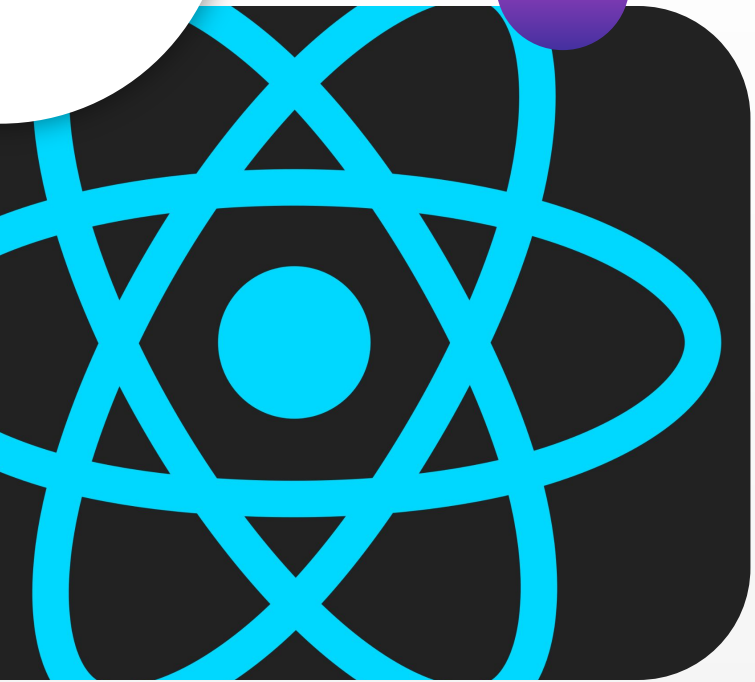


React Js

Eng.Mohamed Gamal

Redux

React State Managment

Create New Project

We will create new project to implement Redux structure

Layout

NavBar

NavBar is a part of layout that controller the top side of site

Footer

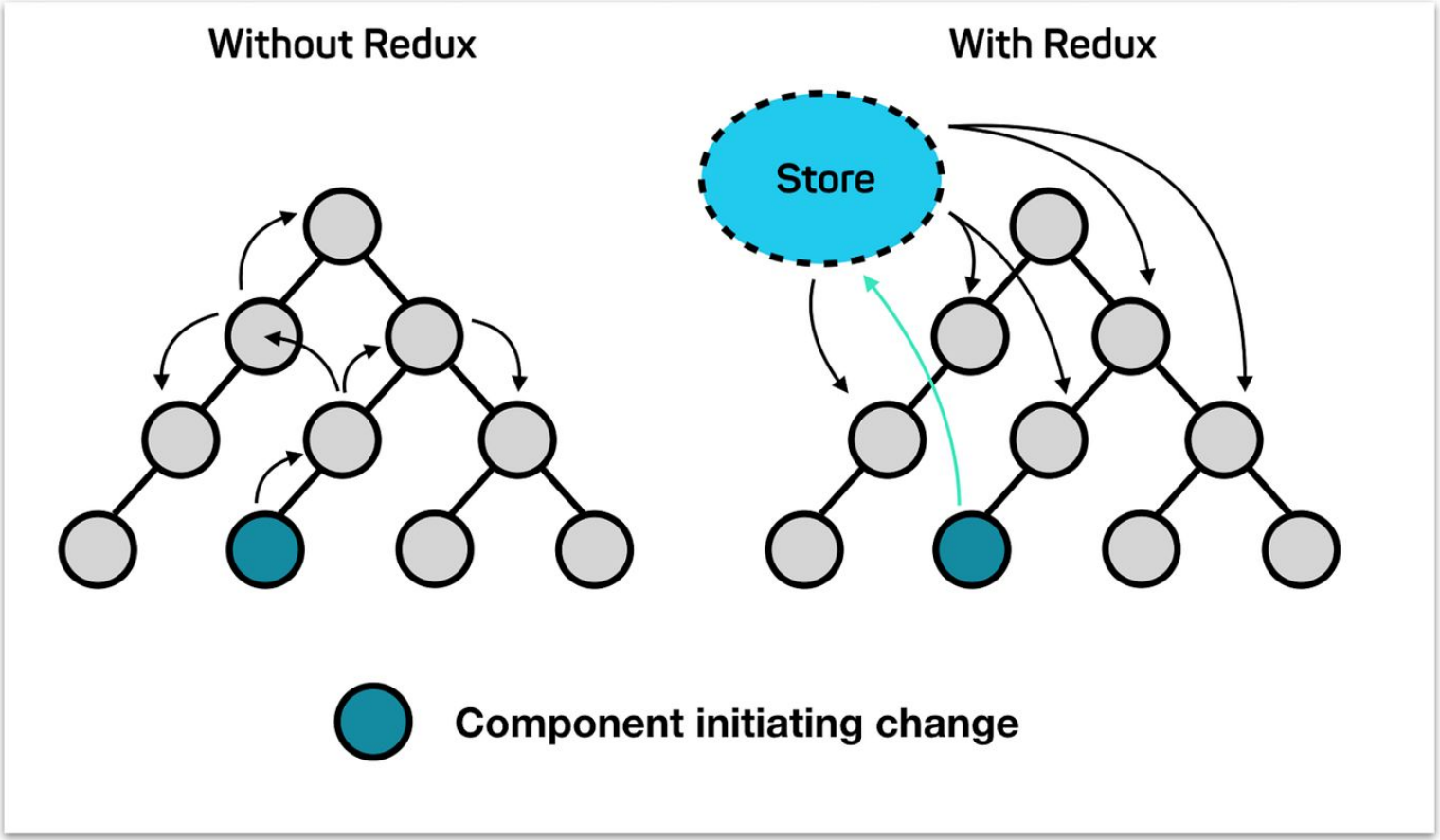
Footer is a part of layout that controller the bottom side of site

