



HELLO REDUX

Dr. Jose Annunziato

Directory structure

index.js

App.js

reducers/

containers/

components/

actions/

npm install redux --save

npm install react-redux --save

npm start

First create a simple React application

```
import React from 'react';  
import ReactDOM from 'react-dom';  
import App from './App';  
ReactDOM.render(  
  <App />,  
  document.getElementById('root')  
);
```

Render a HelloWorld container

```
import React, { Component } from 'react';
import HelloWorld from
  './containers/HelloWorld'
class App extends Component {
  render() {
    return (
      <HelloWorld />
    );
  }
}
export default App;
```

And then the container renders Hello component

```
import React, { Component } from 'react';  
import Hello from './../components/Hello'  
class HelloWorld extends Component {  
  render() {  
    return <Hello/>  
  }  
}  
export default HelloWorld
```

Finally, Hello component just renders Hello World!

```
import React from 'react';  
const Hello = () => {  
  return (  
    <div>  
      <h1>Hello World!</h1>  
    </div>  
  )}  
export default Hello
```

Parameterize Hello Component

Dr. Jose Annunziato

Accept a message as a parameter

```
const Hello = ({ message }) => {  
  return (  
    <div>  
      <h1>Hello World!{message}</h1>  
    </div>  
  )  
}
```


And then pass the message as an attribute

```
import React, { Component } from 'react';
import Hello from '../components/Hello'
class HelloWorld extends Component {
  render() {
    return <Hello message="Hello!!!"/>
  }
}
export default HelloWorld
```

Accept event handlers and bind to buttons onClick

```
const Hello = ({ message, onClick, reset }) => {  
  return (  
    <div>  
      <h1>{ message }</h1>  
      <button onClick={onClick}>Click</button>  
      <button onClick={reset}>Reset</button>  
    </div>  
  )  
}
```

Implement Actions

Dr. Jose Annunziato

Implement the actions in `src/actions/index.js`

```
export const HELLO_WORLD = 'HELLO_WORLD'  
export const RESET = 'RESET'
```

```
export const helloWorld = () => {  
  return { type: HELLO_WORLD }  
}
```

```
export const reset = () => {  
  return { type: RESET }  
}
```

Map State and Event Dispatchers

Dr. Jose Annunziato

Refactor HelloWorld to connect state and dispatchers

```
import { connect }  
  from 'react-redux'  
import { helloWorld, reset }  
  from './../actions'  
import Hello  
  from './../components/Hello'
```

Refactor HelloWorld to connect state and dispatchers

```
const mapStateToProps  
  = (state, ownProps) => {  
  return {  
    message: state.helloWorld.message  
  }  
}
```

Dispatch Hello's events to properties

```
const mapDispatchToProps =  
  (dispatch, ownProps) => {  
    return {  
      onClick: () => dispatch(helloWorld()),  
      reset: () => dispatch(reset())  
    }  
  }  
}
```


HelloWorld connects Hello's state and dispatcher

```
const HelloWorld = connect(  
  mapStateToProps,  
  mapDispatchToProps  
)(Hello)
```

```
export default HelloWorld
```

Implement Reducers

Dr. Jose Annunziato

Implement reducers in src/reducers/index.js

```
import { combineReducers } from 'redux'  
import { HELLO_WORLD, RESET } from '../actions'  
  
let initialState = { message: 'Hello' }
```

Implement the reducer(s)

```
const helloWorld = (state=initialState, action) => {  
  switch (action.type) {  
    case HELLO_WORLD:  
      return Object.assign({}, state,  
        { message: 'Hello, World!' })  
    case RESET:  
      return state = initialState  
    default:  
      return state  
  }  
};
```

Combine the reducers

```
const helloReducer = combineReducers({  
  helloWorld  
})
```

```
export default helloReducer
```

Provide a Provider

Dr. Jose Annunziato

Create the store and provide it to the App

```
import { Provider } from 'react-redux'  
import { createStore } from 'redux'  
import helloReducer from './reducers'
```

```
let store = createStore(helloReducer)
```

```
ReactDOM.render(  
  <Provider store={store}>  
    <App />  
  </Provider>,  
  document.getElementById('root')  
);
```