Eric Zorn

10/17/2017

ICT 4300

Professor/Dr. Lakhani

**The Frameworks Which Changed the DOM**

**ICT 4300: Web Enabled Information Systems**

**February 24, 2012**

The web framework is one of the most revolutionary concepts that has been brought to the internet browser. For a long time, web pages were being built statically with HTML and CSS. HTML stands for Hyper Text Markup Language and CSS stands for Cascading Style Sheets. It was only recently that other languages that act on both the client side and the server side had been introduced as crucial languages. The web’s most popular language that is being used in modern day is JavaScript. To understand where most of these frameworks are coming from and to understand the power that they convey, you have to understand the history of the languages that they were built in. Currently, JavaScript is becoming and arguably is the most powerful programming language on the planet to date. The language is not to be confused with the Java programming language, which is much older. JavaScript was created in just ten days during the month of May in 1995. The language was not always known by the name of JavaScript. Its original name was known as Mocha which had been given by the founder of Netscape, Marc Andreesen. The name was then later changed to Live Script and then eventually JavaScript shortly after. The company ECMA eventually bought the rights to the language and added their own changes/updates to the language. It then adopted the name of ECMAScript. At the point, the language gained a large amount of interest from the development community and began to be tested by developers around the world. Interestingly enough, Microsoft had mentioned that they had no intention to fully incorporate the language into their Internet Explorer browser. Instead, they were planning on fully implementing the .NET server-side language to take over the powerful processes that JavaScript would have done. JavaScript then became an open sourced language and had become extremely criticized and taught by a famous man known as Douglas Crockford who had been an employee of Yahoo at the time. Not long after, the term AJAX was coined. This is still used today, where the JavaScript language is used to send and retrieve data over the server in the background and negating the need for the page to fully reload. This allows for the applications to be more dynamic and useful to the different users of the application. Now, due to the fact that JavaScript has become open sourced, there have been many different libraries and frameworks that have been developed that are also open-sources. Two of the most famous JavaScript libraries that have been developed are DOJO and jQuery. Both of these libraries help perform the tasks of Vanilla JavaScript with writing less code that has already been written as different methods and properties by the library developers (W3.org, June 27, 2012).

Now that we know some of the history behind the JavaScript language and the origination of the different libraries that have been used, we can dive into the most important topic on the internet today. This topic is about the different frameworks that are used through the web browser and the internet for a full-stack experience. To date, there are many different frameworks and libraries that are open sourced to the public for people to be able to write dynamic web applications. Most of them are written in JavaScript or variations of JavaScript like TypeScript or JSX. However, there are other frameworks that have been created with other languages like Python and Ruby. We will dive into these as well. As of 2017, Hackernoon states that the top five JavaScript frameworks are Angular-s (Angular.js and Angular 2) made by Microsoft with their TypeScript language, React JS built by Facebook (Focusing on UI with a virtual DOM for high traffic management), Vue.js (evolved from ember and combines the knowledge of React and Angular with optional JSX support for single-page applications), Ember.js (Allowing for UI performance and server-side rendering of the DOM with use from companies like Chipotle, Blue Apron, Nordstrom, Kickstarter, LinkedIn, Netflix and many others), and last but not least, Meteor.js (Great for back-end development and front-end development. It is developed with modules that are going to be fully written with pure JavaScript. It is used by many companies like Mazda, Honeywell, and IKEA (Hackernoon, Eugenia Korotya, Jan 19, 2017). Some of the other frameworks that are widely known and used today are derived from the Python and the Ruby programming languages like Django and Rails, otherwise known as Ruby on Rails. Django’s most notable feature is that it is a server-side web framework that is written in Python. Django can be written on pretty much any operating system and uses many different unique features like models to represent the data structures that we would store our app’s data in. Django also comes prepackaged with an admin site to store and control data that is moving across the server and the database. Django also comes pre-equipped with its own security protocols and methods (Mozilla Developer Network, 2017). Ruby on Rails is a web framework that has been built on top of the Ruby programming language that utilizes syntax from Smalltalk, Python, and Perl. It is considered to be high-level and is object-oriented. The Ruby on Rails framework is interpreted and therefore cannot be considered to be as high level as Java and or C or C++. It is said that you can develop a Rails application at least ten times faster than you would be able to with a standard Java framework. The Rails framework is most well-known for its back-end capabilities and database integration (Tutorials Point, 2017).

