

WEb programming research Assignment

Sajjad Zaidi



December 8, 2018

1. *Mongoose vs MongoDB*

In terms of Node.js, mongodb is the native driver for interacting with a mongodb instance and mongoose is an Object modeling tool for MongoDB. Mongoose is built upon the MongoDB driver to provide programmers with a way to model their data.

*Why are we using mongoose*

We opted for mongoose because it uses Object Relational Mapping (ORM) whereas mongodb uses Object Document Mapping (ODM). Having a fixed schema gives a structure and more maintainability to your application code. Its development time is also far better than mongodb.

*CRUD Operations*

Insert operations

MongoDB provides the following methods to insert documents into a collection:

db.collection.insertOne()

db.collection.insertMany()

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

Read Operations

Read operations retrieves documents from a collection; i.e. queries a collection for documents. MongoDB provides the following methods to read documents from a collection:

db.collection.find()

You can specify query filters or criteria that identify the documents to return.

Update Operations

Update operations modify existing documents in a collection. MongoDB provides the following methods to update documents of a collection:

db.collection.updateOne()

db.collection.updateMany()

db.collection.replaceOne()

In MongoDB, update operations target a single collection. All write operations in MongoDB are atomic on the level of a single document. You can specify criteria, or filters, that identify the documents to update. These filters use the same syntax as read operations.

Delete Operations

Delete operations remove documents from a collection. MongoDB provides the following methods to delete documents of a collection:

db.collection.deleteOne()

db.collection.deleteMany()

In MongoDB, delete operations target a single collection. All write operations in MongoDB are atomic on the level of a single document. You can specify criteria, or filters, that identify the documents to remove. These filters use the same syntax as read operations.

1. *POST VS PUT*

The fundamental difference between the POST and PUT requests is reflected in the different meaning of the Request-URI. The URI in a POST request identifies the resource that will handle the enclosed entity. In contrast, the URI in a PUT request identifies the entity enclosed with the request.

When you know the URL of the thing you want to create or overwrite, a PUT method should be used. Alternatively, if you only know the URL of the category or sub-section of the thing you want to create something within, use the POST method.

In view of idempotence nature of PUT, POST was utilized in the employee AJAX application to refresh records. Subsequently we could update an employee the same number of times as we needed, had we utilized PUT, it would just give result one time.

If we replace POST with PUT, we have to specify URI for newly created or updated resource.

1. *PUT VS PATCH*

PUT is used to update the entire entity whereas PATCH can update a set of attributes.

PATCH would be preferred in this particular case described in question as you only want to update a single attribute.

1. *React vs AngularJS*

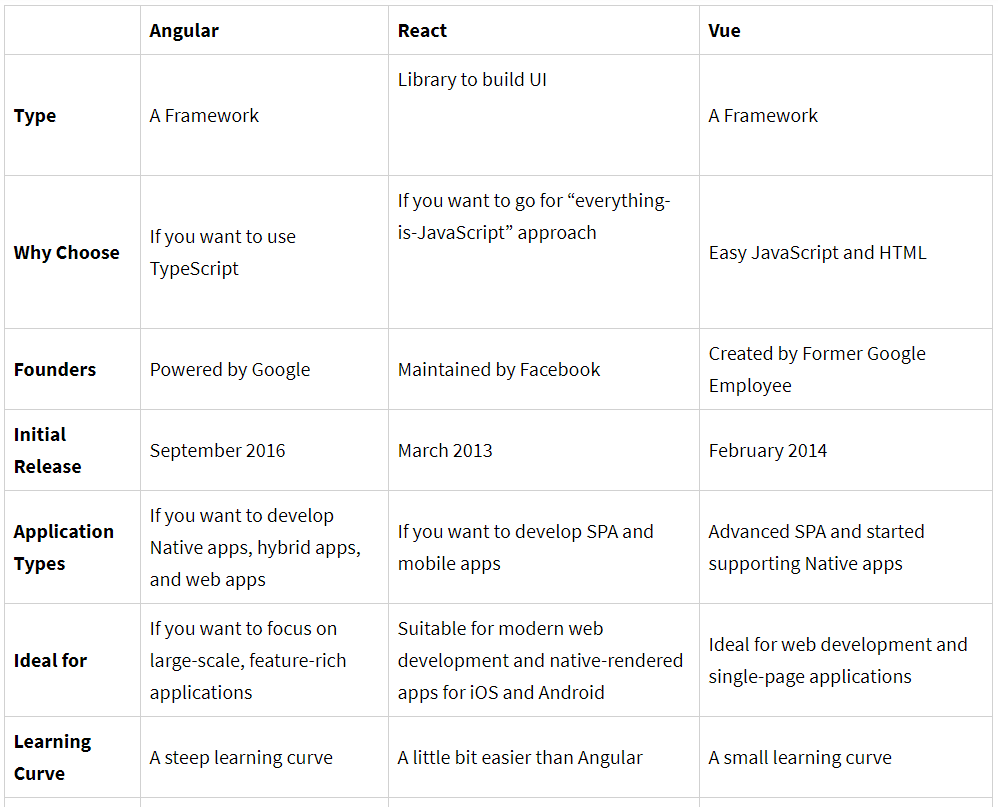
React is a library for building composable user interfaces. It encourages the creation of reusable UI components which present data that changes over time. It is not a complete application framework like angular, it is just a view layer. So, it is not directly comparable to frameworks like angular. Some people say that comparing React and Angular is like comparing apples to oranges. While one is a library that deals with views, the other is a full-fledged framework.

