Passing the Sidebar data via Props

In order to render our contacts in the Sidebar, we need to send these contacts as props and then turn them into an array for iteration, using lodash.

If you take a look at the entire code now, you'll agree that the entry point of the app remains `index.js`

`Index.js` then renders the App component. The App component is then responsible for rendering the Main and Sidebar components.

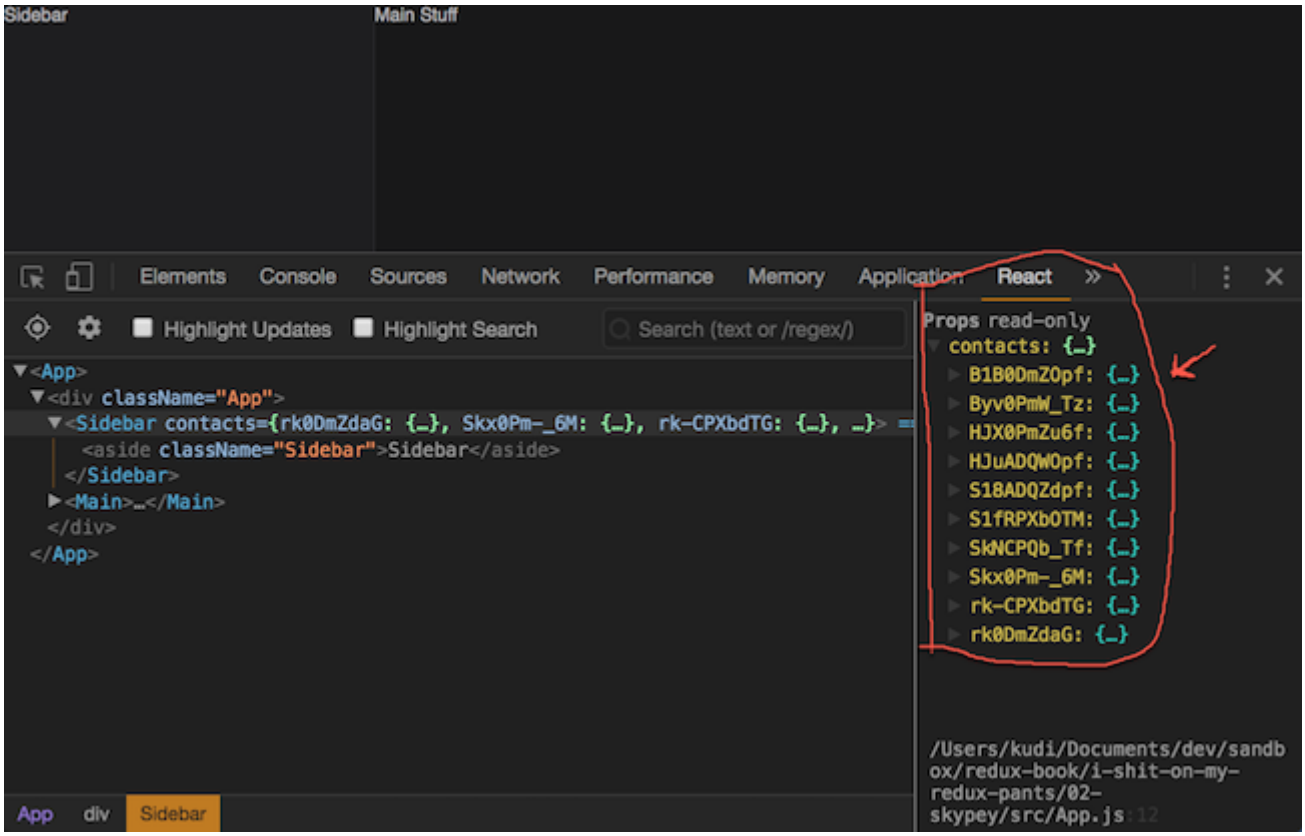
For Sidebar to have access to the required contacts data, we'll pass in the data via props.

In App.js, retrieve contacts from the store, and pass it on to Sidebar like this:

App.js:

```
const App = () => {
  const { contacts } = store.getState();
  return (
    <div className="App">
      <Sidebar contacts={contacts} />
      <Main />
    </div>
  );
};
```

App.js



As I have done in the screenshot above, inspect the Sidebar component and you'll find the contacts passed as a prop, with contacts being an object with IDs mapped to user objects.

