

An Interpretation and Anatomization of Angular: A Google Web Framework

Anirudh Bhaskar¹, Manjunath A.E²

¹B.E. Department of Computer Science and Engineering, R.V. College of Engineering

²Assistant Professor, Department of Computer Science and Engineering, R.V. College of Engineering

Abstract - With the rapid advancement in technology and more focus being emphasised upon user satisfaction, it is vital for various web application frameworks to constantly keep themselves updated in order to meet the needs of the current market. With a plethora of web frameworks available, it is essential that the developer chooses the correct framework. In this paper, a detailed analysis of the origin, history, features and advantages of the Google produced framework, called Angular is investigated. Along with this, a discussion as to why Angular is preferred for Single Page Applications is also given.

Key Words: Angular, Single Page Applications, Web frameworks, HTML, TypeScript, Client side and Server Side routing.

1. INTRODUCTION

A couple of years before its initial release, a Google employee going by the name of Miško Hevery, created a famous web framework called AngularJS used solely for internal purposes. The purpose of AngularJS was to build appealing web applications for the several projects Miško was working on. However a few years later, in the year of 2010, a team spearheaded by Miško started making web applications using AngularJS. It was then decided to release AngularJS as an open source software which will be available to the public thereby enabling everyone to make fluid web applications. Half a decade later, with various advancements in JavaScript and the release of newer and faster frameworks such as React caused many problems for AngularJS. AngularJS was just a small project that was created at google for building simple web applications. It was not able to cope up with the demands of large enterprise applications thereby causing it to go obsolete. The core team at Google went on to take an executive decision and write a brand-new framework from the ground up and it was called Angular 2 (called "Angular" nowadays). The first release of Angular 2 was released on 14th September 2016. In this paper, an in-depth analysis of Angular is carried out.

2. SINGLE PAGE APPLICATIONS

Before diving into the details of Angular, it is important for one to know what a Single Page Application (SPA) is. In [1], an SPA is defined as a web application which is comprised of many individual components and these components can be replaced independently thereby preventing the entire web page to reload upon a user action.

In the olden days there were a plethora of HTML files which were interlinked and upon every user action the web page used to reload and go to the respective HTML file. This took a lot of time, and the end users were left dissatisfied.

The next question that arises is that, how do we implement SPA's? The answer is simple. We make use of the concept of client side routing. Client side routing is a type of routing in which when a user carries out an action, at any given instance the full webpage will not be reloaded. Instead, JavaScript or TypeScript will update the URL and then display the required content. One must keep in mind; Angular can be used to build a variety of applications. Single Page Applications just happens to be one of the more famous applications used in the world today.

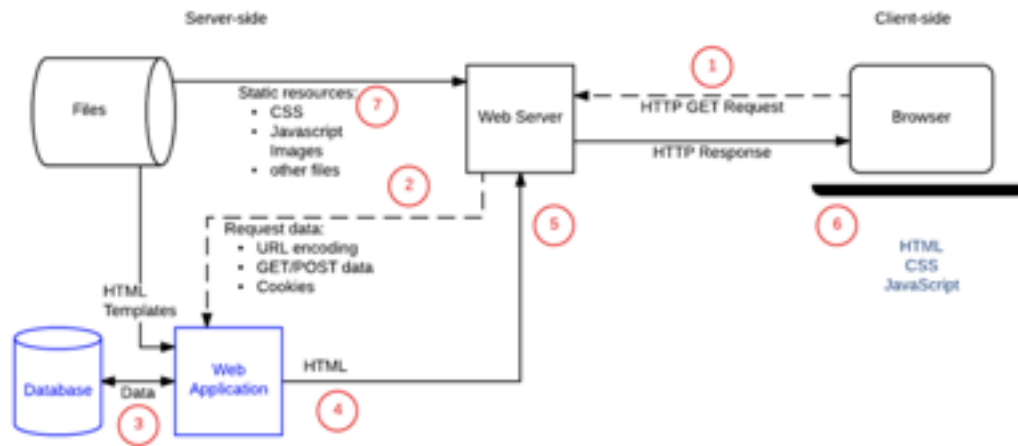


Fig -1: Client Side- Server Side Request and Response Cycle[2]

