

# REDUX HOOKS

**Jose Annunziato**

# Hello Reducer

```
const hello = () => ({message: 'Hello World'});
```

```
export default hello;
```

**HELLO WORLD!**

**EXAMPLE**

**Jose Annunziato**

# Hello Component

```
import React from "react";
import {useSelector} from "react-redux";
const HelloReduxExampleComponent = () => {
  const message = useSelector((hello) => hello.message);
  return(
    <h1>{message}</h1>
  );
};
export default HelloReduxExampleComponent;
```

# Provider

```
import React from "react";  
import HelloReduxExampleComponent  
  from "./component";  
import hello from "./reducer";  
import {createStore} from "redux";  
import {Provider} from "react-redux";  
const store = createStore(hello);
```

# Provider

```
const HelloWorldReduxExample = () => {  
  return(  
    <Provider store={store}>  
      <HelloReduxExampleComponent />  
    </Provider>  
  );  
};  
export default HelloWorldReduxExample;
```

# TODO

# EXAMPLE

**Jose Annunziato**

# ToDo Reducer

```
const data = [  
  { do: "Buy milk", _id: "123", done: false },  
  { do: "Walk dogs", _id: "234", done: false },  
];
```

```
const todos = () => {  
  return data;  
}  
export default todos;
```



# Combine Reducers

```
import React from "react";
import HelloReduxExampleComponent
    from "./HelloReduxExampleComponent";
import Todos from "./Todos";
import {Provider} from "react-redux";
import hello from "../reducers/hello";
import todos from "../reducers/todos";
import {createStore, combineReducers} from "redux";
const reducers = combineReducers({hello, todos})
const store = createStore(reducers);
```

# Provider

```
const ReduxExamples = () => {  
  return(  
    <Provider store={store}>  
      <Todos/>  
      <HelloReduxExampleComponent/>  
    </Provider>  
  );  
};  
export default ReduxExamples;
```

# Todos Component

```
import React from "react";
import {useSelector} from "react-redux";
const Todos = () => {
  const todos = useSelector(state => state.todos);
  return(
    <><h3>Todos</h3>
    <ul className="list-group">
      { todos.map(todo =>
        <li className="list-group-item">{todo.do}</li>
      )}</ul></>);
  };
  export default Todos;
```

# FORMS & REACT

**Jose Annunziato**

# Maintain State w/useState

```
import React, {useState} from "react";  
const Todos = () => {  
  const todos = useSelector(state => state.todos);  
  const [todo, setTodo] = useState({do: ""});  
  const todoChangeHandler = (event) => {  
    const newTodo = {  
      do: event.target.value  
    };  
    setTodo(newTodo);  
  }  
}
```

```
return(
```

```
...
```

```
<ul className="list-group">
```

```
<li className="list-group-item">
```

```
<input onChange={todoChangeHandler}  
      value={todo.do}
```

```
      className="form-control"/>
```

```
</li>
```

```
<button onClick={createTodoClickHandler}>
```

```
  Create New Todo
```

```
</button>
```

```
...
```

```
)
```

# Creating Todos

```
import {useDispatch, useSelector} from "react-redux";  
const Todos = () => {  
  const dispatch = useDispatch();  
  const createTodoClickHandler = () => {  
    const action = {  
      type: 'create-todo',  
      todo  
    };  
    dispatch(action);  
  }  
}
```

# Handle Create Event

```
const todos = (state = data, action) => {  
  switch (action.type) {  
    case 'create-todo':  
      return [  
        ...state,  
        action.todo  
      ]  
    ...  
  }  
}
```



# Deleting Todos

```
const Todos = () => {
```

```
  ...
```

```
  const dispatch = useDispatch();
```

```
  const deleteTodoClickHandler = (todo) => {
```

```
    const action = {
```

```
      type: 'delete-todo',
```

```
      todo
```

```
    };
```

```
    dispatch(action);
```

```
  }
```

# Handle Delete Todo Event

```
<button onClick={() => deleteTodoClickHandler(todo)}>  
  Delete  
</button>
```

Handle delete-todo event

```
const todos = (state = data, action) => {  
  switch (action.type) {  
    case 'delete-todo':  
      return state.filter(todo => todo !== action.todo);  
    case 'create-todo':
```

...

# Updating Todos

```
const Todos = () => {  
  const [todo, setTodo] = useState({do: "", done: false});  
  const updateTodoClickHandler = (todo) => {  
    const action = {  
      type: 'update-todo',  
      todo  
    };  
    dispatch(action);  
  }  
}
```

# Notify on Change Todo's State

```
<input  
  checked={todo.done}  
  onChange={  
    (event) => updateTodoClickHandler(  
      {...todo, done: event.target.checked})  
  }  
  type="checkbox"/>
```

# Handle Update Todo Event

```
const todos = (state = data, action) => {  
  switch (action.type) {  
    case 'update-todo':  
      const newTodos = state.map(todo => {  
        const newTodo = todo._id === action.todo._id ?  
          action.todo : todo;  
        return newTodo;  
      });  
      return newTodos;  
    case 'delete-todo': ...
```