We will dive further into each of these frameworks individually as we examine the underlying fundamentals of how many of these frameworks are run and executed. As mentioned before, many of these languages and or frameworks are known to be interpreted and not compiled in the browser or before execution to the browser. This in and of itself, makes programming with these languages faster and more dynamic than other high-level programming languages that require being compiled to machine language before running. Most of these frameworks also use a concept of MVC or Model View Controller. Essentially, this is an architectural pattern that will spilt the application into three different components. There is the model, the view, and the controller as the name states. Each of these different components will be responsible for handling different tasks for the application to properly run in the internet browser. MVC is one of the most frequently used standards for web development frameworks because of its easily scalable nature for frameworks. The model component corresponds to all of the data-related logic that the user is going to be working with. In the model, this data can represent what is being transferred between the view and the controller or any business related logic. Tutorials Point provides us with the example of a customer object retrieving the customer information from the database, manipulating it, and then updating it back to the database or using the information to render a value to the DOM. Second, is the View. The view component is used to supply and render all of the logic that will be displayed to the UI or User Interface. This will control the textboxes, dropdown menus, or any other interaction that the user will formally engage in with the DOM. The controller is the interface that calculates and executes all of the business logic and or incoming requests. It then manipulates the data using the Model component and will send the data to the view controller to display the output of the data in the organized manner. The example that Tutorials Point provides us for the controller is, the customer controller will handle the interactions/inputs from the view and then update the database with the model logic. They also state that the same controller will be used to view customer data (Tutorials Point, 2017).

Many of these MVC frameworks are known to work as single-page applications. The acronym for this is known as SPA. This is simply when the application will be accessed via a web browser but become more dynamic than your typical standard web page. Today, it is very hard to distinguish the difference with the extreme push for dynamic web applications over your standard website. SPA applications often use AJAX more heavily to communicate to the back-end server without having to do a full page refresh. Most of the client-side JavaScript is still able to be run while the server is being called with the database to update the DOM. This is where the benefit of a virtual DOM in React may come in handy with faster load times. Most all of these pages will be rendered on the client-side. There are cons to SPA applications. Although they tend to be very trendy and faster loading than your typical refreshed browser page, it can cause havoc to the browser. The browser in these applications will do most of the “heavy lifting” says Code School. This in turn will mean the performance can be compromised, especially on some of the less powerful mobile devices. Some of the other cons come with Search Engine Optimization or SEO. Developers will have to ensure that the content can be discoverable with specific links to the content in search engines or social media websites that provide different link previews. To prevent the burden of performance that can be called with AJAX from the client-side is the implementation of server-side code. In this scenarios, the code will be run, interpreted, and executed on the server which will then export the proper code to the browser page or DOM. This will mean that the applications can be loaded and used quicker for the user. It will also help for SEO discoverability (Code School, 2017).

Let’s begin talking specifically about the extremely popular framework that we have already mentioned of Angular JS and Angular 2/3. Angular has become one of the neatest and widely used frameworks to date. Angular’s website has a great instruction and example of why Angular is being used over other libraries and frameworks on the market. The developer gave an example compared to jQuery. The example entailed an input and an HTML tag for a heading. In order to live update, the DOM with the value of the input, we had to identify the input ID and the heading ID by storing both of them in variables. We then set an event listener in on the input variable for key up. Inside of the key up method, we wrote a callback function to take the value of the input and place that value as the innerHTML of the heading tag. For people who are either new to jQuery or JavaScript, this can be a slightly tedious process and confusing to some people. The developer then demonstrated what the process would look like in Angular. We would simply reference the Angular framework over a CDN or download the entire framework and save it to a sub-directory in your project’s root directory. Once the framework is referenced in your HTML document, you would simply initialize the DOM to read as an angular application by writing “ng-app” directly next to the word html in the parent html element in the page’s HTML structure. On the input, you would write an attribute of ng-model and give it a name of whatever you would like. In the heading, you would simply write {{name}} with the word name being whatever you assigned the name to be with the ng-model. Now this sounds a bit more confusing than it actually is. Angular in this example makes the DOM resemble pretty much the original versions of HTML code with nothing written in script tags. It executes and interprets all of the proper JavaScript code behind the scenes in the browser and makes for more efficient code with a cleaner aesthetic. Angular obviously can become much more complicated than the example above but is extremely powerful. The company behind the framework actually will pride itself on being an MVW framework which would stand technically for Model View Whatever (Himanshu Bhalla, Passionate Software Engineer, July 20, 2016). Developers are arguing if Angular JS has moved to MVVM for Model-View-ViewModel which is an architectural design pattern for implementation of user interfaces. It focuses on separation of concern between the UI and the Data/Model (Stack Overflow, 2017). It also adds an extra label known as the View Model to enhance the manageability, scalability, and testability. HTML and CSS may simply make a page look great with static content, but that is no longer acceptable over the internet in the browser. Dynamic web pages are needed and Angular JS allows you to extend the HTML vocabulary and elements in your application. It allows for more dynamic and interactive web pages for the user with ease of coding for the developer (AngularJS, 2017).

