IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

An investigation on Single Page Application development using The Angular Framework

[1]Shiv Shankar Shah, [2]Upasna
[1,2]Department of Computer Science and Engineering, Ganga Institute of Technology and Management, Kablana.

Abstract

Angular is an open-source, component-based JavaScript framework for building modular, scalable single-page applications written in TypeScript. The main use of it is to develop SPA and Google maintains its version. As a framework, Angular has obvious advantages while also providing developers with a standard for use. It allows users to build large applications in structured way.

This research paper aims to analyses the current state of web UI development using the Angular framework in general. We write Angular applications by writing HTML templates, writing classes to manage these templates, adding application logic to services, and mapping objects and services in templates. We will launch the application by booting the base module. Angular renders your app's content in the browser and responds to user interactions based on the instructions you provide.

Angular usage has exploded because of its unique features. Angular helps build interactive and powerful single page applications (SPAs) through its core features such as template rendering, two-way linking, modularization, RESTful API processing, injection success, and AJAX processing.

Keywords: AngularJS, Angular

Introduction To Angular Framework

Angular is completely Model-View-Controller (MVC), component-based, an open-source framework that uses Typescript language which allows us to build single page applications. It provides clear instructions on how to build applications and provides information on both while rendering the actual DOM.

In Single Page Application approach, we must load the application only once. After loading the application in browser, if we do any navigation, we will not see any reloading thing on the browser. This approach gives us fast loading of application and help in great user UI rendering experience for all the users.

Angular is totally depended on components which makes the developer work easy and within very less time developer can manage it as we develop the application using component-based structure. First, AngularJS came in the picture in 2010 which was the first release of angular and its first version. It has used the JavaScript frontend language which doesn’t support mobile application. Later in 2016, Angular second version came into the picture then we were able to develop mobile based application too. In Angular application, we have different modules, components, services, pipes, directive, interceptor, router module etc. where we can separate our code forming a tree structure with child and parent combination and with no such relationship.

We can say that an Angular application is the tree to component and these components are further enabled to add behavior to UI through services, pipes, directives, dependency injection, and modules are logical unit of a big application, many modules are tied together to build a robust Angular Application. AngularJS which is the first version of Angular created the Single Page Application which brought revolution in internet market and IT industry. It is a client side that is browser side JavaScript framework and known as AngularJS. After first version AngularJS changed to Angular in 2016. Angular is not same as AngularJS. It is written in Typescript and Angular also uses Typescript for the component class. Other than the main development team, Angular 2 and above versions are almost identical in the development and Angular have nothing in common in structure. We are getting some changes in every six months. We have a template view, CSS/SCSS file and class component together merged in @Component decorator. We apply directive in template view. We do dependency injection in the class constructor and create private reference to use the service. We can create service and share common function between components just by injecting the services.

Angular 15 is the latest version of Angular which was released in November 2022. It contains general functionality of Angular 2+. These versions are called Angular. Angular is almost used in all types of the application like Health care, Banking, School, Accounts, Chat library, Online Booking Application etc. These apps are very convenient and help us in our daily life by giving users a good experience and connectivity. Angular improves the performance as it provides lazy loading strategy if application is complex. Each Angular version improves on the previous version, fixes bugs, fixes issues and adapts to the complexity of the current platform. If we want to build more mobile-friendly apps and/or more complex apps, you'd better upgrade to the current version. The development process aims to adapt to design standards and easy-to-use reusable web applications. We can improve the performance of the application by making the build in the production mode. We can get minified vendor.js file by the production mode build and, we get very few files in the build output dist folder. Day by day some improvement we are getting in the Angular framework which is very helpful in the development and fast rendering of the application. Ivy Engine which was Angular 8 engine minified the build and renders the module very fast as compared to previous.

Components are the foundation root of the application which helps in faster and manageable code development. Components include TypeScript classes, HTML templates, and styles with the @Component () decorator. It teaches Angular how to make a component HTML template. Angular architecture protects angular component to inherit the styles as the component is encapsulated by the default. By default, all the component is encapsulated with emulated. If we are doing the development in the component structure, then we will have easy to write unit test cases and it becomes very easy to test. Every Angular component has an HTML format file which acts as a view in UI that tells what the component behavior will be in the browser. In HTML we can show some dynamic data by binding or string interpolation of the variable defined in the class component.