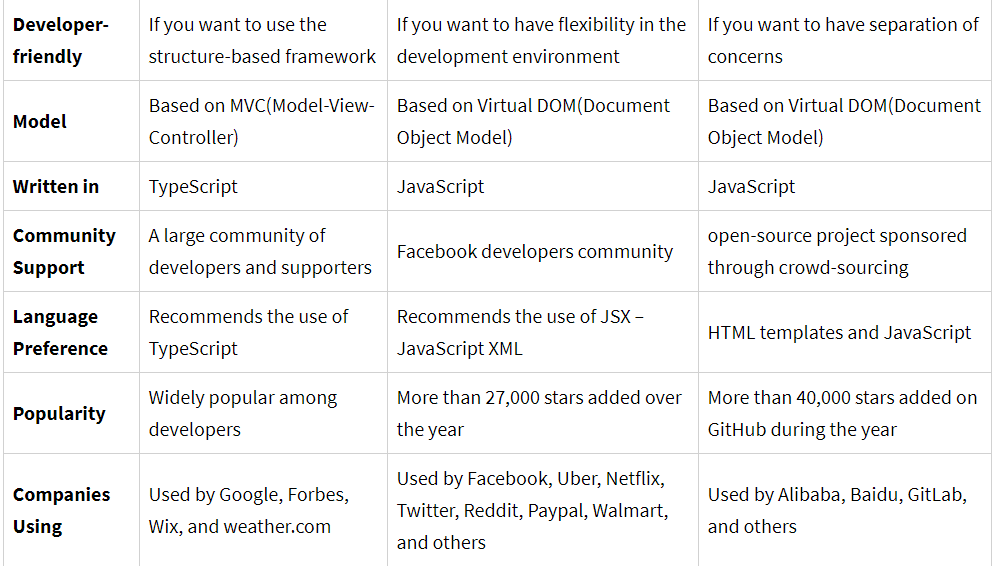
Angular offers a long list of in-built features. Having all of these features available out of the box is highly convenient when you don’t want to spend time picking the libraries yourself. However, it also means that you’re stuck with some of them, even if you don’t need them. And replacing them will usually require additional effort. For instance, we believe that for small projects having a DI system creates more overhead than benefit, considering it can be effectively replaced by imports.

1. *Vue.js*

Vue.js is an open-source JavaScript framework for building user interfaces and single-page applications.

