

TrackTik Frontend Challenge 🙌

Congrats!

You are ready to complete the TrackTik Frontend Challenge

🌟. ☐ (' ⌚) 🌟. 🌟.

Your goal is to create a web application with two screens. Your UI should be compatible with mobile; desktop compatibility is a bonus.

- A site list screen [VIEW](#)
- A site details screen [VIEW](#)

To complete this challenge, you need to communicate with our REST API.

Our REST API is dumb but flexible if you want to do some extra features. The toolbar does not need to be set-up, but you are of course welcome to implement it if you want to show extra skills.

At TrackTik we started to use Vue.js but you can use other framework and any kind of library.

This test project is a start and you have to imagine that many complex features will be implemented in the future. You need to use development best practices so that developers who will work on this project in the future can maintain it easily and add more features without doing a lot of refactoring.

Please provide a quick README.md which contains all the instructions to run your application

API 🙌

Our REST API is built around [json-server](#). Want a quick tour? Follow this [link](#).

When you don't use this API for a while, dataset could be regenerated. Keep it in mind if you added resources or if you play with localStorage 😬.

You are allowed to do mutation on resource via (POST/PUT/PATCH/DELETE) but, if things go wrong, you can rebuild a valid dataset via GET /build-db

All resources enable:

- [sites](#)
- [clients](#)
- [version](#)

- [me](#)
- [db](#) ⚠️

⚠️ Please don't use /db endpoint on our challenge, this endpoint is just a helper to overview our API dataset ⚠️

Use on our local machine

Don't provide /build-db endpoint
npm install -g json-server

```
curl -o db-initial.json http://{enter-api-url}/db  
cp db-initial.json db.json  
json-server --watch db.json
```

Thank you 🙏

- [json-server](#)
- [Showdown](#)