The beginners

GUIDE TO REACT & REDUX

The Complete React Web Development Course (with Redux)

Section 1

REACT 101

- Install Visual Studio Code
- Install Node.js
- Setting up a Web Server
- Setting Up Babel
- Exploring JSX & JSX Expressions
- ES6 Variables & Arrow Functions
- Manual Data Binding
- Forms & Input
- Arrays in JSX

In this section we will dive into the very basic fundamentals of React and setting up your own web developer environment to write and execute your own code. We will go through installing a very popular text editor release by Microsoft called Visual Studio Code to installing Node.js on your machine and then onto setting up your very own local web server and installing Babel.

Once we have setup your developer environment we will continue to learn the fundamentals of JavaScript (ES6) and JSX syntax to build our foundational skills on both languages and technologies on which the React Library is built upon.

After learning the basics we will move onto the next sections where we will start learning more about the React component architecture and will build our own applications.

It will soon become apparent why React is considered a very popular frontend JavaScript library as you are not required to learn a new language but rather learn an application architecture based on components.

INSTALLING VISUAL STUDIO CODE

There are many free and paid text editors out there available to download and use to write your own code, popular editors include Atom, Sublime Text, Notepad++ and Visual Studio Code to name a few. We will be using Visual Studio code as our text editor of choice to write our code as it allows us to install some useful packages to help write our code.

To install Visual Studio code simply visit the following website and download and install the application onto your machine: https://code.visualstudio.com/

Visual Studio Code has some useful extensions which you can install:

- Bracket Pair Colorizer
- ES7 React/Redux/GraphQL/React-Native Snippet
- Liver Server
- Prettier - Code Formatter

INSTALLING NODE.JS

Node is JavaScript on the server. You can visit the following website to download node onto your machine: https://nodejs.org/en/

Download the latest version of node that is available on their website.

To check that you have node.js installed on your machine, simply open up your terminal and enter the following command:

```
$ node -v
```

This will allow us to double check that node was installed onto our machine as we now have this new command and it also shows us what version of node you have installed on your machine. When installing node we also got NPM (node package manager) which allows us to install various dependencies/packages such as React or Yarn and other libraries. NPM and Yarn aims to do the same job. To check that you have npm installed enter the following command in your terminal:

```
$ nom -v
```

To install yarn on your machine globally run the following command in your terminal:

```
$ npm install -g yarn
```

On windows machines you will need to restart your machine to complete the installation. To check that yarn has installed successfully, run the following code in your terminal:

```
$ yarn –version
```

SETTING UP A WEB SERVER

To setup a developer web server we can achieve this in two ways using live-server.

Firstly you will need to create a directory (folder) for your application. This folder will act as a place for all your project code. This folder can be called anything for example 'indecision-app'. In this example we will create a sub-folder called public and store our basic HTML file.

If we open VS Code and have installed live-server extension we can simply open up our html document and right-click to open the file with live-server. Every-time we update our project files in the folder, the live-server will refresh the browser which will update our application with the changes automatically.

Alternatively, we can use npm or yarn to install live-server globally onto our machines using either command in the terminal:

```
$ npm install -g live-server     or     $ yarn global add live-server
```

To check that we have installed live-server on our machine correctly we would run the command:

```
$ live-server -v
```

To run live server from the terminal, navigate to your file directory using cd and the file path. Note: you can use **cd** to change directory, **ls** (or **dir** on windows) to list all the files within the folder. You can use **cd ~** to navigate back to your user folder. Once you have navigated to your project directory run the following code:

```
$ live-server public
```

Note: you would run live-server followed by the folder name in the directory you wish to serve through the live web server (in our example we had a sub-folder called public which contained our HTML file). Any changes made in the folder will automatically update in the browser.