*Comparison*





*Example*

<script>

new Vue({

el: '#container',

data: {

value: '',

},

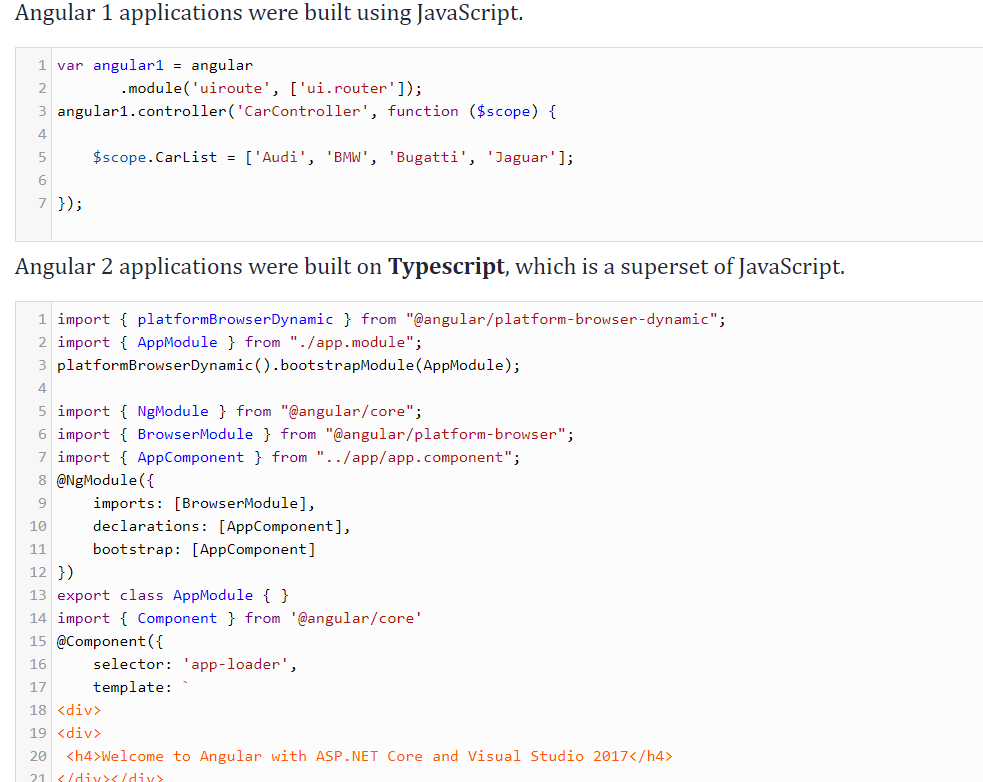
});

</script>

1. *AngularJS vs Angular 2*

AngularJS is an open-source, JavaScript-based, front-end web application framework for dynamic web app development. It utilizes HTML as a template language. By extending HTML attributes with directives and binding data to HTML with expressions, AngularJS creates an environment that is readable, extraordinarily expressive and quick to develop.

Angular IO is open-source, TypeScript-based front-end web application platforms. In Angular 2, controllers and $scope were replaced by components and directives. Components are directives with a template. They deal with a view of the application and logic on the page. There are two kinds of directives. These are structural directives that alter the layout of the DOM by removing and replacing its elements, and attributive directives that change the behavior or appearance of a DOM element.

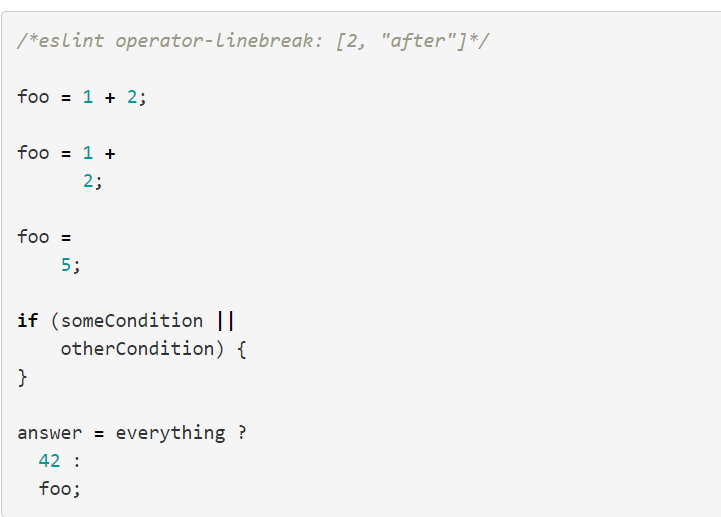


1. *Lint*

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. The term is now applied generically to tools that flag suspicious usage in software written in any computer language.

*JSLint* is a static code analysis tool used in software development for checking if JavaScript source code complies with coding rules. It is provided primarily as a web application through jslint.com, but there are also command-line adaptations. It was created in 2002 by Douglas Crockford.

*ESLint* is an open source project originally created by Nicholas C. Zakas in June 2013. Its goal is to provide a pluggable linting utility for JavaScript.



1. *Angular VS AJAX*

Angular is a full-fledged framework whereas ajax is only used to communicate with the server. So, in a small website one should only make use of ajax whereas in a bigger website one should prefer angular as it allows to produce reusable UI components and makes maintenance easier.