Ashton Alonge
Patrick Whitlock

Technical Specifications for Fantasy Soccer API

User stories/User requirements

Users will be able to:

- Create a fantasy soccer team. Users will be able to add up to 11 players and remove players from their team
- Maximize the performance of their fantasy team given a budget, where each player has a value associated with them
- Create fantasy leagues and compete with other teams
- Add friends
- Get players' stats from games
- Compare fantasy team performances based off player performance
- Log in with a username and password

Endpoints

POST /players

This endpoint adds a player to the database. Each player has the following statistics based on real life performance:

player_id (auto assigned by system)
player_name
irl_team_name
player_position
player_value

PUT /players/{id}/info

This endpoint allows the user to edit the following player info of the player with the specified player_id:

irl_team_name
player_position
player_value

GET /players/{player_id}

This endpoint returns a single player by the specified player_id. For each player, it returns:

player_id
player_name
irl_team_name
player_position
player_value
the following game stats (if the player has played in at least one game):
number of total goals
number of total assists
number of total shots on goal
number of total passes
number of total turnovers

GET /players/

This endpoint lists all players that have played at least 1 game. They are sorted by the specified sort input, and listed up to the limit.

Sort options include:
number of goals
number of assists
number of shots on goal
number of passes
number of turnovers
player value

POST /users

This endpoint creates a new user with a request body that specifies a username, a boolean of whether or not the user is an admin, and a password. The user_id that is automatically assigned by the database.

GET /users

This endpoint returns a list of users. The user can provide:
`user_name` - show users whose user_name matches the given string
`limit` - how many users to show
`offset` - how many users to skip over

POST /users/login

This endpoint takes in a user_name and a password in the request body, and returns whether or not the password was correct.

POST /fantasy_teams/

This endpoint allows users to create a new fantasy team with a specified name, and assigns their user_id to the team. Each team has a name, a fantasy league in which the team competes, and a balance, which is the maximum amount of money that can be used to purchase players. The fantasy league is auto set to null and can be changed in PUT /fantasy_teams/{fantasy_league_id}/join. The balance is auto set to $10,000,000.

POST /fantasy_teams/players

This endpoint allows the user to add a player to their fantasy team roster by player_id. If the team balance is less than the value of the player, or the team is already full (11/11 players), the user will not be able to add the player to their team.

DELETE /fantasy_teams/{fantasy_team_id}/players

This endpoint will remove a player from their fantasy team roster by player_id. The amount of money that the team spent on the player will be added back to the team balance after removing the player.

PUT /fantasy_teams/{fantasy_league_id}/join

This endpoint adds a fantasy team to a fantasy league,which updates the fantasy_league_id column of a team.

GET /fantasy_teams/{fantasy_team_id}/score

This endpoint will return the score of the specified fantasy team, which is a sum of the team's player scores

POST /fantasy_leagues

This endpoint allows the user to create a new fantasy league. User provides the name of the fantasy league. All leagues will be created with a budget of $1,000,000

GET /fantasy_leagues

This endpoint lists fantasy leagues from the database. User can provide:
`fantasy_league_name` - shows leagues whose name matches the given string
`limit` - how many leagues to show
`offset` - how many leagues to skip over

GET /fantasy_leagues/{fantasy_league_id}

This endpoint returns the matching fantasy league by id and shows all the fantasy teams in the league. The teams are sorted by point value, so the highest performing teams will be at the top of the list.

POST /friends

This endpoint adds a friendship between two users. The user provides two user_ids to be added to the table.

GET /friends

This endpoint lists out the friendships between users. The user can also provide:
`user_id` - show the friendships this user_id is a part of.
`limit` - how many users to show
`offset` - how many users to skip over

POST /games/

This endpoint adds a new game to the games table with the following specified information:

the id of the game
the id of the player
number of goals
number of assists
number of shots on goal
number of passes
number of turnovers

GET /games/{game_id}

This endpoint returns the following info of the game specified by the game_id, and for each player that played in that game:

the id of the game
the id of the player
number of goals
number of assists
number of shots on goal
number of passes
number of turnovers