3. ANGULAR FUNDAMENTALS

3.1 History of Angular

To overcome the deficiencies of AngularJS, the core team at Google decided to move from the Model View Controller (MVC) architecture used in AngularJS to an architecture which comprises of components. A component is basically a directive with a template. A component comprises of template file(HTML), style file(CSS) and a typescript file.

Angular is a typescript based framework. Typescript is basically a superset of the ECMAScript6 (ES6) which is a scripting language specification standard. In other words, Typescript is a superset of JavaScript. This, in turn, enables typescript to have all the features of JavaScript (ES5) along with the latest features provided by ES6 such as lambda operators and iterators. Another major change that was implemented in the newer Angular framework was the inclusion of the command line interface more commonly known as CLI. This was a revolutionary innovation which made the life of front-end programmers a lot easier.

3.2 Angular Features

1. Data Binding

What is Data Binding?

In computer science, the term data binding generally refers to the technique of appending/binding data from various sources and connecting it to the web application such that it is accessible or displayed to the user.

In Angular there are various types of data binding:

- **Interpolation:** This achieved by encapsulating a variable within curly braces (i.e) {{variable name}}. This can be placed within an HTML tag, and the required data will be rendered. Interpolation generally works well when one wants to add dynamic values
- **Property Binding:** Property binding works only in one way (i.e) one way binding, from the component to the Document Object Model (DOM). This means that if the data is changed in the component, the changes made will be reflected in the DOM. Property binding binds values to the DOM properties of various HTML elements.
- **Attribute Binding:** This is a type of Property Binding in which the data is bound to attributes of various HTML tags. For

example the "src" tag is an attribute of the "img" tag in HTML. It must be noted that most HTML attributes have a one is to one upping with the DOM. However, a few exceptions are there like the cols attribute. In such cases one needs to prefix the HTML attribute name with "attr.", therefore it becomes "attr.cols".

- **Event Binding:** This is used to handle events which are raised from the DOM. These could include keystrokes, mouse movements, clicks and so on. Table 1 [3] gives you the most important event binding functions.

Table -1: List of Common Event Binding Elements

EVENT NAME	FUNCTION
(click)	Performs action on button click
(keyup)	Carried out when key is released
(keydown)	Carried out when key is pressed down
(mouseup)	Carried out when mouse button is released
(mousedown)	Carried out when mouse button is pressed down
(blur)	Carried out when an element loses focus
(focus)	Carried out when an element is in focus

There are other types of binding such as style binding, class binding and many more. These vast number of options provided by angular makes it easy to bind our data hence making it very desirable amongst various software developers across the world.

2. Template Variables

Template variables are prefixed with a # and are defined within an HTML tag. For example: `<input #email>`, #email is a template variable. The main advantage of template variables is that it makes readability of the code easier. One does not need to use the \$event tag as a function parameter, instead the template variable can be passed.

3. Two Way Binding

Two way binding basically allows dynamic changes in the User Interface to be mapped to the underlying model in the backend. Two way binding is carried out with the following syntax : `[(ngModel)].[7]`

4. Pipes

Angular offers the programmer both the option of built in pipes such as, uppercase, lowercase, decimal and currency, and also custom pipes which the programmer can code according to his/her requirement.

5. Building Re-Usable Components

This is one of the core features of Angular, and it is this feature which makes Angular so popular in the industry today. Reusability of code is one of the main features which programmers desire and Angular provides this. By using @Input, @Output properties and the EventEmitter() [4] function it becomes easy to make a component reusable.

6. Angular Directives

Angular provides various directives with which we can modify and manipulate the DOM with ease.

