**Why Redux?**

1. Pass data between Components

2. Store data from API and use it on components (Redux-Saga)

3. Store Form data (Redux Form) (Formik is better instead of Redux Forms)

Steps

1. Store

- Reducer

- State

2. Reducer (It is an agent)

- State

- Action

3. Subscribe (Getting connected to the base)

4. Dispatch (Action)

Step2:

const reducer = function (state, action)

{

if(action.type === "ATTACK"){

return action.payload;

}

if(action.type === "GREENATTACK"){

return action.payload;

}

return state;

}

Step1:

const store = createStore(reducer, "Peace");

Step3:

store.subscribe(() => {

console.log("Strore is now", store.getState());

})

Step4:

store.dispatch({type: "ATTACK", payload: "Iron Man"})

store.dispatch({type: "GREENATTACK", payload: "Hulk"})

**Quickly React**

- Store - Stores all the data and provides methods to manipulate this data (createStore()).

- Provider Component - (injects data from the store into components.) Make it possible for any other components to take data from the store

- connect() - Wraps any component and let you map certain parts of your application state from the store to the component's properties.

Reducer -

function todoApp (state = initialState, action)

{

switch (action.type)

{

case 'SET\_VISIBILITY\_FILTER' :

return Object.assign({}, state,{

visibilityFilter : action.filter

})

case 'ADD\_TODO':

return Object.assign({}, state,{

todos : ...state.todos,

{

text : action.text,

completed : false

}

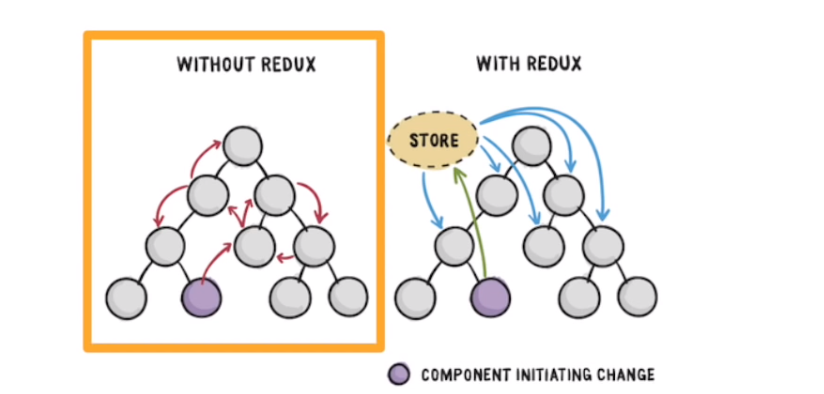
})

