## React Key Concepts

### The Birth of React.js

Previously we have **jQuery** which allows developer easily interact with the DOM across all the browsers.

Then the library like **backbone.js** came up because our JavaScript files gets bigger and bigger it helps to organize this JavaScript files.

**SPA**

Because it becomes easier and easier to work with DOM and also **Ajax** came, we have the birth of **SPA.** Basically we focus less on HTML and lot more on JavaScript; we only load the application code once instead of making new request to the server where it return new document instead now our application acted more like desktop application where we stay on the same page the entire time and the js file simply changes or update the html file/DOM to display new things

In 2010 **Angular.js** which created by google really became a standard way of building the application. For that we have better container where things like controller, views and models and this idea of MVC where each JavaScript files is divided into different things it did.

Since the App got bigger and bigger we have more complexity, using Angular.js is very tough to find the bug and the data is flowing everywhere. In 2013 Facebook releases **react** and the solution is really good and 2014 angular.js completely rewrite to **angular**

### React Concepts:-

* **Don’t touch the DOM, I will do it.**
* **Building website like lego blocks. (Reusable component).**
* **Unidirectional Data Flow.**
* **UI, The rest is up to you.**

#### Declarative vs Imperative

* + **Imperative**

Many existing framework and library before react where directly manipulating the DOM on every page and this way of programming we referred as **Imperative** as we directly change the individual parts of the app in response to various user events. **Problem** is it becomes difficult to see the relationship between events and the edge cases.

* + **Declarative**

Instead of Imperative react came up with concept called **declarative**, as we know DOM manipulation leads to performance issue, it takes a long time. So react say just declare the how the app looks like (give me the state), I will do everything like find the best way to use the DOM. i.e. I contains one JavaScript object that describe how the app should look like

This resulted in:-

1. Less complexity
2. Better code quality
3. Faster Developer time

#### Component Architecture

React is all about where we have multiple components surrounded one another and we can also reuse them in other places/inside another component.

Component is nothing but a JavaScript function that receive some input/props and return an html but from a JavaScript.

|  |
| --- |
|  |

#### One Way Data Flow

Means if any state changes it will from top to bottom it can never go from bottom to top (means it can never go up).

One way data flow is a restriction because of this it easier to understand how the app works.

|  |
| --- |
|  |

#### UI Library

React care about component, virtual DOM and I am only going to work with the view, everything else we need can use other module, library mix and match to the need.

React 360:-<https://facebook.github.io/react-360/>

React blessed: - <https://github.com/Yomguithereal/react-blessed>

React Desktop: - <https://reactdesktop.js.org/>

#### How to be a great react Developer

|  |
| --- |
|  |

## React Basics

### Creating the react app

<https://create-react-app.dev/docs/getting-started/>

* + Old way

npm install –g create-react-app

This is the package which creates a startup project that contains webpack, babel, lint and debugging.

Create-react-app <project name>

* + New ways is

First uninstalled the create-react-app: **npm uninstall -g create-react-app**

Then type the below command with project name

**npx create-react-app <app-name>**

<https://create-react-app.dev/docs/getting-started/>

### Understanding project structure and files

* + **In Package.json**
* **react-scripts library** it helps us not to worry about webpack and babel.
* **react library** helps to write html like syntax inside a JavaScript file.
* **react-dom library** helps to interact with the DOM.

|  |
| --- |
| {  "name": "monsters-rolodex",  "version": "0.1.0",  "private": true,  "dependencies": {  "@testing-library/jest-dom": "^4.2.4",  "@testing-library/react": "^9.5.0",  "@testing-library/user-event": "^7.2.1",  "react": "^16.13.0",  "react-dom": "^16.13.0",  "react-scripts": "3.4.0"  },  "scripts": {  "start": "react-scripts start",  "build": "react-scripts build",  "test": "react-scripts test",  "eject": "react-scripts eject"  },  "eslintConfig": {  "extends": "react-app"  },  "browserslist": {  "production": [  ">0.2%",  "not dead",  "not op\_mini all"  ],  "development": [  "last 1 chrome version",  "last 1 firefox version",  "last 1 safari version"  ]  }  } |

* In **source folder** where we write all the react code/files for the application
* In **public folder** where the browser needs the older version of JavaScript and html to understand and the also the build files will be present.
* **npm start** to start the project.
* **npm run build** to build the file for the production.
* **npm run eject** it will show all the webpack , test config files. But don’t it as the files changing/updating is doing by react team member.

### Components

Component can be create using function as well with class. Reason why we use class because react has given ability to write more functionality verses the function return html

* **Function way**

Step 1:- Create a js file

1. It needs to import React and component.
2. Create function which return the html
3. Export the function.

Adding CSS file to the component.

1. Create a new css file and import it in the component

|  |
| --- |
| import React from 'react';  import logo from './logo.svg';  import './App.css';  function App() {  return (  <div className="App">  <header className="App-header">  <img src={logo} className="App-logo" alt="logo" />  <p>  Edit <code>src/App.js</code> and save to reload.  </p>  <a  className="App-link"  href="https://reactjs.org"  target="\_blank"  rel="noopener noreferrer"  >  Learn React  </a>  </header>  </div>  );  }  export default App; |

Step 2:- How to use in other file.

All remember in React application we will have React package as well as ReactDOM to connect react to the DOM

|  |
| --- |
| import React from 'react';  import ReactDOM from 'react-dom';  import './index.css';  import App from './Hello';  import \* as serviceWorker from './serviceWorker';  ReactDOM.render(<App />, document.getElementById('root')); |

* **Class way**

Step 1:- Create a js file

1. It needs to import React and component.
2. Create class which extends Component.
3. Export the class.

Adding CSS file to the component.

1. Create a new css file and import it in the component

Hello.js

|  |
| --- |
| import React, {Component} from 'react';  import './App.css';  class App extends Component {  render() {  return (  <div>  <p>Welcome to React</p>  <h1>Hello World</h1>  </div>  );  }  }  export default App; |

Step 2:- How to use in other file.

All remember in React application we will have React package as well as ReactDOM to connect react to the DOM

|  |
| --- |
| import React from 'react';  import ReactDOM from 'react-dom';  import './index.css';  import App from './App';  import \* as serviceWorker from './serviceWorker';  ReactDOM.render(<App />, document.getElementById('root')); |

## Resources

### UI Library

React 360:-<https://facebook.github.io/react-360/>

React blessed: - <https://github.com/Yomguithereal/react-blessed>

React Desktop: - <https://reactdesktop.js.org/>

### Web Developer Monthly News

<https://zerotomastery.io/blog/?tag=WDM>

### Yarn

It is the replacement of npm created by facebook

<https://classic.yarnpkg.com/en/docs/install#windows-stable>

### IE11 issue

<https://medium.com/@matwankarmalay/create-react-app-ie11-script1002-syntax-error-how-to-get-rid-of-it-d70000c53ddf>