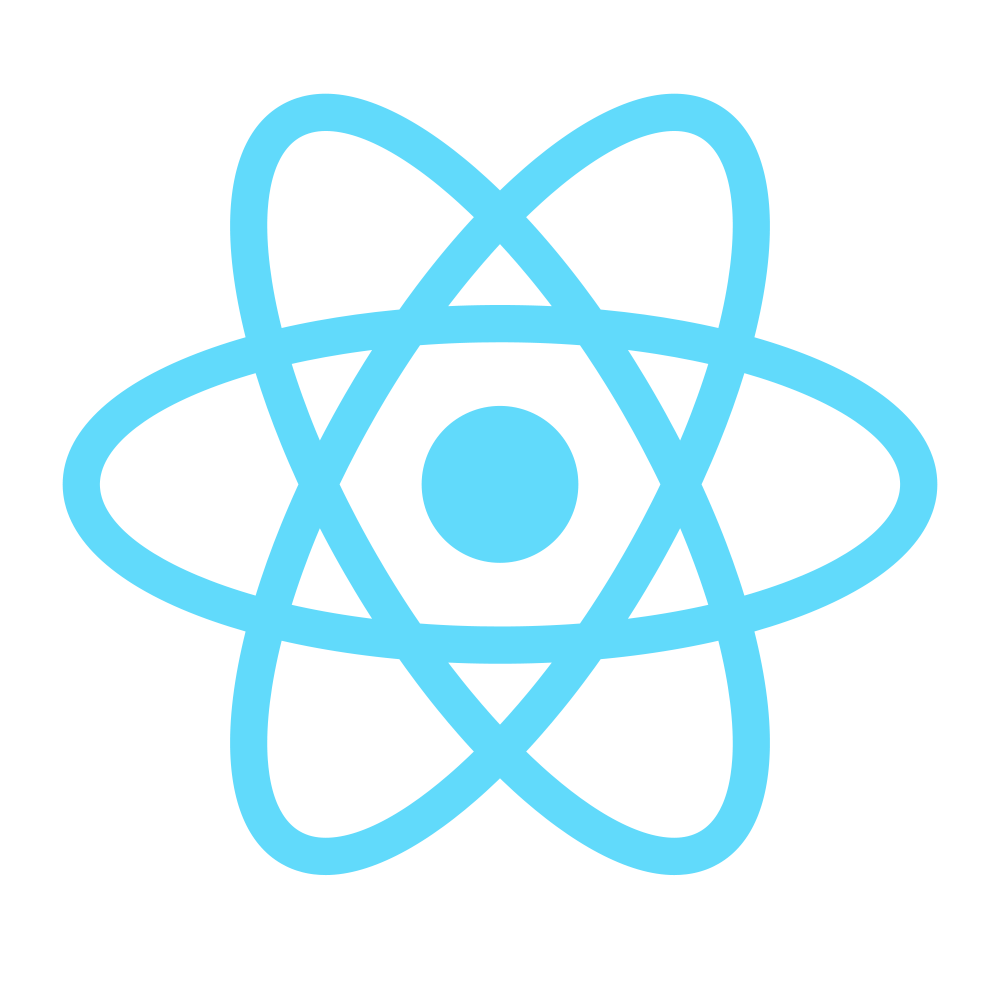
React + Redux



**Focus:** React, Redux

### Course Description

This is the course you've been looking for to master modern web development with React & Redux

**Redux?**We got it. **ES6/Babel?** Covered. **Webpack?** Included!

Mastering React and Redux can get you a position in web development or help you build that personal project you've been dreaming of. It's a skill that will put you more in demand in the modern web development industry, especially with the release of Redux and ReactNative.

This course will get you up and running quickly, and teach you the core knowledge you need to deeply understand and build React components and structure applications with Redux.

We'll start by mastering the fundamentals of React, including JSX, “props", “state", and eventing. Source code is provided for each lecture, so you will always stay up-to-date with the course pacing. After an introduction to React, we'll dive right in to Redux, covering topics like reducers, actions, and the state tree.