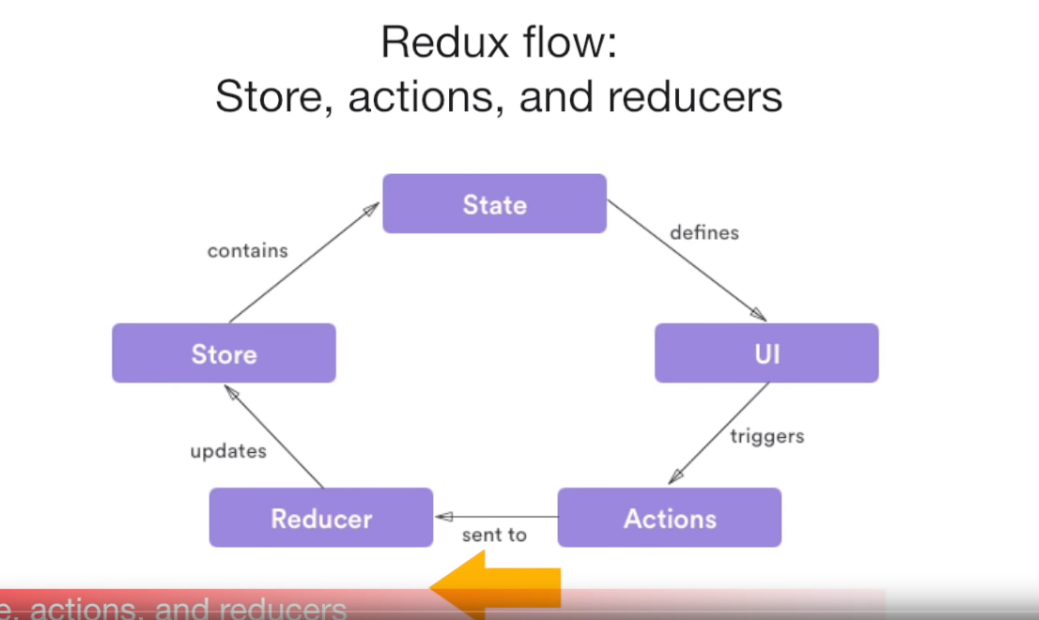
}

}

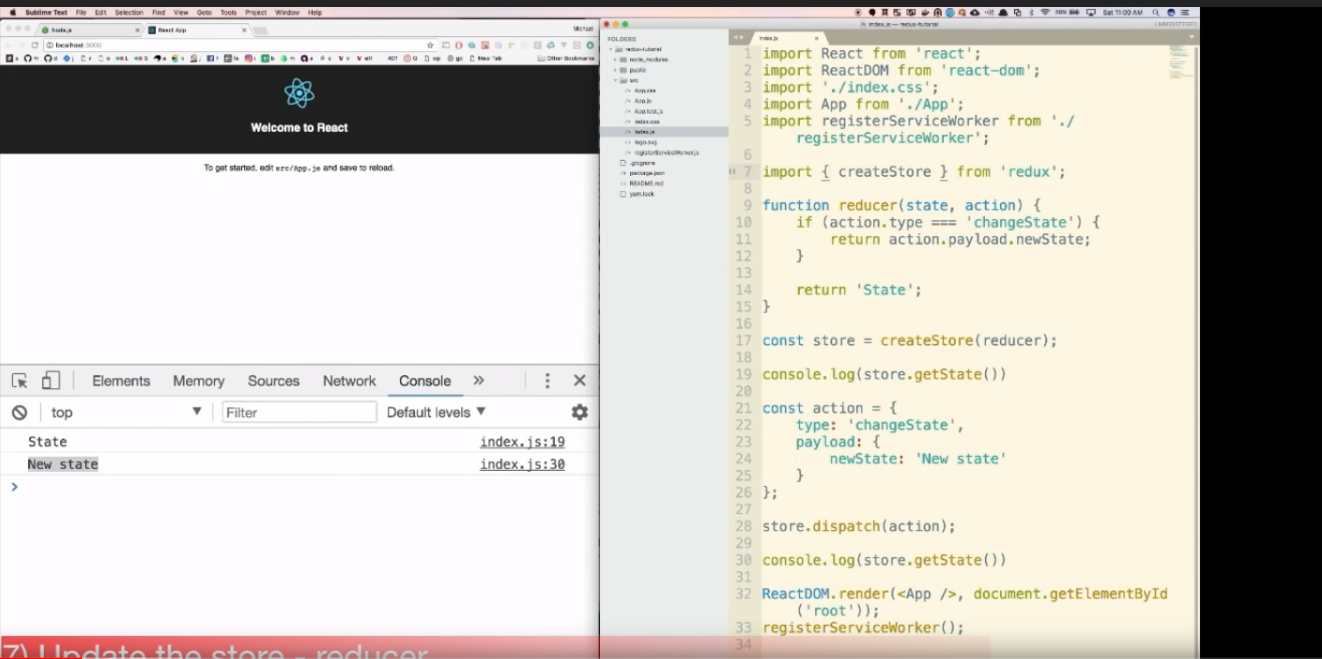
- We may have one or more reducers in our redux application. Every time you call an action, every reducers is called. Reducers are responsible for changing the data in the store.

- We can replace huge switch..case by redux-actions

**How Redux Works?**



**Setup Redux**



* In Reducer, using Switch case is not a good pattern so we can use redux action package to use **‘handleAction’** which kind of map-like plain object, where **keys are action types and values are functions**.

npm install --save redux-actions