Technical Challenge

for SEAT:CODE Frontend Developer

## Welcome!

If you are reading this document, thanks for the effort invested so far. We feel you can be a great member of our family. We want to learn a bit from your technical background to carry on with the process.

## Instructions

### Problem

You will be facing a Kata alike problem. It’s a code challenge small enough so it can be solved without investing too much time, but complex enough to give us some hints on your thinking process, problem-solving, testing, and software engineering skills.

### Timing

The code challenge should take you around 4 hours. Feel free to invest more if you want to, but we understand that your time is a precious asset. We ask you to deliver your solution maximum 48 hours after you have received the challenge. So we can speed up the process and improve your candidate experience.

### How to delivery the Code Challenge

Our preferred method of delivery is using GitHub. GitHub supports personal private projects, so it won’t cost you any pence. We have created a Fake Kent Beck GitHub account ([@kent-beck](https://github.com/kent-beck)) who you will have to invite as a collaborator. If you don’t know how to do it, take a look at [“Inviting collaborators to a personal repository”](https://help.github.com/en/github/setting-up-and-managing-your-github-user-account/inviting-collaborators-to-a-personal-repository) Github Documentation Page. If for some reason you cannot use GitHub, you can zip your code challenge and send it to [candidates@code.seat](mailto:candidates@code.seat). Kent Beck won’t be as happy, but it will do the trick. Send us also an email in case of any issue with the delivery.

### Last words

If you have any tech questions related to the challenge, take an assumption, and carry on. Please provide a small README file on how to compile and run your application. Best of luck! We hope you enjoy the code challenge!

## Code Challenge

***What we expect from you***

* **Required:** Use some public API or create your own mocked API with [JSONServer](https://github.com/typicode/json-server)
* **Required:** Contains a React or Angular 2+ client that consumes that API and can list, show, create, update, and remove the retrieved data on the APP. Apply RxJs good practices and asynchronous views on templates
* **Required:** Use some State Manager as Redux, React Hooks, [NgRx](https://ngrx.io/) or [Akita State Management](https://github.com/datorama/akita).
* **Required:** TDD + Documentation

Write client side in TypeScript.

Use a CSS framework if it helps in the UI or develop your own CSS architecture

**Principal Idea:**

*(you can surprise us with another one If you feel more comfortable)*

* Create a DataTable to sort a list of items with some information, it could have good points as:
  + A search bar that allows search by any field.
  + Sort by field.
  + Add, modify or delete items from the table with an intuitive UX/UI template.
  + Choose one item from the table and display it in a showcase format.

**Optional points:**

* Creativity
* Responsive
* CLEAN code
* SOLID principles
* Good performance