Intro to React

What Is React?

React (also known as React.js or ReactJS) is an open-source, front-end JavaScript library for building user connections or UI components. It is maintained by Facebook and the community of individual developers and various other companies. React can be used as a basis for creating single-page or mobile applications. However, React is only concerned with the management of the state and providing that nation to the DOM, so creating React applications often requires additional router libraries and some customer service.

React Features:

- **Declarative**: React makes it less painful to create interactive UIs. Design a simple view of each state in your app, and React will carefully review and provide relevant sections as your data changes. Screaming views make your code look more accessible and easier to fix.
- **Component-Based**: Create integrated objects that control their state and compose them into complex UIs. Since the concept of components is written in JavaScript instead of templates, you can quickly transfer rich data through your application and keep the state out of DOM.
- **Learn Once, Write Everywhere**: React does not bother about all of your technical stacks, so you can develop new features in React without rewriting the existing code. React can also render the server using Node and the power of mobile applications using React Native.

What are Components?

Components are independent and reusable codes. They work for the same purpose as JavaScript functions but work independently and restore HTML with render () function. In simple words, react is like the lego game, and here components are bricks of lego that are used to build different applications.

Components are of two types:

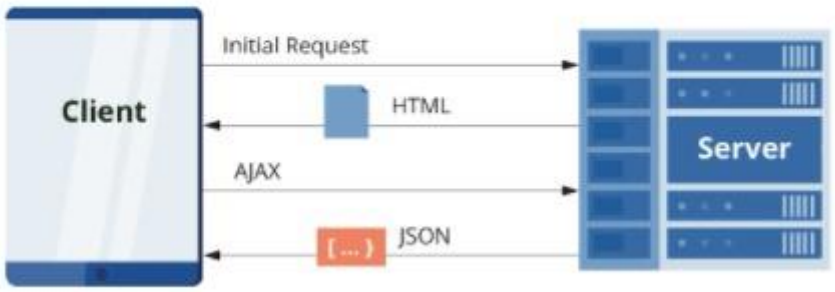
Class components - The class component name must start with a capital letter. This component should contain the React.Component statement, which creates the inheritance for the React.Component and gives your component access to the functions of the React.Component. The component also requires a render () method, which provides HTML.

Function components- A functional component is just a plain JavaScript function that accepts props *(arguments passed into React components)* as an argument and returns a React element. There is no render method used in functional components.

Single Page Applications Vs Multi-Page Applications

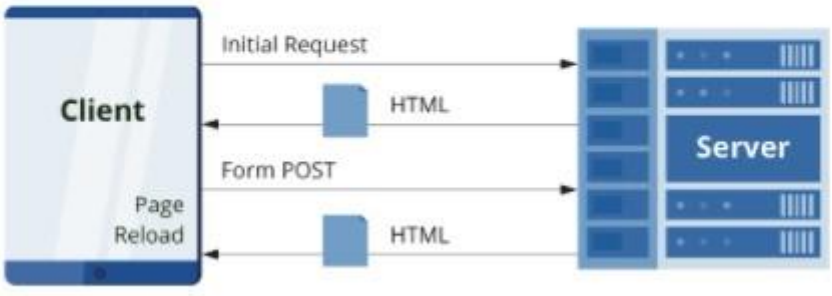
Single Page Applications: A single page application is an application that works inside the browser and does not require reloading the page during use. SPAs are about providing excellent UX by imitating the "natural" environment in the browser - no page reloads, no extra waiting time. It loads the web page you visit and then all other content using JavaScript - much depends on it.

SPA Lifecycle



Multi-Page Applications: Multi-page applications work in a "traditional" manner. Each change (for example, displaying data or submitting data back to the server) prompts the server to provide a new page in the browser.
These applications are larger than SPA.

Traditional Page Lifecycle



Single Page Applications	Multi-Page Applications
SPAs provide increased content load speed because they do not have many pages and load content at once.	In multi-page applications, content is constantly loaded, which increases the load on your server. This can adversely affect web page speed and overall system performance.
Single-page app development is easy because you need to create fewer pages, create less functionality and test and display less content.	Multi-page applications have more features than single-page applications. Therefore, more effort and resources are required to make them. Development time increases in proportion to the number of pages created and the activity to be executed.
Single-page app developers have trouble indexing a website properly	Multi-page applications are more SEO-friendly than single-page
and achieving high search rankings.	applications. Their content is constantly updated. In addition, they have many pages for adding various keywords, images, meta tags.
It is easy to maintain at a low cost.	It is difficult to maintain and is not budget-friendly.

It has the ability to work offline if there are some problems with the internet connection, as it loads all the data at once.	It always requires an internet connection as it does not load all the data at once.

Real DOM vs Virtual DOM

Document Object Model

The DOM is an abstraction of a page's HTML structure. It takes HTML elements and wraps them in an object with a tree-structure — maintaining the parent/child relationships of those nested HTML elements.

Virtual Document Object Model

The Virtual DOM is a light-weight abstraction of the DOM. You can think of it as a copy of the DOM, that can be updated without affecting the actual DOM. It has all the same properties as the real DOM object, but doesn't have the ability to write to the screen like the real DOM. The virtual DOM gains it's speed and efficiency from the fact that it's lightweight. In fact, a new virtual DOM is created after every re-render.