Now we can proceed to rendering the contacts. First, install Lodash from the cli:

```
yarn add lodash
```

Import lodash in App.js

```
import  _ from lodash
```

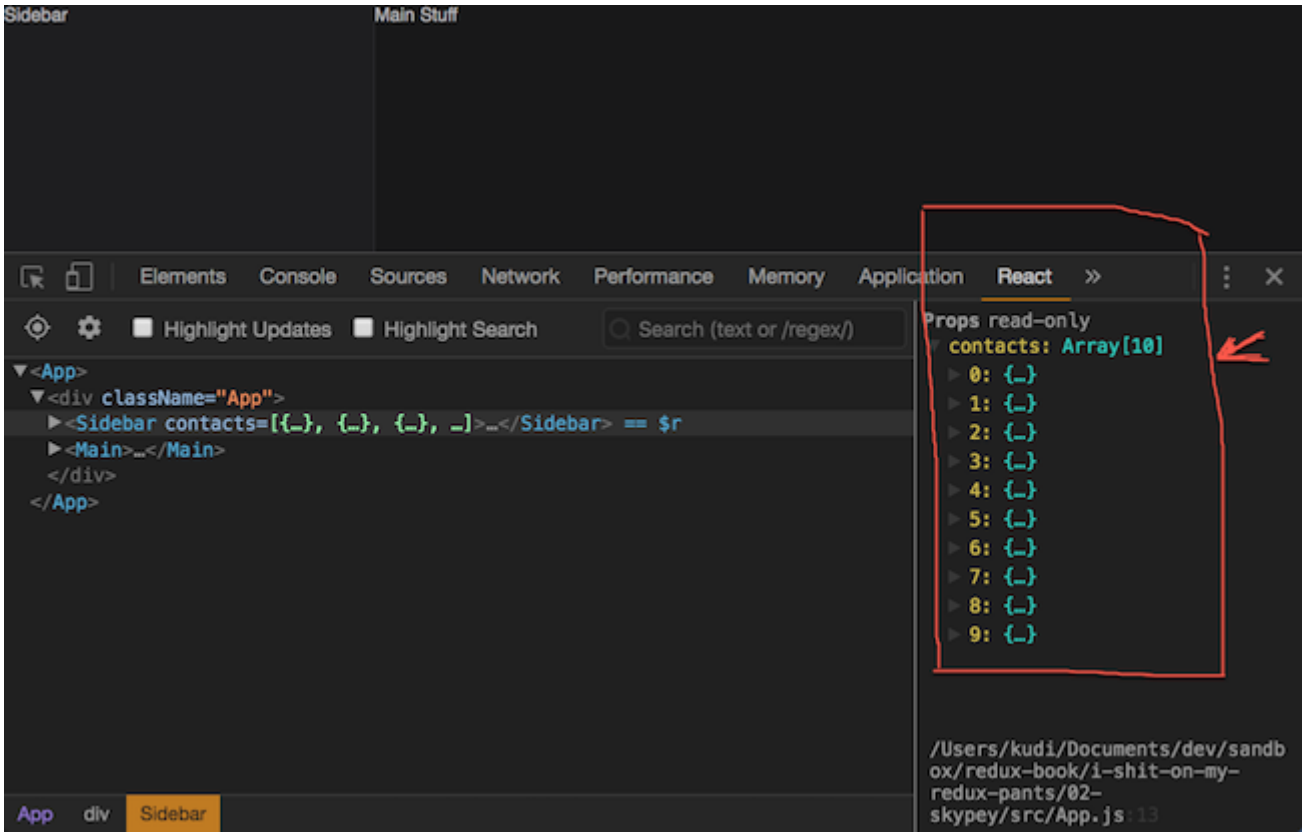
I know. The underscore looks funny, but it's a nice convention. You'll get to love it :)

Now, to use any of the utility methods lodash avails, call the methods on the imported underscore. e.g. .fakeMethod()

Now, put Lodash to good use. Using one of the Lodash utility functions, the contacts object can be easily converted to an array when passed in as props.

Here's how:

```
<Sidebar contacts={_.values(contacts)} />
```



You can read more about the Lodash .values method if you want. In a nutshell, creates an array out of all key values of the object passed in.

Now, let's really render something in the Sidebar.

Sidebar.js:

```
import React from "react";
import User from "./User";
import "./Sidebar.css";
const Sidebar = ({ contacts }) => {
  return (
    <aside className="Sidebar">
      {contacts.map(contact => <User user={contact} key={contact.user_id} />)}
    </aside>
  );
};
export default Sidebar;
```

sidebar.js

In the code block above, we map over the contacts prop and render a User component for each contact.

To prevent the React warning key, the contact's `user_id` **is used as a key**.

Also, each contact is passed in as a user prop to the User component, but this component has not yet been created.

In the next section, we'll develop one of the most fundamental parts of our app, the User object.