

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.
- View how their team is doing based on current team and players scores.
- View and compare players based on certain statistics

Administrators will be able to:

- Add players to the database with statistics based on real life performances.
- Update player statistics (for after a live game happens)

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
player_position
fantasy_team_name
fantasy_league_name
player_height
player_weight
number_of_games_played
number_of_goals_scored
number_of_assists
number_of_shots
number_of_shots_on_goal
pass_completion

PUT /players/{player_id}/statistics

This endpoint will add to the provided statistics. For example, if a player made 2 goals in a live game, the admin can use this endpoint to add those goals to the player statistics.

GET /players/{player_id}

This endpoint returns a player or list of players whose statistic is equal to, greater than, or less than the input value of the specified statistic. If the input is a string, it returns the player or players whose specified statistic contains the input string

GET /players/top?statistic={statistic_name}&limit={limit}

This endpoint takes as input values the amount of players to display, and which statistic to sort by. It returns a list of players of the specified length sorted in either ascending or descending order of the specified statistic.

POST /users

This endpoint creates a new user with a request body that specifies a username, and a user_id that is automatically assigned

POST /users/{fantasy_league_id}/join

This endpoint allows a user to join a league, which updates the fantasy_league_name of the user

PUT /users/{user_id}/friends

This endpoint allows a user to add a friend by their user_id

POST /fantasy_teams

This endpoint allows users to create a new fantasy team with a specified name. Each team has a name, a fantasy league in which the team competes, a roster of 11 players, and a total score.

POST /fantasy_teams/{fantasy_team_id}/roster/players

This endpoint allows the user to add a player to their fantasy team roster by player_id

DELETE /fantasy_teams/{fantasy_team_id}/players

This endpoint will remove a player from their fantasy team roster by player_id

GET /fantasy_teams/{team_id}

This endpoint will return a summary of the fantasy team score based on the stats of the players in the fantasy team.

GET /fantasy_leagues/{league_id}/teams/leaderboard

This endpoint will return a list of all fantasy teams in a specified league, in descending order of team points.

POST /fantasy_leagues

This endpoint allows the user to create a new fantasy league with a request body that includes the name and start date.

GET /fantasy_leagues/?limit={limit}

This endpoint takes as an input value the number of leagues to list, and lists them in descending order of the score of the highest scoring fantasy team in the league

GET /fantasy_leagues/{fantasy_league_id}

This endpoint returns the matching fantasy league by id and shows all the fantasy teams in the league.

Detailed descriptions of edge cases

1. **POST /players/{player_id}** edge cases:
 - a. User does not provide all the statistics of a player
 - i. Statistics not provided will be null
2. **PUT /players/{player_id}/statistics** edge cases:
 - a. Statistic or player not found
 - i. Return invalid input error (user most likely misspelled the word, or tried to edit a stat or player that doesn't exist)
3. **GET /players/{player_id}**
 - a. Player_id not found
 - i. Return 404 players
4. **GET /players/top?statistic={statistic_name}&limit={limit}** edge cases:

- a. Statistic not provided
 - i. Prompt user to provide statistic to sort by
 - b. limit not provided
 - i. Return top 10 players by provided statistic
- 5. **POST /users** edge cases:
 - a. Username not provided
 - i. Return error about missing username
- 6. **POST /users/{fantasy_league_id}/join** edge cases:
 - a. Fantasy_league_id not found
 - i. Return 404 not found error
- 7. **PUT /users/{user_id}/friends** edge cases:
 - a. User id not found
 - i. Return 404 user not found error
 - b. Friend (different user id) not found
 - i. Return 404 friend not found error
- 8. **POST /fantasy_teams** edge cases:
 - a. Name not provided
 - i. Cannot create team until name is provided
- 9. **POST /fantasy_teams/{fantasy_team_id}/roster/players** edge cases:
 - a. Player or team not found
 - i. Return not found error
- 10. **DELETE /fantasy_teams/{fantasy_team_id}/players** edge cases:
 - a. Id not provided
 - i. Prompt user to provide id
 - b. Player with provided id not found
 - i. Return player not found error
- 11. **GET /fantasy_teams/{team_id}** edge cases:
 - a. Team_id not found
 - i. Return 404 team id not found
- 12. **GET /fantasy-leagues/{league_id}/teams/leaderboard** edge cases:
 - a. League id not found
 - i. Return 404 error
- 13. **POST /fantasy_leagues** edge cases:
 - a. Missing name or start date in the request body
 - i. Return error telling the user to input a name or start date
- 14. **GET /fantasy_leagues/?limit={limit}** edge cases:
 - a. Limit not provided
 - i. List first 10 fantasy leagues
- 15. **GET /fantasy_leagues/{fantasy_league_id}** edge cases:
 - a. Fantasy_league_id not found
 - i. Return 404 not found error