3

API Documentation

Team Members: Zack Watkins, Derek Florimonte, Logan Figgins, Frank Koutulas

Team Name: Sushiisthebest

Team Github: <https://github.com/SFSU-CSC-667/term-project-spring-2017-sushiisthebest>

TABLE OF CONTENTS

- I. Login
- II. Sign Up
- III. User
- IV. Games
- V. Achievements

/login

- a. GET
 - b. /login
 - c. This endpoint is responsible for rendering the view login
-
- a. POST
 - b. /login
 - c. This end point responsible for authenticating the user. The body of the request is compared to the password digest stored in the database. If successful the client sent a token used for authentication and redirected to the games landing page
-
- a. GET
 - b. /queue
 - c. Alternatively, if authentication is correct, the user will make a GET request for the Queue Page, allowing the user to sign up for a game

/signup

- a. GET
 - b. /signup
 - c. This endpoint renders the sign up view
-
- a. POST
 - b. /signup
 - c. At this end point if the user email is unique the user data base is updated with information provided in the request body. update the user database with the new user information for the new account. On success, the client is then redirected to the user/:id.

/user

- a. GET
 - b. /user/:id
 - c. At this endpoint the user database table is queried returning all profile information for the user with id = :id
-
- a. POST
 - b. /user/:id
 - c. The user table is updated to reflect which special abilities, and which character avatar the user has selected to be used in active games.

/games

- a. GET
 - b. /games
 - c. This endpoint returns a list of all the currently available games inactive (still accepting players) that were contained in the database's games table. After the query completes the Games List view is rendered data using the data retrieved from the database
-
- a. GET
 - b. /games/:id
 - c. This endpoint renders game lobby view where players wait together until the host starts the game
-
- a. GET
 - b. /games/:id/active
 - c. This point renders the Game Room view. This view represents the actual gameplay as opposed to the route games/:id which is merely the lobby where players wait for the game to begin
-
- a. PUT ?
 - b. /game/:id
 - c. This endpoint Updates / Creates a game in the game table
-
- a. Web IO
 - b. /game/id/Active
 - c. Opens a TCP connection using sockets IO with the server and the client so that the server and client can send IO messages to each other in order to handle all game mechanic logic. *need to ask Jrob how to represent this in the api*
-
- a. GET
 - b. /game/id/end
 - c. This endpoint renders one of 4 different game end views, based on 4 set game end scenarios and then redirects to the achievement endpoint

/achievement

- a. GET
- b. /achievements/user:id
- c. This endpoint will update the user statistics table with user ID based on any experience gained at the end of the game and experience for achievements.