Routing

npm install react-router-dom

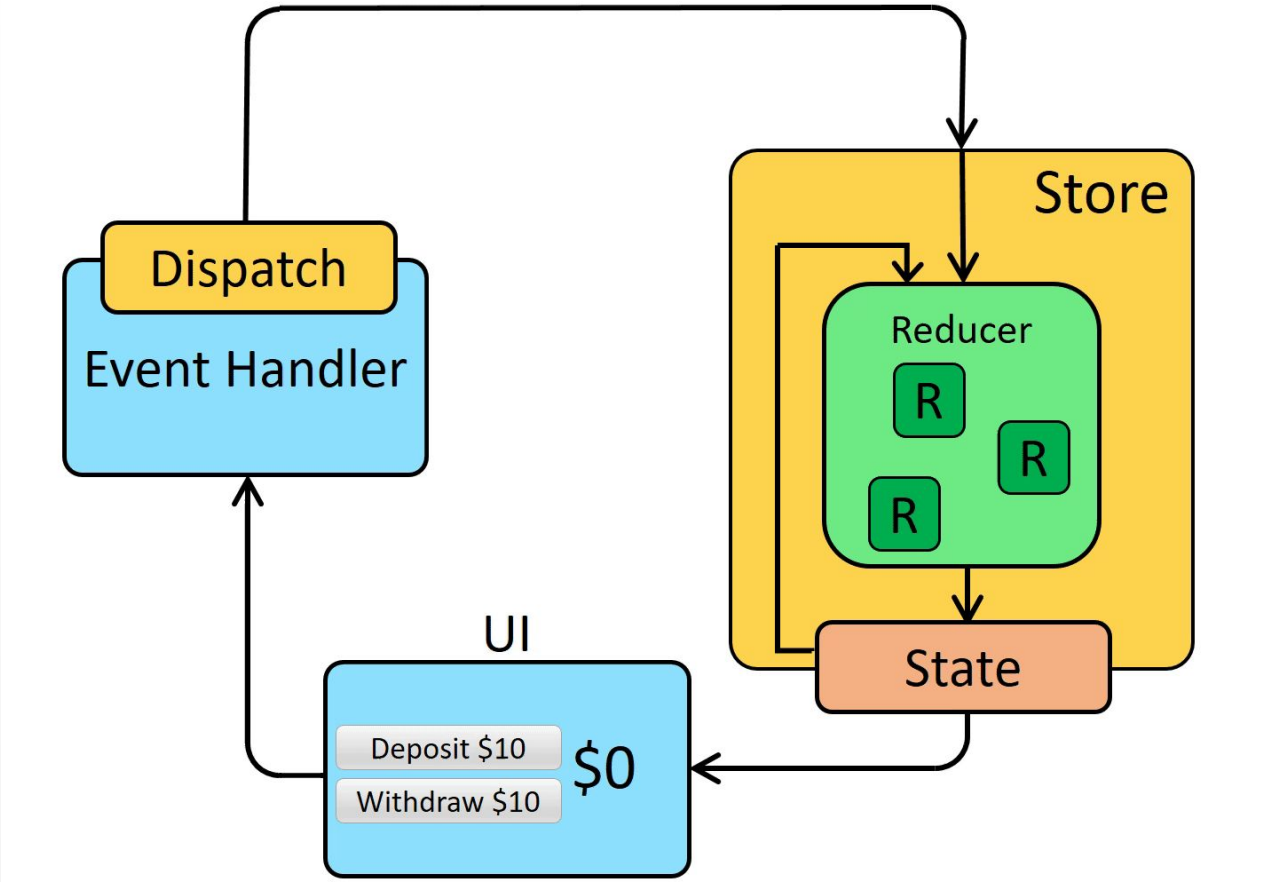
Routing components

1. <BrowserRouter>
2. <Routes>
3. <Route>
4. <NavLink>
5. isActive


Without Redux
With Redux
Store
Component initiating change

Redux Parts

1. Store
2. Action
3. Reducers


Store
Dispatch
Event Handler
Reducer
R
R
R
UI
State
Deposit $10
Withdraw $10
$0

Redux actions

these are objects that should have two properties, one describing the type of action, and one describing what should be changed in the app state.

```
const setUsername = (payload) => {

return { type: "LOGIN", payload: payload }

 }
```

Redux reducers

these are functions that implement the behavior of the actions. They change the state of the app, based on the action description and the state change description.

```
export default (state = {}, action) => {

switch (action.type) {

 case type: return { ...state, ...action.payload, };

default: return state

; }
```

Redux Store

The Redux store brings together the state, actions, and reducers that make up your app. The store has several responsibilities:

Without devtools:

```
const store = createStore(reducers , composeEnhancers );

export default store;
```

With devtools

```
const store = createStore(reducers , composeWithDevTools());

export default store;
```

Wrap your app with redux

We will us provider to wrap app with redux .

<Provider store={store}>

<App />

</Provider>

Redux Hooks

1. useSelector()
2. UseDispatch()

UseSelector()

To read and get different store values

Syntax : const state = useSelector(state => state)

UseDispatch()

To Update store values and dispatch actions

Syntax : const dispatch = useDispatch();

dispatch(action())