**A better React folder structure**

Let’s start with a quick summary about the key points of this article:

* Why React doesn’t have a specific folder structure?
* Why a folder structure is important?
* Let me suggest a better folder structure

Graphical user interface, text, letter

Description automatically generated

**Why React doesn’t have a specific folder structure?**

I’m not planning to answer that question, I actually wonder why there is no specific folder structure.

Reading the [Official React Documentation about file structure](https://reactjs.org/docs/faq-structure.html) we find out that there is no recommended way to structure React projects.

On the one hand, we may think that it’s great, and actually it is; we have the flexibility to decide how we want to structure the files in our project. We use a folder structure that we like, something that we are comfortable with and something which is suitable for that specific project.

On the other hand, there are some reasons why, for me, it’s very important to think carefully about the folder structure.

**Why a folder structure is important?**

Investing some time to identify the best folder structure which works for you would bring some benefits:

* Maintainability: if the same folder structure is used on your projects. Tt’s easier to come back to an old project when you know by heart where is everything and how the different pieces are working together.
* Scalability: on large projects, the codebase could become a real mess easily if a folder structure is not clearly defined.
* Collocation: keeping the code closer to where is use would boost your productivity.
* The good all web: React allows you to mix the different layers of a web application. HTML, CSS, JavaScript and data can be easily mixed, which could reduce the reusability of your code.

**A better React folder structure**

This folder structure is better because, as we saw above, there is no recommended folder structure. So something is better than nothing :)

The following folder structure is based on Next.js. If you are not using it, pages could be moved to the *UI* folder and *Layout Dynamic*should be place somewhere connected to your Router.

A picture containing diagram

Description automatically generated

A UI folder without data fetching will make the process of documenting the UI with Storybook and writing tests with Jest a faster. Mainly because the number of APIs to mock should be none or, at least, very reduced.

In addition, splitting the data layer and the UI layer would make your components easier to reuse in the same project or in some other project.

**File structure in the UI components**

Another important topic is how to structure the files inside a component. Let’s say we have an accordion in our “elements” folder

Diagram

Description automatically generated

Why splitting the Accordion into logic (useAccordion), markup (Accordion) and styling (Accordition.styled)? The can easily go all together in the same file, React allows us to do it. However, let’s see when this pattern could be useful:

A picture containing diagram

Description automatically generated

The image above describe the 4 layers and how they can be shared between web and mobile (with React Native).

* Data Layer (API): it should be 100% independent as it can be fully reused between web and mobile projects.
* Logic Layer (JavaScript): in this case it can be 100% reused as web, so why not a top level “logic” folder? Because collocation is very important, and keeping the the logic close to the markup feels more important for me than having its own folder (which would be a good solution if you are developing in parallel a web and a mobile app).
* Styling Layer (CSS): some styling, if they are CSS-in-JS, could be shared between web and mobile applications. Again, collocation is a relevant concern for the same reasons as described above for the logic.
* Markup Layer (HTML): unfortunately, the markup cannot be reused between web and mobile. React Native could be use in web, but the semantics of the page will be lost and the accessibility suffers as well. This is the main reason why, keeping an independent markup, it’s important toc split the layers if you are planning to develop a web and a mobile app.

**Conclusions**

Use this folder structure or try your own one. But definitely, sit down with your team, raise this topic and decide together how you want to structure your project/s.