## Project tech description

For project I used an application template, generated with **create**-**react**-**app**, it actually did a lot of work for me. For creating this project I used several node.js modules, including react, react-router and redux.

### Component structure

Entry point into my application is AppRoot class, defined in App.js, it rendered into <div id="root"></div> in my index.html page in App.js I also create a redux store and initialize react router.

The next part of the application hierarchy is Routing class which defined in ‘src/components/routing/routing’. It’s a container and it mainly defines routs of our application, 404 error page, custom pages like login and home, and pages with default layout like price and 404. This container is attached to the store for a reason, if we have a visitor, who is not logged in, we hide a route to admin page with login\_routes() function (more information about hiding or showing parts of JSX in next paragraph). <Switch> component allow react-router to stop handling routes if app find matching one, that is how 404 page working, if app did not manage to find a corresponding route, then it will show 404 page.

On top of all routing there is an App\_layout class which simply defines, that application contains menu at every single page and rest of HTML, CSS and logic is defined in {children}. Menu is coded in Navigation class which defined in 'src/components/menu/navigation', its also a container, because we need to manage our menu state according lo login state. In my application login state is defined as state.login\_reducer.user\_id if user id is -1 than we don’t have any user logged in. So, we manage our menu by calling login\_menu\_\* methods to append JSX. Some permanent menu items, like about are hardcoded in JSX.

App\_layout {children} are basically our pages themselves. Pages with the same layout (content and footer), like “price” and “about” are nested from Defaultpage component, which is located at 'src/components/defaultpage/defaultpage'. There are several pages with custom layout (for example without footer) like “home” and “login”, all pages defined as a react components and located in separate folders, like “about” page is in ‘/components/about’, “home” page is in ‘‘/components/home’ and so on.

### SCSS

Project is based on SCSS as main CSS preprocessor syntax, at this point I don’t use webpack. I’m using an extension for VSCODE to compile my SCSS files, called “Live Sass Compiler”

### Json data

All json data is located in 'src/jsondata' this folder is containing sample json data, in next steps of project development it will be transfered into backend. At this moment there are 3 json files:

* about\_content.json (information for about page)
* prices.json (information for price page)
* users.json (logins and passwords)

To convert json into JSX I’m using parser components, one of parsers is common component, located in 'src/components/common/page\_json\_parser.js'

, it converts json with paragraphs, pictures, header and footer into article. Second type of parsers is page-related parsers, for now I’m using one to parse price list json into JSX table, this one is located in 'src/components/price/json\_parsers\_price.js'

### Auth

Based on application state, made with redux and “Login” container. After enter creds into form we compare creds to match to users.json file. If there is a match we will replace our state with new state containing state.login\_reducer.user\_id = [id gained from json] after it, containers which are subscribed on user\_id will be re-rendered according new application state. (route /admin will be available, menu will include link to admin and link to logout, link to /login will be removed from menu and /login will get redirection to /home.) Anything is related to login process located at ‘src/components/login/’

### Redux

Application contains storage and several redux containers, used for controlling application state for user auth. At this point application only save its state while its opened. Redux is present in full state, I have one reducer, combined reducers, storage and actions. Storage is declared in 'src/App.js reducers are combined at 'src/components/reducers/index.js'. As for now I have only one reducer and one file with actions for it. They are located at ‘src/components/login/redux\_login\_reducer.js’ and ‘src/components/login/redux\_login\_actions.js’