If you are new to React and Redux, or if you've been working to learn it but sometimes feel like you still don't quite 'get it', this is the React course for you! To learn React you have to understand it.

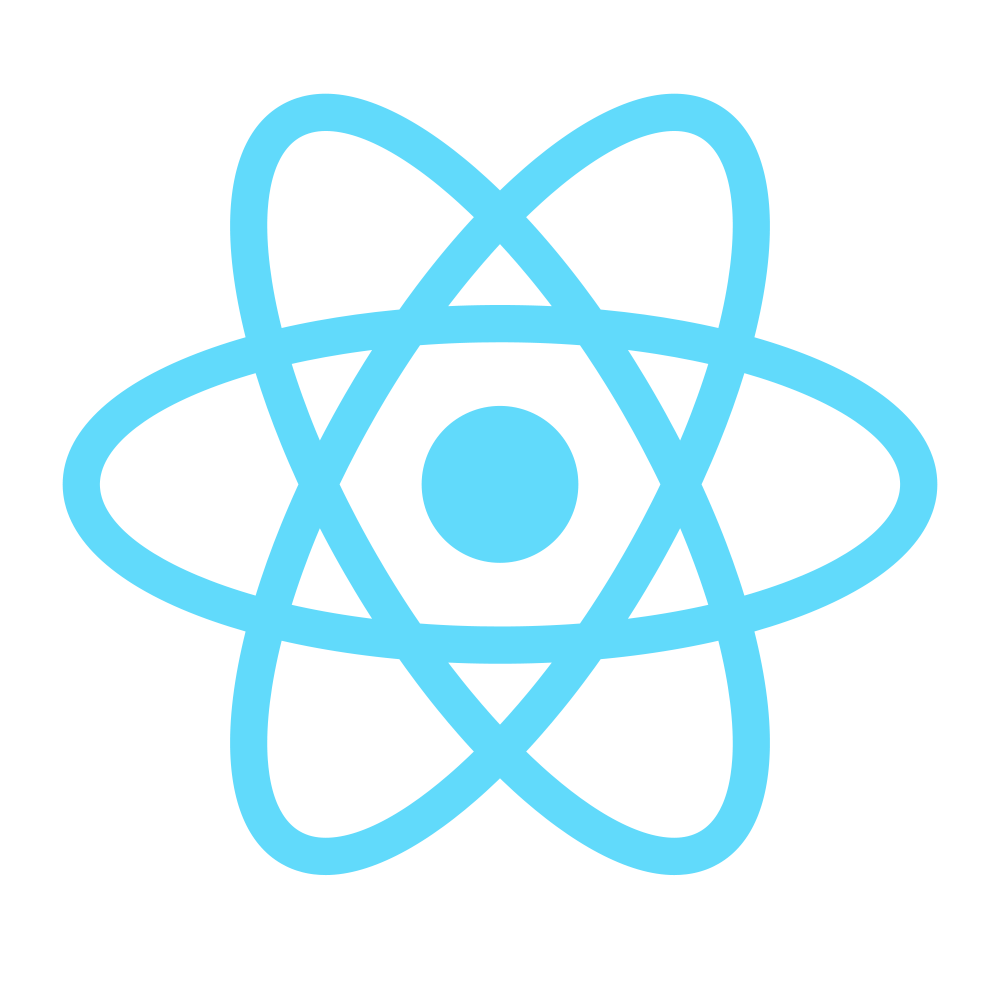
1. Learn how to use React's custom markup language, JSX, to clean up your JavaScript code
2. Master the process of breaking down a complex component into many smaller, interchangeable components
3. Grasp the difference between “props" and “state" and when to use each
4. Develop complex applications that scale in complexity by mastering Redux
5. Dive deeper into Redux by using middleware. No fancy terms required!

