

🚩 Current Skill The Store

Create the store

The store is the one responsible for orchestrating the cogs. The store in Redux is kind of magic and holds all the application's state.

Let's create a store to start playing with Redux. Under the store folder, create a new file `index.js`, path to the file ***src/js/store/index.js***, and initialize the store.

```
// src/js/store/index.js

import { createStore } from "redux";
import rootReducer from "../reducers/rootReducer";

const store = createStore(rootReducer);

export default store;
```



As you can see, store is the result of calling `createStore`, a function from the Redux library. `createStore` takes a reducer as the first argument and in our case we passed in `rootReducer` (not yet present).

The most important concept to understand here is that the state in Redux comes from reducers.

createStore function

```
createStore(reducer, [preloadedState], [enhancer])
```



Creates a Redux store that holds the complete state tree of your app. There should only be a single store in your app.

Arguments:



1. **Reducer:** (Function): A reducing function that returns the next state tree, given the current state tree and an action to handle.

2. **[preloadedState]**: The initial state. You may optionally specify it to hydrate the state from the server in universal apps.

3. **[enhancer]**: The store enhancer. You may optionally specify it to enhance the store with third-party capabilities such as middleware, time travel, persistence, etc.

the arguments putted between [] mean that this arguments are optional.

The createStore function returns a store (an object that holds the the complete state of your application.

[< Previous](#)

[next >](#)