



LESSON 4-Redux-Toolkit

CSSE4103 – FULL STACK DEVELOPMENT

What is Redux Toolkit?

- **Redux Toolkit** (also known as "RTK" for short) is our official recommended approach for writing Redux logic. The `@reduxjs/toolkit` package wraps around the core redux package, and contains API methods and common dependencies that we think are essential for building a Redux app.

Installation

Install the dependencies in your client/src.

npm install @reduxjs/toolkit react-redux

Check package.json

```
client > {} package.json > ...
11   "@hookform/resolvers": "^3.3.3",
12   "@reduxjs/toolkit": "^1.9.7",
13   "@testing-library/jest-dom": "^5.17.0",
```

Create a Redux Store

Create a file named **src/store.js**.

```
import { configureStore } from '@reduxjs/toolkit'
export const store = configureStore({
  reducer: {},
})
```

Provide the Redux Store to React

In `client/src/index.js`.

```
import { store } from './store'; //Import the store from the
store.js
import {Provider} from "react-redux"

const root =
ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <Provider store={store}> { /* put a <Provider> around your
<App>, and pass the store as a prop:*/}
    <React.StrictMode>
      <App />
    </React.StrictMode>
  </Provider>
);
```

Create a Redux State Slice:

Create a new file named **User.js** in **src/Features/**. In that file, import the `createSlice` API from Redux Toolkit. The features folder will contain the reducers of your application.

```
import { createSlice } from "@reduxjs/toolkit";

const initialState = { value: [] }; //list of user is an object with
empty array as initial value

export const userSlice = createSlice({
  name: "users", //name of the state
  initialState,
  reducers: {
    addUser: (state, action) => {
      //state is the current value of the state, action is triggered
      //outside the reducer and provides a value to the reducer as payload
    },
  },
});

export default userSlice.reducer;
```

Rules of Reducers

- They should only calculate the new state value based on the state and action arguments
- They are not allowed to modify the existing state. Instead, they must make immutable updates, by copying the existing state and making changes to the copied values.
- They must not do any asynchronous logic or other "side effects".



Add Slice Reducers to the Store

In **src/store.js**, import the `usersReducer` from the `features/User.js` file.

```
import { configureStore } from "@reduxjs/toolkit";
import usersReducer from "../Features/Users"; //import the reducer

export const store = configureStore({
  reducer: {
    users: usersReducer,
  },
});
```


Use Redux State and Actions in React Components

- Please refer to the example project.



References

1. Rules of Reducers

<https://redux.js.org/tutorials/essentials/part-2-app-structure#rules-of-reducers>

2. Redux Toolkit Quick Start

<https://redux-toolkit.js.org/tutorials/quick-start>