### What am I going to get from this course?

1. Build amazing single page applications with React JS and Redux
2. Master fundamental concepts behind structuring Redux applications
3. Realize the power of building composable components
4. Be the engineer who explains how Redux works to everyone else, because you know the fundamentals so well
5. Become fluent in the tool chain supporting React, including NPM, Webpack, Babel, and ES6/ES2015 JavaScript syntax

### What is the target audience?

* Programmers looking to learn React view components
* Developers who want to grow out of just using jQuery
* Very good knowledge, skill and experience in JavaScript OO concepts and Frameworks

** | React**

# Workshop Topics Covered

# Legend: [P] – Presentation, [D] – Demo

1. **Awesome React Ecosystem**
   1. Why React is awesome
   2. Who is using React
   3. Technology stack
   4. Measuring Speed
   5. Why learn React Comparing Angular
   6. Angula2 vs react
2. **Setup React and Webpack**
   1. Installing Starter dependencies
   2. Configure Webpack in webpack.config.js
   3. HTML webpack template for index.html
   4. Configure startup scripts in package.json
   5. Run Webpack setup
3. **ES6 React Guide**
   1. Hello World React
   2. Component Composition and naming
   3. Files and Folder structure
   4. React (Virtual) DOM Terminology
   5. Root Component - index.jsx
   6. Module import
   7. World.jsx component
   8. Class definition
   9. Event Handlers and setState
   10. JSX and the render method
   11. Properties
   12. Event UI binding
   13. PropTypes and defaultProps
   14. Styling React component
   15. Run Development server
   16. React chrome Extension
4. **Production Optimize Webpack**
   1. Add production plugins
   2. HTML & CSS minifying
   3. Webpack production webpack.config.js
   4. Loaders
   5. Profiling Webpack build
5. **Define Components Internals**
   1. Naming files and folders and modules
   2. Imports and exports
   3. Stateless components and pure functions
   4. Classes and Inheritance
   5. Constructor and Binding
   6. Properties and Property types
   7. State management
   8. Lifecycle methods
   9. Event Handlers
   10. Render and ReactDOM.render methods
   11. JSF features and syntax
   12. Accessing DOM Nodes using Refs
6. **Wire Multiple Components**
   1. Render Multiple components programmatically
   2. Composition using parent child node tree
   3. Presentational and Container components
   4. Dynamic children
7. **React Router**
   1. Route Configuration
   2. Route Matching
   3. Histories
   4. Index Routes and Links
8. **Lint React Apps**
   1. Lint config in webpack.lint.config.js
   2. JavaScript Lint using ESLint
   3. StyleLint for CSS
9. **Test App Component**
   1. Mocha Chai Behavior-Driven Development
   2. Enzyme React Component Testing
   3. JSDOM browser.js helper
   4. Sinon spy methods and events
   5. Istanbul code coverage

**| Redux - Learner Level**

# Workshop Topics Covered

# Legend: [P] – Presentation, [D] – Demo, [L] - Lab

1. **Introduction**
   1. Overview of Reactive Programming
   2. Benefits of Uni directional data flow
   3. One-way vs Two-way bindings
2. **Flux Architecture**
   1. Building blocks in Flux
   2. Dispatcher
   3. Stores
   4. Action
   5. Views
3. **Dispatcher**
   1. The Role of Dispatcher
   2. Registering callbacks
   3. Triggering Actions
4. **Store**
   1. The role of store
   2. Maintain Application State
   3. Event Broadcasting
5. **Reducer**
   1. Creating Reducers
   2. Retrieving State
   3. Combining Reducers
6. **Action Creators**
   1. Creating Actions
   2. Dispatching Actions to dispatcher
   3. Dispatching Async Actions
   4. Using Middlewares
7. **Views**
   1. React Components as Views
   2. State Subscribers
   3. Updating data from stores
8. **React-Redux Module**
   1. React-React bindings
   2. Binding Components to Store using ‘Connect’
   3. Injecting Redux store using Provider
   4. Using Providers with React Routers
9. **Redux Testing**
   1. Organizing Test code using Mocha & Chai
   2. Testing Stores
   3. Testing Reducers
10. **More…**
    1. Using ‘Immutable.js’
    2. Consuming REST API
    3. Overview of other ‘flux’ implementations
11. **Advanced**
    1. Hot Reloading Components in browser
    2. Debugging concepts
    3. Server-Side Rendering
    4. React and Redux best practices
    5. Making App production Ready
    6. Review
    7. Walk thorough on final case-study