Next, let’s begin speaking about the other famous framework that is being utilized at the moment. This is known as React.js and has been built by the ever so popular social media company, Facebook. React is also an MVC framework. On the React.js website, it states that React makes painless and interactive UIs. React uses a virtual DOM, so therefore any action that trigger a UI change would automatically be handled and updated from the framework. Many developers actually consider React to be more of a library and less of a framework, due to most all of its purpose being for the front-end and less of the back-end. Much of the framework/library is written in JavaScript or specifically the ES6 model, which is relatively new. For developers who are coming into web development from other object-oriented languages, the React framework will be very self-explanatory. Although the language is written in JavaScript and would require at least an intermediate knowledge of the JavaScript language, there is also a newer version known as JSX. JSX is known as JavaScript and HTML, where the developer can physically type HTML markup directly into their JavaScript files. The files are built in components and are told to change their state depending on their UI. The React team also states that when building React based applications, you can develop new features in React without having to rewrite the existing code. There are many drawbacks to building applications with React. The number one drawback however is the fact that it is not a complete/full-stack framework like Angular. This in itself is why the two different frameworks cannot be comparable to one another. React at its core is used to update the front-end or the DOM. Essentially what React is built to do is take the algorithms and data that have been written in JavaScript for each individual component and then update the DOM by exporting different HTML. The browser will then be able to interpret the HTML and read it to the DOM as HTML markup and Vanilla JS with ES5 or ES6 standards. This like Angular and the other frameworks, ensures a unique and dynamic experience with the virtual DOM and quickly updated data. As you would assume, this is demonstrated through the Facebook web application. All of the different components/views are built separately and rendered into the DOM/UI. Some of these might consist of the messages view, the newsfeed, the side navigation bars, and many more. All of this data is simply rendered from the server with frameworks like Back Bone JS and then manipulated on the client-side DOM (Back Bone JS, 2017). Due to the fact that React mainly is manipulating the view, there is no event handler methods built into the framework other than Vanilla JS events, there is no easy way to perform AJAX calls, and there are no full-stack capabilities for a back-end database and or server (Andy Ray Blog, September 2014). React also uses Flux. Flux is described by Andrew Ray’s Blog as where your view will dispatch an action like an event and then update your DOM based on that action. An example of this would be when a user types in their information into a text field and then tells your data to store and update that data. When the data is updated on the client-side, it then triggers an event where the view will retrieve the data and re-render the DOM. React also is not supported in browsers that are below IE8 which may make viewing the applications difficult for people with older devices or for people who have not updated their application and operating system software. Personally, I believe that building applications is difficult and learning something new is not going to be an easy task for anyone to complete. I believe that if you are building applications that need a very dynamic UI with different components, it is smart to be using React plus Flux. During the Hacker Way presentation on Rethinking Web App Development by Facebook, the Facebook Developers brought up an interesting point. They mentioned that the Facebook application in the past had mainly been built using PHP and then sending that data to be rendered by the browser with client-side JavaScript (Hacker Way: Rethinking Web App Development at Facebook, May 4, 2014). Now, using Back Bone to handle the different views and model event bindings along with React and Flux, developers can build these dynamic UIs and send the data back to the server with their server-side language like PHP, Ruby or Python. The database will update and then send the data back down to the client-side, where React will then update the virtual DOM and the UI. All in all, this comes down to preference for the developer, but React can be very powerful for web developers. As mentioned in the conference/presentation as well, many companies like Airbnb, Venmo, and Instagram are using React as their main framework for their web applications.