- Attribute Directives: These are used to modify the various HTML elements [4] in the DOM layout. Common Attribute directives include NgStyle and NgClass.
- Structural Directives: These directives are used to add or remove a DOM element from the DOM. Various structural directives include *ngIf, *ngFor and *ngSwitch.[5]

Along with these directives, Angular developers have given complete control to the programmer by enabling them to create custom directives according to their requirement.

7. Angular CLI

The Angular CLI (Command Line Interface) allows programmers to serve their application, generate the components and services, add all dependencies and install open source libraries with ease. Table 2 shows the common CLI commands used frequently while making an application with Angular

Table -2: List of Common Angular CLI Commands

COMMAND	FUNCTION
ng serve	Serves the application on localhost/4200 (default). By using --port one can select the port on which the application should run.
ng g c component_name	Generates a component
ng g s service_name	Generates a service
ng build	Builds the application
ng build --prod	Builds the application in production mode
ng new application_name	Creates a new Angular application

8. HttpClient

The Angular team had decided to name the new version after Angular 2 as Angular 4 to avoid confusion that occurred due to the different numbering of various routing packages. In the release of Angular version 4.3, the Angular team released a new library called HttpClient for handling HTTP requests like GET, POST and PUT. HttpClient is a smaller, more functional and powerful library which can handle complex HTTP Requests.

9. Routing

Most browsers are familiar with application navigations. This basically means a web page will be redirected if the URL is changed and navigate to a new page if a user action is carried out [4]. The Angular router module also follows the same concept. For the routing to work perfectly one has to configure the routes in the imports array of the "app.module.ts" file. By using the RouterModule.forRoot() method one can configure the routes. The for root method takes an array of route objects that define the navigation path for applications. The general syntax is given below :

```
RouterModule.forRoot([  
  {path : 'path name', component : component name},  
  {path : 'path name', component : component name},....])
```

After configuring the routes, add the <router-outlet> tag to the "app.component.html" file and make use of the routerLink directives in the HTML tags. The routerLink directive gives the router the control over the various HTML events in which they are present. In other words, routerLink enables navigation upon a user action such as a keystroke or a button click.

In Angular version 4.3 new router life cycle events for guards and resolvers were created.

Other important features which are provided by Angular are:

- Observables [4] and RxJS: Observables are used to pass messages between various parts of your application. An observable is often used for event handling and asynchronous programming. A subscribe() method is always called while using an Observable.
- Dependency Injection: To keep the "component.ts" file clean and independent from live HTTP endpoints during testing. This is because it becomes harder to execute the unit tests. To overcome this, programmers follow a practice in which, the component file should only contain the presentation logic while all other logic such as interacting with HTTP methods and so on, will be written in a service class. Now the question arises, how do we link the service to the component. The answer is Dependency Injection (DI). This feature provided by Angular makes it easier for programmers to separate the implementation logic and the other logic such as retrieving, handling and manipulating data. This, in turn, makes unit testing of the Angular application a lot easier and quicker.
- Animations: The @angular/animations [4] module enables programmers to insert animations into their applications.
- Support of open source libraries such as Angular Material and AgGrid are also provided. This enables programmers to obtain a quick, responsive and attractive user interface with various features.

4. TESTING IN ANGULAR

Unit Testing [6] [8] is one of the key aspects of any software development project. Building Angular applications is no other exception.

Some of the advantages of unit testing are:

- Improves the design of implementations.
- Improves quality of code.
- Enables programmers to add new features without destroying the pre-existing ones.
- Eases the debugging process.
- Reduces the cost of the project.

Due to the importance of Unit testing as mentioned above, the Angular developers decided to provide programmers with testing tools such as karma and jasmine so that testing could be carried out with ease. Jasmine is the framework on which the tests are created, and Karma is the task runner for the tests. By running the command "ng test" the Angular CLI will build the application in watch mode [4] and the Karma test launcher is then subsequently launched.

