

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

add_player_to_db:

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

player_id
player_name
player_position
team_name
player_height
player_weight
number_of_games_played
number_of_goals_scored
number_of_assists
number_of_shots
number_of_shots_on_goal

update_player_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_player:

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 who's specified statistic contains the input string

create_team:

This endpoint allows users to create a new fantasy team with a specified name

add_player_to_team:

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

remove_player_from_team:

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

compare_player_statistics:

This endpoint allows the user to provide two player_id values and returns two players and a comparison between them. The comparison will encompass the following for each numerical statistic:

- The difference
- The percent difference

list_top_players_by_statistic:

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.

get_team_score:

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

Detailed descriptions of edge cases

1. **add_player_to_db** edge cases:
 - a. User does not provide all the statistics of a player
 - i. Statistics not provided will be null
2. **update_player_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_player** edge cases:
 - a. Statistic not provided
 - i. Prompt user to provide statistic to sort by
 - b. Comparison operator not provided
 - i. Return players using the equal comparison operator
4. **create_team** edge cases:
 - a. Name not provided
 - i. Cannot create team until name is provided
5. **Add_player_to_team** edge cases:
 - a. Player or team not found
 - i. Return not found error
6. **remove_player_from_team** 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
7. **compare_player_statistics** edge cases:
 - a. One or both player ids not found or missing
 - i. Return missing/invalid id error
8. **list_top_players_by_statistic** edge cases:
 - a. Missing number of players to return input
 - i. Display all players
 - b. Statistic not found
 - i. Return the list by player id
9. **Get_team_score** edge cases:
 - a. Team name not found
 - i. Return 404 team not found error