datavisyn

Coding Challenge Junior Developer

Goals

The goal of this coding challenge is to get an impression whether you are able to handle the basics of the datavisyn technology stack.

The challenge requires you to set up a very basic web application, using npm and the create-react-app tool by facebook.

Prerequisites

If you don't have any experience in Typescript or React the TypeScript React Starter tutorial by Microsoft is a good starting point.

https://github.com/Microsoft/TypeScript-React-Starter

Follow this tutorial in order to get a feeling of how the web technology stack works. The section about Redux is not that important in the first place.

We use Visual Studio Code as IDE. You can download it at https://code.visualstudio.com/.

You can find the Typescript documentation here
https://www.typescriptlang.org/docs/home.html
https://www.typescriptlang.org/docs/handbook/basic-types.html

If you don't have a github account, please register one and share your work with us.

We recommend to develop under Ubuntu. For the installation of the environment under Ubuntu, the following commands will be helpful:

```
$ sudo apt install git                  // Installs git
$ sudo apt install npm                  // Installs npm
$ sudo apt update
$ sudo apt install nodejs               // Installs nodejs
$ sudo npm install -g create-react-app  // Installs the create tool
```

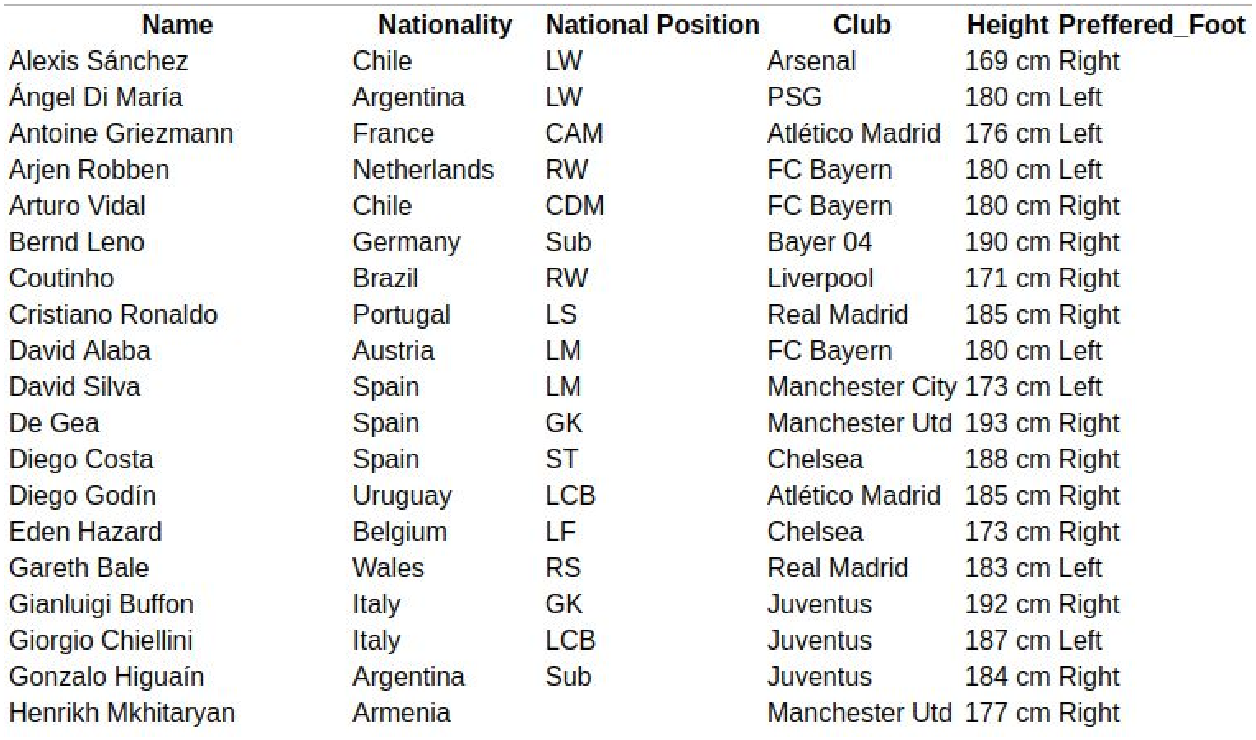
Challenge

Player Table

Your are given a file that contains a number of soccer players as an array of JSON objects. Your task is it to create a web app that lists up these soccer players in a plain HTML table. The table must be sorted descendingly by the player’s name and should contain the following columns:

- Name
- Nationality
- National Position
- Club
- Height
- Preferred Foot

The plain HTML table will look like this:

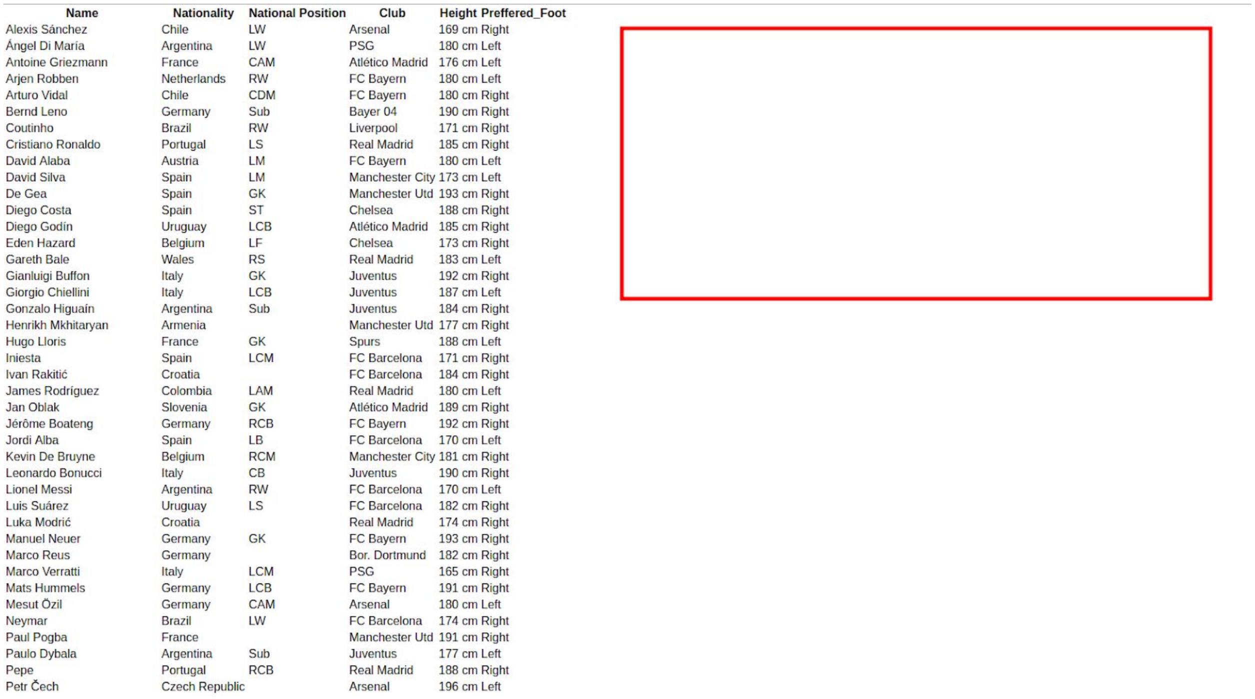


Name	Nationality	National Position	Club	Height	Preffered_Foot
Alexis Sánchez	Chile	LW	Arsenal	169 cm	Right
Ángel Di María	Argentina	LW	PSG	180 cm	Left
Antoine Griezmann	France	CAM	Atlético Madrid	176 cm	Left
Arjen Robben	Netherlands	RW	FC Bayern	180 cm	Left
Arturo Vidal	Chile	CDM	FC Bayern	180 cm	Right
Bernd Leno	Germany	Sub	Bayer 04	190 cm	Right
Coutinho	Brazil	RW	Liverpool	171 cm	Right
Cristiano Ronaldo	Portugal	LS	Real Madrid	185 cm	Right
David Alaba	Austria	LM	FC Bayern	180 cm	Left
David Silva	Spain	LM	Manchester City	173 cm	Left
De Gea	Spain	GK	Manchester Utd	193 cm	Right
Diego Costa	Spain	ST	Chelsea	188 cm	Right
Diego Godín	Uruguay	LCB	Atlético Madrid	185 cm	Right
Eden Hazard	Belgium	LF	Chelsea	173 cm	Right
Gareth Bale	Wales	RS	Real Madrid	183 cm	Left
Gianluigi Buffon	Italy	GK	Juventus	192 cm	Right
Giorgio Chiellini	Italy	LCB	Juventus	187 cm	Left
Gonzalo Higuaín	Argentina	Sub	Juventus	184 cm	Right
Henrikh Mkhitaryan	Armenia		Manchester Utd	177 cm	Right

You can load the JSON data directly into your React Component with import declaration. There is no need to load the data from an external source or location. Once you load the data, try to use Javascript Array methods like .sort or .map to transfer the array into table rows. See https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array for further information. Please pretty up the table using CSS or (SCSS).