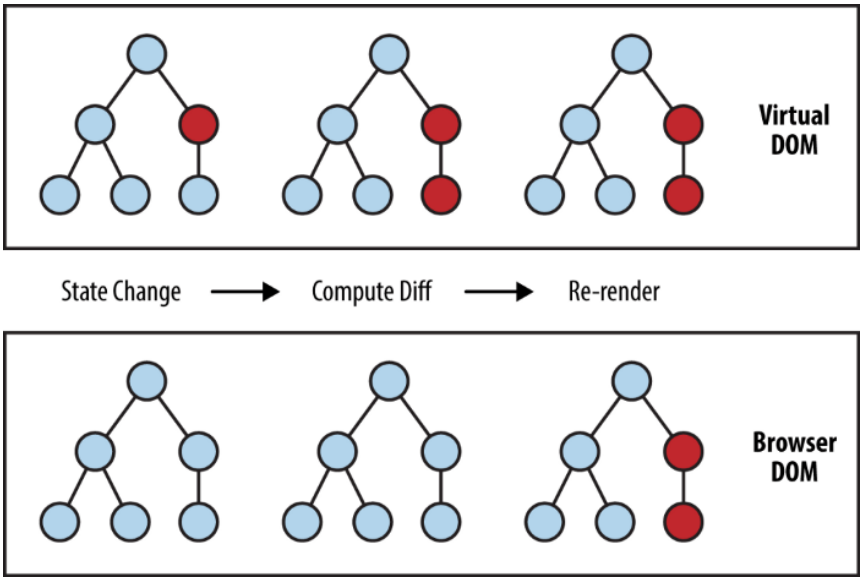
The virtual DOM (VDOM) is a programming concept where an ideal, or "virtual", representation of a UI is kept in memory and synced with the "real" DOM by a library such as ReactDOM.

Real DOM	Virtual DOM
It represents the UI of your application.	It is only the virtual representation of the DOM. It is similar to a lightweight duplicate of DOM.
It updates slow.	It updates Faster.

It can directly update HTML.	It cannot directly update the HTML.
Creates a new DOM if element updates.	Updates the JSX if element updates.
Too much memory wastage.	No memory wastage.

How does updates work in React?

- On the first load, ReactDOM.render() will create the Virtual DOM tree and real DOM tree.
- As React works on Observable patterns, when any event(like key press, left click, api response, etc.) occurred, Virtual DOM tree nodes are notified for props change, If the properties used in that node are updated, the node is updated else left as it is.
- React compares Virtual DOM with real DOM and updates real DOM. This process is called Reconciliation. React uses Diffing algorithm techniques of Reconciliation.
- Updated real DOM is repainted on browser.



React compares the Virtual DOM with Real DOM. It finds out the changed nodes and updates only the changed nodes in Real DOM leaving the rest nodes as it is. This process is called **Reconciliation**.

React Vs Other Frameworks

What is a framework?

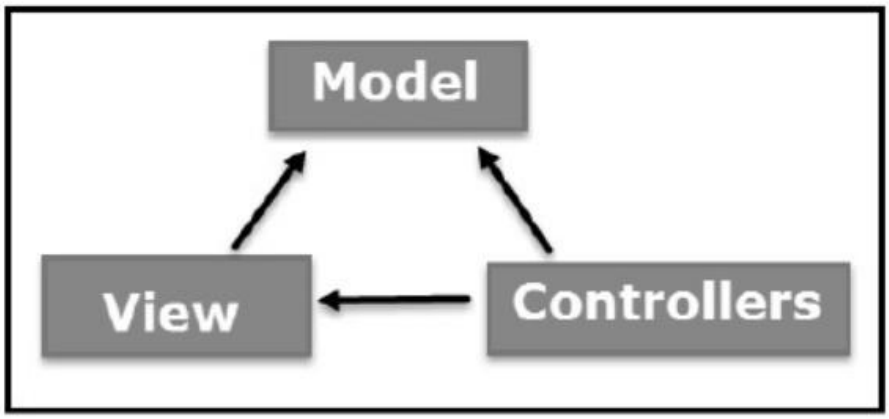
The **framework** defines open or non-executed functions or objects that the user writes to create a custom application. Since Framework is an application, it has a wide range and includes everything needed to create a user application tailored to its need.

What is Library?

The library provides a set of auxiliary functions/objects/modules that your application calls for specific functionality. Libraries usually focus on a narrow range. The reason we need the library is very simple, i.e. code reuse, use code already written by other developers.

What is the MVC framework?

The Model-View-Controller (MVC) is an architectural model that separates an application into three main logical components: the Model, View and Controller.



- **Model:** The model component consists of all data-related logic that works with the user.
- **View:** The view component is used for all UI logic of the application.
- **Controller:** The controllers act as an interface between the model and the view components to process all business logic and incoming requests, modify the data using the model component and interact with the view to display the final output.

So **React** is a library that is just the **view** part of the MVC framework, whereas other frontend frameworks like angular is an extensive framework that is more than the view part of the MVC framework. According to Google statistics, react is easy to learn as it is not so vast and is the most searchable front-end library.

Some Terminal Commands

- **pwd :** PWD stands for Print Working Directory. It prints the path to the working directory starting from the root.
- **ls :** ls is a shell command that lists directory contents of files and directories.
- **cd :** The CD command in Linux is called the Change Directory Command. It is used to change the current working directory.
- **clear :** clear is the standard operating system command used to clean the terminal screen.

Summarising It

Let's summarise what we have learnt in this module:

- Learned about React library and components.
- Learned about single page and multi-page applications.
- Differentiated between framework and library.
- Compared React with other frontend frameworks.

Resources

Lecture Resources:

- Visual Studio Download Link: https://code.visualstudio.com/download
- NodeJs Download Link: https://nodejs.org/en/download/

Further Readings:

- Single page apps v/s Multi page apps: https://asperbrothers.com/blog/spa-vs-mpa/
- React v/s Angular Comparison: https://hackernoon.com/reactjs-vs-angular-comparison-which-is-better-805c0b8091b1