Components often define part of the user interface (UI). Component is the main thing that holds template, class, and styles files configuration in @Component decorator. Every Angular app is made up of one or more Angular components and all of them are objects. If a view component has a selector value (name), it can access data from JavaScript/TypeScript classes or objects. Inside the component we can create the methods, handles the UI part. We can make API call in the class file and, we have style file CSS file where we can design the element and change the behavior of the element.

We can do property binding using square brackets and string interpolation by double curly braces. Angular will show the updated value in the UI when the variable value changes. We can use square brackets in the html element where we can bind its attribute with the value defined in the class component. We can define some event listeners to change the color of the element on like mouse in and mouse out. Angular helps in writing the HTML, logic and styles in different files which is easy to maintain and debug the errors. All the business logic we can put into the class component file, elements tags in the html file and styles in the CSS or SCSS file and we can give the file path in the meta data of the component decorator. This concept makes the code of Angular clean and complexity to fix the problem also reduces. A new developer also can understand the project easily.

Research View on Angular Framework

Angular is a mature framework with good design and out-of-the-box functionality, including HTTP requests, build utilities, dependencies, animations, internationalization, and test runs. The Angular development team decided to rely on some external libraries and make them an integral part of the framework. Originally Angular was released as a framework where users had to purchase a license. At that time Angular had several clients and they decided to release it as an open-source library to increase usage. One of the advantages of Angular is that it is a framework developed by Google, there are experts at Google to help developers solve their problems, and it provides a large open community for developers. Angular provides developers with an efficient way to build a web application and provides developers with end-to-end tools that help web developers build both front-end and back-end. Angular is a JavaScript library whose purpose is to create an efficient front-end using the Model-View-Controller design pattern.

Angular 2 is designed to be functionally efficient. AngularJS is built on a platform whose features cannot be improved further. The relationship between AngularJS and Angular 2 has been broken since Google decided to build Angular 2 on a different platform to keep working, but there are many ways to improve AngularJS apps. The safe way is to use ngUpgrade to build the app when changing components to Angular 2 model. It is required to plan to update the catalog for a part. The aim is to ensure that the equipment does not use non-essential programs; the length of the control application should be reduced and most importantly it should contain only part of the data.

Angular 2 uses server-side rendering instead of client-side rendering, which will reduce the application's load time. This is done by changing the key. Consider a static website where the entire page must be rebuilt while browsing the page, this is where server-side rendering comes in handy. When browsing an application that uses server-side rendering, only the changed content is restored, while other parts of the application (such as the navigation bar and footer) are not restored, saving performance. Angular 2 is designed to be optimized for mobile devices using framework dependencies, which brings significant improvements. Angular 2 also provides tree-based one-way key detection which provides usability. Dependency Injection is a feature introduced by Angular 2. Its main function is to provide dependencies on objects, not convert them to real objects. This is better when reusing, checking, and testing equipment.

Design The Component with Angular Material

We can make layout easily with the Angular Material and it is one of the readymade libraries of Angular or it is user interface library created by Angular team to help Angular developers to build modern applications in a structured and functional way. By using this library, we can improve the user experience for end users, thus making our app popular. The library contains ready-to-use components that can be used out of the box with little or no additional code. We can use Angular command to install packages of this library and it provides good support and all the API example we can find in https://material.angular.io/. We can install this library by typing ‘ng add @angular/material’ command in the CMD terminal by going into that directory.

We can create beautiful user interface of the web application. We don’t need to create different component for each kind of button. Angular material provides the list of button category, and we can just pick the button and put the selected button directive on the button tag. This will reduce the project side as number of components is reused by Angular material API. First Angular team created Material library in 2014 for AngularJS and later when

Angular 2 came in 2016 then Angular material was created for this which is more advanced and easier to implement in the project. We the developer task has become easy as we can implement the component just by looking the documentation available in https://material.angular.io/. If we have some requirement of responsive UI, we can go for Angular Material library as it provides responsiveness UI. Back then, it was tagged as AngularJS to make apps more attractive and run faster. Google rewrote the code from the ground up and removed JS, then changed its name to Angular Material in September 2016. The UI/UX component is called Angular Materials.

Dependency Injection

Dependency injection help us to inject the services or any library dependency in the constructor of the component and create one private reference variable and start using in the component. It doesn't take care of the object creating at run time. All these activities are managed by the Dependency Injection module of the Angular Framework. This helps component just to inject the code and start using it. We can write clean, flexible, and maintainable code in the project. We must practice a lot with the dependency injection so that we can easily use in our component. We have reference concept in the Angular which help in the creation of the object at run time. We can also inject the services into the constructor and its object created is only one.

