

# Front End Developer Test

This test is for us to see how you approach a problem and your way of solving it.

This test needs to be done using React with Mobx or Redux with NodeJS API.

The project needs to contain in the readme file with all the instructions to download and run the project.

Pluses:

- . Git version control with feature branches (GitHub, Bitbucket, GitLab).
- . Deploy the front-end and back-end using Heroku, Google Cloud or any cloud service.
- . Tests using Jest and Enzyme.

Feel free to add anything “extra” that you deem as relevant for us to understand how you solve each question.(Algorithms, links, etc.).

Please do not rush through it, take your time. This is not about how fast you do it, it’s about how you do it.

## Question 1:

Write a function that connects to <https://restcountries.eu/> and gets a unique country from a specific name given using the Node back end and send it to the front end.

## Question 2:

Using the same API ( <https://restcountries.eu/> ), and from an array of string, write a function that returns a list of countries where their name matches at least a part of one of these string use the Node back end and send it to the front end.

## Question 3:

Using the same API ( <https://restcountries.eu/> ) in the React front end list all the countries and a field to filter the country by name.

## Question 4:

Considering a Slot machine defined like this:

- Reel1: ["cherry", "lemon", "apple", "lemon", "banana", "banana", "lemon", "lemon"]
- Reel2: ["lemon", "apple", "lemon", "lemon", "cherry", "apple", "banana", "lemon"]

- Reel3: ["lemon", "apple", "lemon", "apple", "cherry", "lemon", "banana", "lemon"]

Coins:

The user start with 20 coins, when the user run the function the user will lose 1 coin,

*1 spin = 1 coin*

- . 3 cherries in a row: won 50 coins
- . 2 cherries in a row: won 40 coins
- . 3 Apples in a row: won 20 coins
- . 2 Apples in a row: won 10 coins
- . 3 Bananas in a row: won 15 coins
- . 2 Bananas in a row: won 5 coins
- . 3 lemons in a row: won 3 coins

Using these data, create a function that, when it's called by the front end, gives back the result of a spin and show the result.

### **Question 5:**

Create form with validation all fields (Name, Email, Password) should be required, send to the back end. The endpoint need to save the user somewhere, you can use in memory or any database (SQL or NoSQL).

### **Question 6:**

Add Authentication using JWT create form to login and save the user info on local storage.