Player Detail View

In the area that is indicated with the red square, detailed information for each selected player should be shown. The selection can be made either by clicking on a table row. You can also add a button or link to each row which triggers the selection.



Name	Nationality	National Position	Club	Height	Preffered_Foot
Alexis Sánchez	Chile	LW	Arsenal	169 cm	Right
Ángel Di María	Argentina	LW	PSG	180 cm	Left
Antoine Griezmann	France	CAM	Atlético Madrid	176 cm	Left
Arjen Robben	Netherlands	RW	FC Bayern	180 cm	Left
Arturo Vidal	Chile	CDM	FC Bayern	180 cm	Right
Bernd Leno	Germany	Sub	Bayer 04	190 cm	Right
Coutinho	Brazil	RW	Liverpool	171 cm	Right
Cristiano Ronaldo	Portugal	LS	Real Madrid	185 cm	Right
David Alaba	Austria	LM	FC Bayern	180 cm	Left
David Silva	Spain	LM	Manchester City	173 cm	Left
De Gea	Spain	GK	Manchester Utd	193 cm	Right
Diego Costa	Spain	ST	Chelsea	188 cm	Right
Diego Godín	Uruguay	LCB	Atlético Madrid	185 cm	Right
Eden Hazard	Belgium	LF	Chelsea	173 cm	Right
Gareth Bale	Wales	RS	Real Madrid	183 cm	Left
Gianluigi Buffon	Italy	GK	Juventus	192 cm	Right
Giorgio Chiellini	Italy	LCB	Juventus	187 cm	Left
Gonzalo Higuaín	Argentina	Sub	Juventus	184 cm	Right
Henrikh Mkhitaryan	Armenia		Manchester Utd	177 cm	Right
Hugo Lloris	France	GK	Spurs	188 cm	Left
Iniesta	Spain	LCM	FC Barcelona	171 cm	Right
Ivan Rakitić	Croatia		FC Barcelona	184 cm	Right
James Rodríguez	Colombia	LAM	Real Madrid	180 cm	Left
Jan Oblak	Slovenia	GK	Atlético Madrid	189 cm	Right
Jérôme Boateng	Germany	RCB	FC Bayern	192 cm	Right
Jordi Alba	Spain	LB	FC Barcelona	170 cm	Left
Kevin De Bruyne	Belgium	RCM	Manchester City	181 cm	Right
Leonardo Bonucci	Italy	CB	Juventus	190 cm	Right
Lionel Messi	Argentina	RW	FC Barcelona	170 cm	Left
Luis Suárez	Uruguay	LS	FC Barcelona	182 cm	Right
Luka Modrić	Croatia		Real Madrid	174 cm	Right
Manuel Neuer	Germany	GK	FC Bayern	193 cm	Right
Marco Reus	Germany		Bor. Dortmund	182 cm	Right
Marco Verratti	Italy	LCM	PSG	165 cm	Right
Mats Hummels	Germany	LCB	FC Bayern	191 cm	Right
Mesut Özil	Germany	CAM	Arsenal	180 cm	Left
Neymar	Brazil	LW	FC Barcelona	174 cm	Right
Paul Pogba	France		Manchester Utd	191 cm	Right
Paulo Dybala	Argentina	Sub	Juventus	177 cm	Left
Pepe	Portugal	RCB	Real Madrid	188 cm	Right
Petr Čech	Czech Republic		Arsenal	196 cm	Left

The detail information must contain the values of the following parameters:

- Name of the player as header
- Visualization
  - Ball_Control
  - Dribbling
  - Aggression
  - Acceleration
  - Speed
  - Shot_Power

You can display these values as a table or use a chart to visualize this data. When you use a chart we recommend the following react wrapper for chart.js:
http://jerairrest.github.io/react-chartjs-2/
https://github.com/jerairrest/react-chartjs-2

You can also use vega-lite or plotly-js which also has a nice react wrapper.

Coding Hints

The create-react-app tool creates a whole project for you with a default component called App. You can keep this default component and implement both elements (player table and player detail view) as sub components with the same file App.tsx.

You can import the data directly into the file App.tsx.

The players don't have any artificial IDs, but their names are unique. So for this example it is perfectly fine to use the player's name as an ID if you need it.

Take a look on how state and props work on React components. Here is a good tutorial: https://reactjs.org/docs/state-and-lifecycle.html In this example, a simple clock is implemented. The current time is stored in the state. Everytime the state is updated, the clock re-renders itself. In this way, you can trigger the rendering of the player detail view.