Modularity And Working of An Angular Application

Angular apps have a base module called AppModule which provides the boot process to start the app. Components or services in the context of an Angular module are classified as components or services. In the module we can do the grouping of the built in modules, services, pipes, components, directives etc. We can have many modules in the project. One which is created by default is the root module and other is the feature module. We must import CommonModule for feature module and BrowserModule for root module. There can be only one BrowserModule in the project and other will be common module.

The working of Angular application is simple as every Angular app consists of a file named angular.json. This file will contain all the configurations of the app. While building the app, the builder looks at this file to find the entry point of the application. Inside the build section, the main property of the options object defines the entry point of the application which in this case is main.ts. The main.ts file creates a browser environment for the application to run, and, along with this, it also calls a function called bootstrapModule, which bootstraps the application. These two steps are performed in the following order inside the main.ts file:

```
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
platformBrowserDynamic().bootstrapModule(AppModule);
```

In the above line of code, AppModule is getting bootstrapped. The AppModule is declared in the app.module.ts file. This module contains declarations of all the components. As one can see in the above line, AppComponent is getting bootstrapped. This component is defined in app.component.ts file. This file interacts with the webpage and serves data to it. After this, Angular calls the index.html file. This file consequently calls the root component that is app-root. The root component is defined in app.component.ts. The HTML template of the root component is displayed inside the <app-root> tags.

The Conclusion – Single Page Application

Angular is a great framework developed by Google. It's easy to create single page apps and collaboration between backend and frontend. There are many reasons companies choose to use Angular to build large, non-trivial applications, and each organization has slightly different reasons why the framework works well for their team. Getting to know someone with Java or .NET experience can be as simple as Angular in TypeScript. Or optimizing using a device-oriented approach with critical state management is what the team needs to tackle a major problem in the application. There are a lot of things we haven't listed in this guide yet, because if you don't find yourself in some unusual use, you can take advantage of them without thinking. For example, we probably wouldn't consider a new conversion three to five times faster than AngularJS, unless we need to do something very complex.

Most of us won't need to touch the dependency addition process and bypass it at the protection level, but we can if we must. Most of us won't need to write our own custom renderers for Angular, but it's possible and done. The strength of the framework lies in the fact that many of the benefits come naturally to business developers so that they can, almost obviously, be tracked. A developer who didn't crash a large ES5 JavaScript application while tackling challenges. The communication between Angular component is very easy. I have used @Input decorator for sending data from parent to child component. To send data from child to parent component I have used @Output decorator. If there is no such relation of component or if we want to communicate between any levels of component, we can use Redux or services of RxJS library. We can create Behavior Subject or Subject to do communication between the components. We can subscribe the Observable of this object and we can read inside subscribe of this observable. This is very convenient to do communication between the Angular projects.

References

[1] Anton Moiseev, Yakov Fain, 2016, “Angular Development with TypeScript,” 2nd ed., New York, USA, 2016.

[2] Asim Hussain, “Angular: From Theory to Practice”, USA, 2017

[3] Venkata Keerti Kotaru. "Angular for Material Design", Springer Science and Business Media LLC, 2020

[4] WAGNER, Joshua. Reduce JavaScript Payloads with Tree Shaking. 2020. Available also from: https://developers.google.com/web/ fundamentals/performance/optimizingjavascript/tree-shaking.[Online; accessed 20-December-2020].

[5] GAUNT, Matt. Service Workers: An Introduction. 2020. Available also from:https://developers.google.com/web/fundamentals/primers/service-workers . [Online; accessed 20-December-2020].

[6] GOOGLE, LLC. Why Use Server-side-rendering? 2020. Available also from: https://angular.io/guide/universal # why - use - server - side - rendering. [Online; accessed 20-December-2020].

[7] GOOGLE, LLC. Angular Documentation. 2020. Available also from: https://angular.io/docs. [Online; accessed 20-December-2020].

[8] FACEBOOK, INC. React Documentation. 2020. Available also from: https://reactjs.org/docs/. [Online; accessed 20-December-2020]

[9] MASARYK UNIVERSITY. KYPO Cyber Range Platform. 2020. Available also from: https://crp.kypo.muni.cz/. [Online; accessed 20-December-2020]

[10] HUMBLE, Jez; FARLEY, David. Continuous delivery: reliable software releases through build, test, and deployment automation. Addison-Wesley, 2015.