



Frontend developer test

Author: Jordi Boix Ramos

Date: 03/06/2022

Contenidos

1	Introduction.....	3
2	Test description.....	3
2.1	Functionality	3
2.2	Points to evaluate.....	3
2.2.1	Design	3
2.2.2	Implementation.....	3
2.2.3	Connection.....	4
3	Enlargements	4
4	Delivery.....	4

1 Introduction

Project's technical team offers this technical test with the aim of broadening the knowledge of the interviewee's abilities, as well as his or her way of working and approach to problem solving.

It should be noted that this test does not have a single answer, as the interviewee is expected to personally qualify the approach and resolution to the problem described in this document.

2 Test description

The test consists of designing, implementing and connecting a platform with "cards" for the display of posts or comments made by a user. To do this, we will use as an example the free content offered by <http://jsonplaceholder.typicode.com>, and, more specifically, the results provided by the GET call of the <https://jsonplaceholder.typicode.com/posts>.

2.1 Functionality

The final functionality of the solution must allow the user to manage and visualise the posts of all the users available in the tool.

Conceptually, we speak of visualising a post as the user being able to see the title of the post (identified by the **title** attribute), the user who created the post (identified by the **userId** attribute), and the text content of the post (identified by the **body** attribute). This display should be summarised on a single screen with an aggregation of all available posts.

This visualisation should allow the user to delete posts. This functionality should only be applied at the client code level without the need to launch a request to delete the post in the backend, although the layout will be taken into account.

2.2 Points to evaluate

2.2.1 Design

The design of the interface of the tool described will be positively valued for its analysis of the problem and for anticipating possible scenarios that may occur at the level of functionality.

2.2.2 Implementation

Here everything related to the code level will be assessed. This part is much more complex to specify and therefore we will set certain necessary and optional objectives.

- Objectives required
 - Code entirely in TypeScript.
 - Typing without any types.
 - Code in React
 - Use of functional components
 - Good indentation
 - Do not use external libraries that perform the function described in the document. Libraries that are a means to achieve the final functionality

are allowed (a library that provides a dropdown for the card would be allowed, a library that creates the whole card would not).

- Optional
 - Unit testing
 - e2e testing
 - Scss styles and css variables
 - BEM
 - Flexbox
 - Grid-CSS

2.2.3 Connection

For this part, the posts platform must be directly connected to the jsonplaceholder results, so that the data is requested and dynamically displayed on the platform.

For this step, the way to implement it will be assessed. Any sample implementation of the functionality using the Redux store will be considered positively (even if it is not the best solution in principle).

3 Enlargements

In addition to the functionality described above, a number of extensions are proposed in case the interviewee is interested in the test and wants to extend the functionality of the platform:

- Editing posts
- Basic layout
 - Navbar
 - Sidebar
 - Content
 - Breadcrumbs
- Login screen
- Colour palette and style guide
- CSS styles and animations

4 Delivery

The delivery format of the project must be in .zip format or through the link to the repository where it has been implemented. This repository must be ready to be launched and display the developed platform by installing the necessary dependencies.