

Container and Component Folder Structure

As a part of Redux convention, we will move our files into two main directories. The Container folder will contain all the components which interact with Redux directly. The other components will be moved to the Component folder.

There's a bit of refactoring you need to do before we move on to coding the *Skypey* application.

In Redux applications, it is a common pattern to split your components into two different directories. Every component that talks directly to Redux, whether that is to retrieve state from the store, or to dispatch an action, these components should be moved to a containers directory.

Other components, those that do NOT talk to Redux, should be moved over to a **components** directory.

Well, well, well. Why go through the hassle?

For one, your codebase becomes a little cleaner. It also becomes easier to find certain components as long as you know if they talk to Redux or not.

So, go ahead.

Have a look at the components in the current state of the application, and reshuffle accordingly.

So you don't screw things up, remember to move the components' associated CSS file.

Here's my solution:

1. Create two folders. **Containers** and **components**
2. **App.js** attempts to retrieve contacts from the store. So, move App.js and App.css to the containers folder.
3. Move **Sidebar.js**, **Sidebar.css** , **Main.js** and **Main.css** to the components

folder. They do not talk to Redux directly for anything.

4. **Please do not move Index.js and Index.css.** Those are the entry point of the App. Just leave those at the root of the project directory.
5. Please move **User.js** and **User.css** to the containers directory. **The User component does NOT talk to Redux yet but it will.** Remember that when the App is completed, upon clicking a user from the sidebar, their messages will be shown. By implication, an action will be dispatched. In the coming sections, we'll build this out.
6. By now, a lot of your import urls will be broken i.e., the components that imported these moved components. You have to change their import url. I'll leave this up to you. It's an easy fix :)

Here's an example solution for #6 above: In App.js, change the Sidebar and Main imports to this:

```
import Sidebar from "../components/Sidebar";  
import Main from "../components/Main";
```



As opposed to the former:

```
import Sidebar from "./Sidebar";  
import Main from "./Main";
```



Got that?

Here are some tips to solve the challenge yourself:

1. Check the Sidebar.js import statement for the User component.
2. Check Index.js import statement for the App component.
3. Check App.js import statement for the store

Once that is done, you'll have Skypey working as expected!