By having an in depth knowledge of testing tools programmers can easily unit test their code as Angular provides all the testing facilities that are desired by the programmer.

5. ADVANTAGES OF ANGULAR

- Can create and progressive web applications with high performance and minimal installations
- Can create native web applications.
- Provides the ability to create cross platform applications across various platforms such as Mac OSX, Windows and Linux.
- Code generation is very quick, and Angular provides Ahead of time (AOT) compilation.
- One of the major advantages of Angular is that it is Universal. The code can be served on various server side scripting platforms such as Node.js, .NET and PHP.

- With the introduction of the Router Module Angular splits the code segments with relative ease and efficiency thereby improving the performance.
- Productivity is also one of the key advantages of Angular. Appealing and responsive templates can be created using various open source libraries such as Angular Material and AgGrid. The Angular command line interface also makes life easier for programmers due to the various features offered by it. Finally, Angular is supported by various Integrated Development Environments (IDE's) thereby reducing the burden on the programmer.
- Testing, Animations and accessibility are all provided to the programmer hence ensuring that complex projects can also be carried out by Angular.[4]

6. FUTURE SCOPE

With the future of web applications changing rapidly it is important for the Angular team at Google to keep updating the application to the modern needs and wants of programmers and clients. With the increasing popularity of the web framework called React which is created by Facebook, Angular faces stiff competition and they must be always up to date otherwise they risk themselves losing a lot users. The latest version of Angular is Angular 9 and it is called as Project Ivy. Project Ivy was released recently and had many upgraded features to the older stable versions of Angular such as Angular 6. Angular 9 has a faster build speed, the latest version of typescript (i.e) TypeScript 3.8, provides end to end testing which can now support grep and invertGrep. Finally there is an improved HTML and expression syntax highlighting if you use Visual Studio Code as your IDE. The Angular team is planning to expand on these features and build a more robust platform in the future release of Angular. The future release will be the 10th version of Angular and it is in the pipelines to be released soon.

7. CONCLUSION

Angular was created to overcome the difficulties that were being faced by AngularJS. The whole architecture was changed from the MVC architecture to a hierarchy based component architecture. Angular is ideal for single page applications, however, other types of applications can be created by Angular due to its vast features and cross-platform capabilities.

In this paper, a thorough overview of the origin and history of Angular is given along with a detailed explanation of the key features which makes Angular so unique and widely used in the industry. Finally, an insight to testing is given and the paper is wound up with the advantages of Angular and the future scope of Angular.

ACKNOWLEDGEMENT

I would like to extend my gratitude towards my guide Mr. Manjunath A.E for his continuous support and guidance on my work on this project and paper. I would also like to thank the Department of Computer Science and Engineering at RVCE for giving me the chance to write this research paper on my final year project.

REFERENCES

- [1] Madhuri A. Jadhav, Balkrishna R. Sawant, Anushree Deshmukh, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (3) , 2015, Single Page Applications using AngularJS.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.
- [2] "Mozilla Developer Network (MDN) Docs",https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Client-Server_overview.
- [3] <https://www.positronx.io/useful-list-of-angular-7-event-types-for-event-binding/>.

- [4] "Angular Documentation", <https://angular.io/>.
- [5] Sneha Ambulkar, Pushpanjali M.Chouragade, International Journal of Scientific & Engineering Research, Volume 7, Issue 2, February-2016, AngularJS.
- [6] Mohamed Sultan, Future Technologies Conference (FTC) 2017 29-30 November 2017 | Vancouver, Canada, Angular and the Trending Frameworks of Mobile and Web-based Platform Technologies: A Comparative Analysis
- [7] Miguel Ramos, Marco Tulio Valente and Ricardo Terra, AngularJS Performance A Survey Study.
- [8] Miguel Ramos, Marco Tulio Valente, Ricardo Terra and Gustavo Santos, AngularJS in the Wild: A Survey with 460 Developers.