

READYMADE BASIC REACT APP:

<https://github.com/facebook/create-react-app>

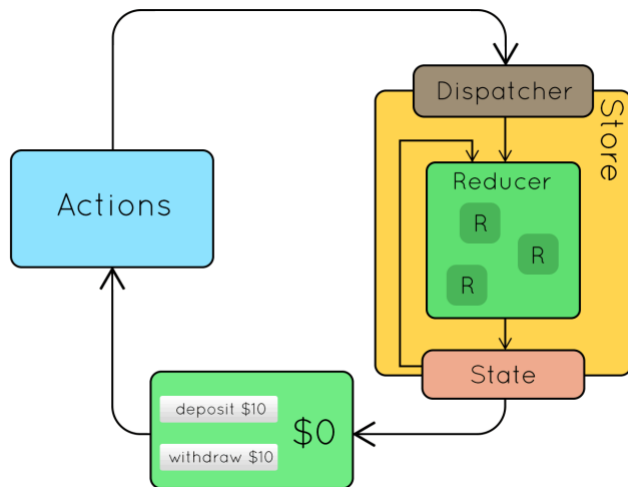
READYMADE BOILER PLATE WITH THE BASIC REACT APP with REDUX:

<https://github.com/codewithbernard/react-redux-boilerplate>

START YOUR FIRST REACT AND REDUX APP with API CALL:

Check Image about Redux

(<https://bumbu.github.io/simple-redux/>)



BASIC STEPS TO FOLLOW – TOTAL 8 Steps:

1. install the normal Redux packages + thunk (dispatcher) & axios axiosthrough npm
 - a. To Install React App:
 - `npx create-react-app my-app`
 - b. Redux, Thung & axios :
 - `npm install --save redux react-redux redux-thunk`
 - `npm install --save axios`
 - You might need a chrome plugin called: "ExtensionAllow-Control-Allow-Origin" for out of domain ajax calls.
2. Create Folders in the Src folder aka root folder:
 - a. **actions** Folder (The folder where you call the APIs – in our example)
 - This folder must have a file called '**index.js**' this is the file where you will make the **API call** and check your self-defined data types by calling the '**types.js**' file (this file only returns a name against an alias – check file).
 - b. **reducers** Folder (The folder where you perform actions on APIs – in our example)
 - This folder must have a file called '**index.js**' this is the record/collection of all reducers.
 - c. **components** Folder (For normal react components – not used in our example)

3. Make basic changes into your Root folder's Index.js file (the root folder version)

// BASICS OF NPM: To add npm package run npm install --save <package_name>

// Default – No change

```
import React from "react";
```

// Default – No change

```
import ReactDOM from "react-dom";
```

// Default – No change

```
import App from "./App";
```

// Default – No change

```
import registerServiceWorker from "./registerServiceWorker";
```

// NEW:Call the Provider Object using 'react-redux'

```
import { Provider } from "react-redux";
```

// NEW:Check the redux diagram—we need to create a store & apply middlewear

```
import { createStore, applyMiddleware } from "redux";
```

// NEW:calling directory will automatically look for index.js & this is a file which imports other reducers and merge them.

Mainly in our example it is "DataReducer.JS"

```
import reducers from "./reducers";
```

// NEW:calling reduxThunk [Dispatcher Part]

```
import reduxThunk from "redux-thunk";
```

// NEW:Finally initiating the store and passing reducers plus middlewear as reduxThunk – because it's a dispatcher.

```
const store = createStore(reducers, applyMiddleware(reduxThunk));
```

// Default – No change

```
ReactDOM.render(
```

// NEW:Wrap the lone <App /> tag with <Provider store={store}> passing the store as attribute.

```
<Provider store={store}>
```

```
<App />
```

```
</Provider>,
```

// Default – No change

```
document.getElementById("root")
```

// Default – No change

```
);
```

// Default – No change

```
registerServiceWorker();
```

Actions folder changes:

4. FILE: **index.js** is to be created inside the **actions** folder.

This is the file where you will call your main APIs using **axios**

Here is a sample of the file code with details:

```
// Calling a typical alias/object self-defined data type from types.js inside actions folder.
```

```
import { FETCH_DATA } from "../types";
```

```
// Calling a axios using axios – for calling unhosted external API – here you will require the help from that chrome plug-in for calling external API as it allow you to do that.
```

```
// Common import for both shortcut or longcut code.
```

```
import axios from "axios";
```

```
// Shortcut of the code – If you find this difficult to understand. I have a longcut version under this.
```

```
export const fetchData = () => async dispatch => {  
  const res = await axios.get("https://swapi.co/api/people");  
  dispatch({  
    type: FETCH_DATA,  
    payload: res.data.results  
  });  
};
```

```
// Longcut of the code – If you find shortcut difficult to understand.
```

```
export const fetchData = () => {  
  return async dispatch => {  
    const res = await axios.get("https://swapi.co/api/people");  
    console.log(res);  
  };  
};
```

5. FILE: **types.js** is to be created inside the **actions folder**.
This is the file where for creating aliases for self-defined data types, returning a const variable only.
Here is a sample of the file code with details:

```
// Plain strings for types
export const FETCH_DATA = "FETCH_DATA";
```

Reducers folder changes:

6. FILE: **dataReducer.js** is to be created inside the **reducers folder**.
Here is a sample of the file code with details:

```
// Calling a typical alias/object self-defined data type from types.js inside actions folder.
import { FETCH_DATA } from "../actions/types";

// state is passed as an array although it should be obj along with 'action'.
// we expect returned state to be an Array obj – hence.
export default (state = [], action) => {
  switch (action.type) {
    case FETCH_DATA:
      return action.payload;
    default:
      return state;
  }
};
```

7. FILE: **index.js** is to be created inside the **reducers folder**.
This is the file where you will combine all your reducers, but we have only one in our example
“**dataReducer.js**”
Here is a sample of the file code with details:

```
// Calling reducers, you have just created. So that you don't need to call them individually, but we have only one here
'dataReducer.js'
import dataReducer from "../dataReducer";
```

// For combining all reducers by using redux, you need to import the object first by using redux.

```
import { combineReducers } from "redux";
```

```
export default combineReducers({
```

```
// Assigning the key value and returning it back after combining.
```

```
  data: dataReducer
```

```
});
```

Others:

8. Run application

Write using terminal

npm start

inside the “react-redux-boilerplate” folder