

European Soccer Leagues Database

Author: Doniphan Molina. Frontend Developer

Objectives:

Build an app that allows you to see and interact with European Football data. The app must have the following sections:

- **Dashboard:** In this page the user will be able to see all the Leagues, and details from each league (name, number of teams, current match day, total match days, total number games)
- **League Details:** In this page the user will be able to see all the teams that are in an specific league (Name, team crest or logo if provided, points, games lost, games won, games tied), in descending order according to their position in the league, user needs to be able to reverse the sorting to ascending.

When the user clicks on a Team, a modal should come up with more detailed information.

The user can mark the Team as Favorite.

And finally, a link to the Team Matches page.

- **Team Matches Page:** In this page the user will see a list of all the matches for a team, ordered ascending by the match date, user should be able to reverse that sorting.

Information for each match shown should be:

- Some indication of whether the games has been played or not
- Home Team name

- **Favorites Page:** A Page where the user can see a list of the teams he has marked as favorite, he should see the team name, crest, a link to the League Details where the teams belongs to, and a link to the Team Matches Page for the team.

NOTE: This does not need to be persistent, can be forgotten by the app on refresh.

Notes.

- I managed to achieve/simulate some persistence with the list of favorite teams by using the localStorage of the browser.
- The application is responsive-ready and mobile-friendly.
- The tables divisions stack into divs when seeing them in mobile.
- There is a league that has groups so I have to make specific functions to manage its data that is different than the others.
- Some teams doesn't have badges (logos) so I made a directive that checks for this and if false, assigns a default one.
- Some teams doesn't have list of players.
- I used ngAnimate to achieve animations on lists based on ngRepeat.
- I used a little library called ngDialog.js to manage modals.

- I used Bootstrap to manage grid system and some styles.
- I used FontAwesome to get some icons.

Tree

```
Root
|
--css
----bootstrap.min.css
----main_style.css
----ngDialog-theme-default.min.css
----ndDialog.min.css
----ripple.css
----skin-purple.min.css
----style.css
|
--img
----bg.jpg
----congruent_outline.png
----genericBadge.png
----red-news-date-bg.png
--js
|
----controllers
-----controllers.js
|
----libs
-----angular-animate.min.js
-----angular-route.min.js
-----angular-sanitize.min.js
-----angular.min.js
-----ngDialog.min.js
----services
-----services.js
|
----app.js
----routes.js
|
--vendor
----bootstrap
----font-awesome
|
--views
----dashboard.html
----favorites.html
----fixtures.html
----home.html
----league.html
----teamModal.html
|
index.html
readme.md
```

JS Files and its methods

app.js

Includes the main module and its dependencies

routes.js

Includes the routes for every page in the site.

“/” , main route, points to Dashboard.

“/league/:id” , points to league table details.

“/team/fixtures/:id” , points to matches of the team selected,

“/favorites” , points to list of favorite teams.

services.js

Includes the factories and directives used in the controllers.

- **Authorization factory**: makes the request to the API using the token provided.
- **Dashboard factory**: gets list of leagues.
- **LeagueDetail factory**
 - **LeagueDetail.leagues**: get list of leagues
 - **LeagueDetail.league_detail**: gets list of teams in a league with details
 - **LeagueDetail.team_players**: gets list of players of team selected.
 - **LeagueDetail.team_fixtures**: gets list of matches of team selected.
 - **LeagueDetail.team**: gets detail of team selected.
 - **LeagueDetail.league_teams**: gets list of teams in a league
- **Checks factory**
 - **Checks.get_favs**: gets favorited teams from the local storage.
- **checkImage directive**: Checks if team has badge. If don't, assigns a default one.

controllers.js

- **leaguesDashboardCtrl**: puts list of leagues and its details into the scope.
- **leagueCtrl**: puts into the scope a list of teams in a league and its statistics. Also has a function to open ngDialog with specific details of a team clicked. Has functions for favorite and unfavorite a team.
- **fixturesCtrl**: puts into the scope a list of matches of the selected team with details.

- **favoritesCtrl**: gets the list of favorite teams from the local storage. Has functions for favorite and unfavorite a team. Also has a function to openingDialog with specific details of a team clicked.

Views

This directory contains html templates for render the information presented to the user.