

Routes:

- /login
  - GET
    - Render page
  - PUT
    - Validate credentials
    - Update online status
    - Retrieve account id
    - Redirect to /selection
- /register
  - GET
    - Render page
  - POST
    - Validate username doesn't already exist
    - Create new account
- /account
  - GET
    - Retrieve required account details
    - Render page
- /images
  - GET
    - Retrieve images
    - Render page
- /account/friends
  - GET
    - Retrieve required friend details
  - DELETE
    - Remove friend
- /account/mantra
  - GET
    - Render page
  - PUT
    - Update mantra
    - Redirect to /account
- /account/:username

- GET
  - Retrieve required account details
  - Render page
- POST
  - Add friend or block target
- DELETE
  - Remove friend or unblock target
- PUT
  - Validate lack of target pending session
  - Update target pending session
  - Delete any existing games
    - Update user rank data
    - Update opponent rank data
  - Create new game
  - Redirect to game/:id
- /achievements
  - GET
    - Retrieve achievement documents
    - Render page
- /highscores
  - GET
    - Retrieve highscore documents
    - Render page
  - PUT
    - Validate target username
    - Redirect to account/:username
- /selection
  - GET
    - Render page
  - POST
    - Retrieve user's current game sessions
    - Redirect to previously played game first
    - If there is no previous game then redirect to pending game
      - Remove user pending session
    - If there is no pending game then retrieve a game session with no player2
      - Update user current session
    - If there is no one else waiting for a game then create a new game document with the user as player1
- /game

- GET
  - Render page
- /game/:id
  - GET
    - Retrieve game data
    - Reset user's pending session
  - PUT
    - Upload ready status
    - Check if both users are ready
      - If so then upload user game data
      - Update game data (such as the game step)
      - Retrieve game data
  - POST
    - Add the achievement just earned (whatever it may be)
  - DELETE
    - Update rank data
    - Remove game document
    - Update both user's current game sessions

How are users paired for a game?

If user1 challenges user2 then user2's challenger becomes user1 (assuming user2 doesn't already have a challenger). User1 is immediately redirected to the game. When user2 selects to play a new game then he will be redirected to the game with user1. However, if user2 already has a game in session (that he left prematurely) then he will be redirected to that game instead. To prevent any conflicts in this process, user1 will be forfeited from his current game (if it exists) if he selects to challenge somebody. The idea is that if a user accidentally leaves the game session, he will not be forfeited immediately.

What is the game step?

The current step of the game will determine what functions the interface for the game has. Step 0 means that there is 1 player waiting for an opponent, the player can set up his board at this point and the opponent's window will say "waiting for opponent". Step 1 means that an opponent just joined the game, on initialization the opponent's ranking and mantra will show up, and now both players can set up their boards and the opponent window on both sides will say "opponent is setting up the board". Step 2 is the main game, and step 3 is the win/lose screen.

Collections:

account document

```
{
  "username" : <string>,
  "password" : <string>,
  "onlineStatus" : <boolean>,
  "birthDate" : <string>,
  "profileImage" : <BinData>,
  "mantra" : <string>,
  "wins" : <int>,
  "losses" : <int>,
  "rank" : <int>,
  "achievementPoints" : <int>,
  "achievementsList" : {
    "name" : <string>,
    "date" : <string>
  },
  "friends" : <string[]>,
  "blocked" : <string[]>,
  "currentSession" : <string>,
  "pendingSession" : <string>
}
```

The accounts collection is used by

- /login to validate credentials
- /register to check existence of username
- /account to populate page
- /account/:username to populate page
- /highscores to check rank data
- /selection to check pending sessions
- /game/:id to check rank data

achievement document

```
{
  "name" : <string>
  "description" : <string>
}
```

```
    "points" : <int>
    "hidden" : <boolean>
}
```

The achievements collection is used by

- /achievements to populate page

game document

```
{
    "_id" : <string>,
    "player1Username" : <string>,
    "player2Username" : <string>,
    "gameStep" : <int>, (connecting, setup, battle, or
    finished)
    "player1Grid" : <int[]>,
    "player2Grid" : <int[]>,
    "player1Ready" : bool,
    "player2Ready" : bool
}
```

The game document is used by

- /selection to check for players waiting for an opponent
- /game/:id to populate the page
- /game/:id to transmit game data between opponents

profileImage document

```
{
    "image" : <BinData>
}
```

The profileImage document is used by

- /images to populate the page