**BILAL ARSHAD**

**L16-4064**

**1.MONGOOSE VS MONGODB**

Mongoose is built upon the MongoDB driver to provide programmers with a way to model their data.

Using Mongoose, a user can define the schema for the documents in a particular collection. It provides a lot of convenience in the creation and management of data in MongoDB. On the downside, learning mongoose can take some time, and has some limitations in handling schemas that are quite complex.

However, if your collection schema is unpredictable, or you want a Mongo-shell like experience inside Node.js, then go ahead and use the MongoDB driver. It is the simplest to pick up. The downside here is that you will have to write larger amounts of code for validating the data, and the risk of errors is higher.

In MongoDB, **Insert operations** target a single collection. All write operations in MongoDB are atomic on the level of a single document.

db.users.insertOne (

{ name:”sue”,

age:26,

status:”pending”

})

**Read Operation:**

db.users.find(

{{age:{$gt:18}},

{name:1,address:1}

).limit(5)

**Update Operation:**

db.users.updateMany(

{ age:{$lt:18}},

{$set:{status:”reject”}}

)

**Delete Operation:**

db.users.deleteMany(

{status:”reject”} )

**2.POST VS PUT**

Both PUT and POST can be used for creating.

You have to ask "what are you performing the action to?" to distinguish what you should be using. Let's assume you're designing an API for asking questions. If you want to use POST then you would do that to a list of questions. If you want to use PUT then you would do that to a particular question.

**Great both can be used, so which one should I use in my RESTful design:**

You do not need to support both PUT and POST.

Which is used is left up to you. But just remember to use the right one depending on what object you are referencing in the request.

Some considerations:

* Do you name your URL objects you create explicitly, or let the server decide? If you name them then use PUT. If you let the server decide then use POST.
* PUT is idempotent, so if you PUT an object twice, it has no effect. This is a nice property, so I would use PUT when possible.
* You can update or create a resource with PUT with the same object URL
* With POST you can have 2 requests coming in at the same time making modifications to a URL, and they may update different parts of the object.

**3.PUT VS PATCH**

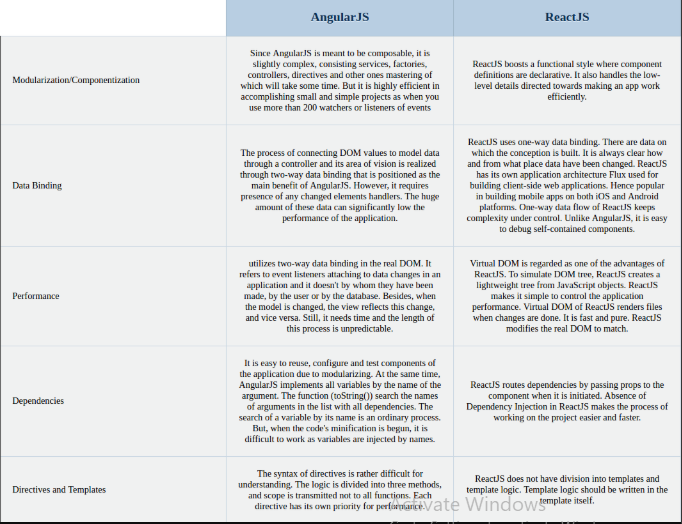
**PUT:** The http PUT method only allows a complete replacement of the document. It is used when user knows all the fields, or answers.

**PATCH:** Used to modify an existing http resource i.e. for updating little bits at a time.

Partial Update: If PUT is used, by sending the entire resource with updated values, there is a chance of unnecessary bandwidth being consumed. PATCH is the preferred option here.

Similarly, PATCH should be used to update name in an online university application. Here, a single field is being updated and not the entire document. PATCH will speed up this process.

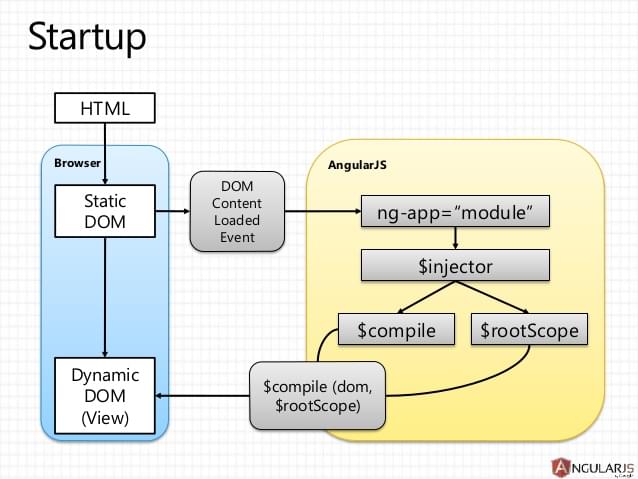
**4.Comparison of ReactJS with AngularJS**



