

English Premier League Table Viewer

The **English Premier League** is the top-tier league of soccer (football) teams, which consists of 20 teams. Every year, each team in the league plays 38 matches (2 matches against every other team).

At the end of the season, the team with the most points wins the Premier League title. The bottom three teams are relegated to the Championship League, which is the league one tier below the Premier League. The top 4 teams are eligible to play in the UEFA Champions League the next season, and teams ranked 5 and 6 are eligible to play in the Europa League.

Each win is worth 3 points, each tie is worth 1 point, and a loss does not give a team any points.

We would like you to create a function that generates the Premier League table (ie: the standings).

Given the full list of matches in the season from

<https://github.com/openfootball/football.json/blob/master/2016-17/en.1.json>

your script must create a sorted table containing the following information for each team:

- rank (1, 2, 3 20)
- team name
- wins
- draws
- losses
- goals for
- goals against
- goal difference
- points

The table must be sorted by **rank**. The rank is determined by points, then goal difference, then goals scored.

Optional: The resulting table should be displayed within a Grid Control on a web page. If possible, the grid control should let you sort each of the columns. If the grid control supports it, highlight the UEFA-eligible teams with a light green background the Europa-eligible teams in a light yellow background, and the relegated teams in a light red background.

Note: *If you do not have access to a grid control for this test, you can use a standard HTML table.*

Optional: The data should be stored in a relational database on your laptop. The design of the schema is up to you. The schema should be flexible enough to store the results of matches for future seasons, and perhaps even flexible enough to add new leagues, such as La Liga, Bundesliga, etc. (You may even want to expand it to include information about who scored goals in the matches, but that is not required for this project.)

If you find yourself with extra time at the end of the test, create a RESTful API for your server. It should support fetching the complete table or fetching the data for a certain team. If you have information about the goal scorers, you may want to add REST APIs so that someone could find out who the top goal scorers are.

Requirements for the Project

If you are going to use a database to store the data, then you need to have a relational database installed on your computer. Acceptable databases are SQL Server, MySQL, and Postgres. You should have migration files or scripts so that we can recreate your database.

Rules for Submitting your Project

Use **Git** to check in your code and send us a link to your project.

- We will clone the code from Github, open the solution in Visual Studio Code, and run the solution and run any unit tests.
- The code should be able to compile and run without any modification.
- If you use an external piece of infrastructure (ie: a database), then you should have SQL statements that create the database.
- Any setup steps should be documented in the README file.