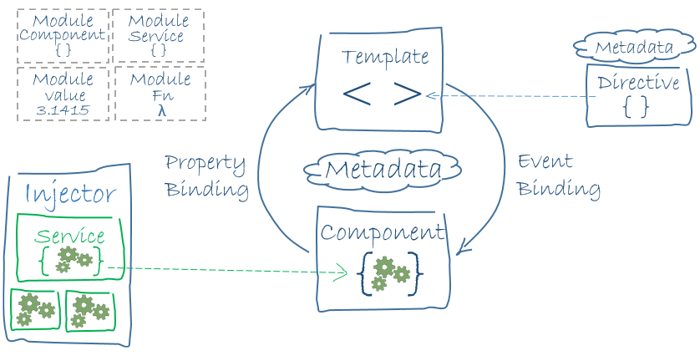
**6. AngularIO vs AngularJS**

AngularIO is a platform that makes it easy to build applications with the web. Angular combines declarative templates, dependency injection, end to end tooling, and integrated best practices to solve development challenges. Angular empowers developers to build applications that live on the web, mobile, or the desktop. Angular, on the other, hand has a component-based architecture. Every Angular application has at least one component known as the root component. Whereas, AngularJs a different framework, which is the ancestor of Angular. AngularJS is the first version of Angular. It works with JavaScript and it is still getting supported but it is not compatible with Angular. AngularJS is a client side framework of JavaScript which is based on MVC Architecture and provides a Single-page application solution for dynamic websites.

**FrameWorks and architecture of AngularJS**



**FrameWorks and architecture of AngularIO**



**5.Vue.js**

Vue is a **progressive framework** for building user interfaces. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects.

* **History of each framework**  
  React is a JavaScript library for building UI components for web applications. React is used by Facebook, Uber, Netflix, Twitter, PayPal, Reedit, Tumblr, Wal-Mart and many more.  
  Angular is a TypeScript-based JavaScript framework backed by Google. It is an enormously popular framework for front-end development. Angular is used by Google, Forbes, WhatsApp, Instagram and many more.   
  Vue.js can be used to build attractive UIs using HTML, CSS and JavaScript. Vue is used by Alibaba, GitLab, Baidu, and appreciated by developers and designers globally.
* **Support and Growth of each framework**  
  As React is powered by Facebook and Angular is maintained by Google, there’s no doubt on the growth of both of these frameworks. In both frameworks, updates and release are published frequently but they are well maintained when it comes to migrations.   
  If we consider Vue.js in this way, there’s a migration helper tool which makes migration easier. But in the large app, it might cause a problem as there is no proper roadmap which focuses on versioning and their plans.
* **Popularity of each Framework**   
  According to the 2017 Stack overflow survey, Angular is loved by 51.7% of developers and React is embraced by 66.9% of surveyed developers. React and Angular have almost the same level of users in the category of front-end frameworks. Vue hasn’t occupied a place in any of the above lists but has an ability to participate in this battle.
* **Difference between Frameworks and libraries**

With React, you can do multiple integrations as you can pair, exchange, and integrate libraries with other great tools out there. At this point, React works out of the box due to its flexibility to offer seamless integration but, with this, there are more chances of to wrong and it requires more dependencies.

Angular is a framework because it provides you with a good start to build an application with the complete setup. You don’t need to look into libraries, routing solutions, and the structure. React and Vue, on the other hand, are more flexible and universal than Angular.

Vue is the cleanest because it not only helps you keep your code efficient with the perfect balance of internal dependencies and flexibility but is also a very easy and straightforward JavaScript framework which aims to simplify web development.

**7. Linting? What is the use of JSLint? What is the use of ESLint?**

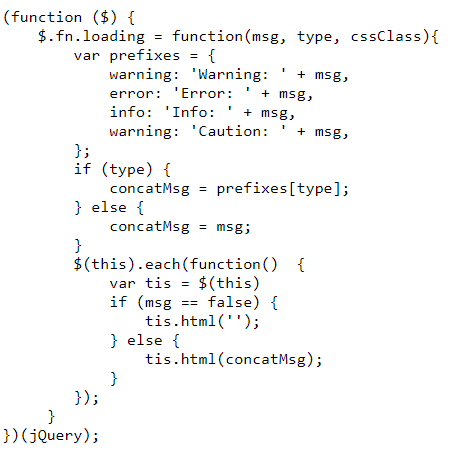
**Linting** is the process of running a program that will analyze code for potential errors. Lint was the name originally given to a particular program that flagged some suspicious and non-portable constructs (likely to be bugs) in C language source code.

**JSLint** is a JavaScript program that looks for problems in JavaScript programs. It is a **static code analysis** tool used in software development for checking if JavaScript source code complies with coding rules.

**ESLint** is a tool for identifying and reporting on patterns found in ECMAScript/JavaScript code, with the goal of making code more consistent and avoiding bugs. In many ways, it is similar to JSLint and JSHint with a few exceptions such as ESLint uses Espree for JavaScript parsing, ESLint uses an AST to evaluate patterns in code and ESLint is completely pluggable, every single rule is a plugin and you can add more at runtime.

**Examples:**

**1.**

****

**2.**

****

**8.** **AngularJS over AJAX and vice versa**

Similar to the jQuery, angularJS is also based on JavaScript. But it's scope in terms of web development is much broader than jQuery. It is actually a complete framework which would change the way you write your code (both html as well as JavaScript part).

AJAX is a JavaScript concept for fetching data from server. There are predefined methods both in Angular as well as jQuery for implementing AJAX.

Following are the conditions where I would prefer each framework over each other:

**AJAX** = a way to retrieve data from a web server without reloading the entire web page;

**AngularJS** = JavaScript framework to build single page applications, that is webpages that act like an application (a calculator, a to-do list, …)