Web Applications using Angular with REST API

Damini Jaiswal[1], Prof. Lalitha V P[2]

[1]Department of Computer Science and Engineering, R.V. College of Engineering, Bengaluru, Karnataka, India
[2]Professor, Department of Computer Science and Engineering, R.V. College of Engineering, Bengaluru, Karnataka, India

---***---

Abstract - *A Web application or Web app is an application that is distributed via the Internet using an interface like browsers from a remote server. Many tools that can make web development easier and web application framework is one of them. Angular is one such widely used framework which helps to develop single page web applications across all platforms and provides speed, performance and incredible tooling. RESTful web services are lightweight and scalable services. For the single-page applications bring the data back into the net application using these essential services. The built-in HTTP Client services of Angular binds the major functions for retrieving the data from server where REST service is hosted. The paper mainly focuses on integrating angular with RESTful web services and the related basic architecture, it also discusses various versions of angular and its impact on business values and development.*

Key Words: Framework, Application, RESTful web services, Angular, HTTPClient

1. INTRODUCTION

There is a rapid evolution in web development since the last two decades. Web applications are designed for a wide range of purposes and use-cases, are user-friendly and have innumerable end users from an multi-national organization to an individual. The regular evolution of the technology has led to increased use of web. Web is used as an economical means of communication and exchanging resources by almost all businesses and millions of industries. Web applications have disrupted the world by providing wide range of possibilities. Businesses make use of web applications to become more streamlined and straightforward, can widen their reach and achieve their goals and grow faster. Numerous clients can be targeted simultaneously over the internet via these applications. Customers today want it all which leads to problems for the developers. There can be issues in the development phase such as changing requirements for business reasons and in creating single page application in a very clean and maintainable way.

Using a good web application framework helps speed up the development phase with much ease and angular is one such framework. Numerous new mobile and web-based application frameworks have been released and adopted in both communities of software development industry and research[1]. Most of the frameworks include an array of libraries and responsive user interactions, often referred to as 'single-page applications'[4]. A Web server, an application server and a database are the basic requirements for a web application to work. The requests coming from a client is managed by the web servers, while the requested tasks are taken care and completed by the application server. Built on the REST architecture, RESTful web services provide maintainable, lightweight, and scalable services.

2. BASIC STEPS

For web development, Google's angular framework is one of the most prominent one. It uses typescript language. Some of its strongest features include routing, dependency injection, model-binding, and the capability to create custom reusable components (formally known as 'directives'). Angular provides the Tree Shaking feature which is removing the unused piece of codes or libraries in the project which saves memory and increases the project performance as well [5], [6].

HTML is responsible for structuring templates and is styled with CSS. If using higher versions, we can also integrate bootstrap css styles with the angular project using angular CLI.

Step1: To make sure you're using the latest version of NodeJS, npm and the CLI itself.

- Updating nodeJS-Latest version can be downloaded from nodejs.org but first uninstall the installed versions.

- Updating the npm by running: npm install -g npm

- Updating the CLI by running: npm install -g @angular/cli

Step2: Now go to the location where project has to be saved then run ng new app-name. Now depending on the wording you're using you might be asked two questions for which hit enter to choose the default. Now the new folder will be created.

Step3: run ng serve that will build and serve the app, rebuilding on file changes. Now the server runs on the locally hosted 4200 and the app can be seen by going to the browser and entering localhost:4200.

There will be a default app module which has an NgModule that describes how the application parts fit together. Every application has at least one angular module, the root module, which must be present for bootstrapping the application on launch[2].

3. ANGULAR CONNECTION TO DATABASE

This section discusses how angular can be connected to database through a server. Angular 2+ is a client-side JavaScript framework. 'Angular 2+' is commonly referred to as Angular. Angular is a revolutionary version and a complete rewrite of AngularJS 1.x frameworks. Being much less complicated than Angular1, Angular uses ES2015 and ES2016 in addition to 'Typescript' as a compiler, which supports classes and module loaders[1]. It only runs and executes code in a web browser and does not execute code on a database server.

Whatever you have in your application and you want to store that data in a database and also fetch it from there when your app restarts, when a user revisits your page and so on and it doesn't matter which kind of database we're talking about, if that's an SQL or NoSQL database, doesn't really matter. The thing is you don't connect Angular to a database directly, that means you don't enter your database credentials into your Angular app or anything like that, this would be highly insecure because everyone can read your angular code since it's a front-end javascript application. So what happens is that HTTP requests are sent and you get HTTP responses to and from a server as shown in fig.1. A server in the end is defined as an API here, that is the REST API web server. REST, which stands for "Representational State Transfer" is an architectural style specifying its constraints. Desirable properties such as scalability, modifiability, and performance are provided when applying these constraints to web services.

To identify and analyze the targets of the interaction with ones clients, a set of resources are presented by the RESTful web service. URIs are responsible for identifying and determining these resources. The client is responsible for holding the session state, and all the information necessary to service the request is contained by the client requests. For maintaining a persistent state for a period and allowing authentication, the session state can be transferred by the server to another service such as a database. When it is ready to make the transition, the client begins sending requests to a new state. The client is considered to be in transition, while one or more requests are outstanding.[3] After visiting this API's URLs we get back data which in JSON format. So now we can write code to store, fetch, upload files to the database through this server.

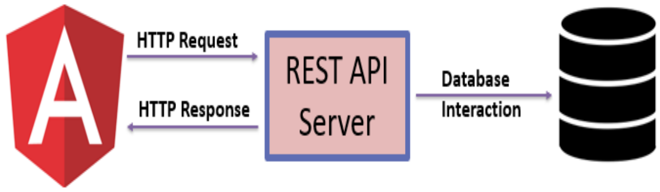


Fig -1: Database Connection through REST API Server

HTTP request is made up some parts as shown in fig.2 like the http verb which can be:

- GET — retrieves data, like a blog article.
- POST — creates data, like a new blog article.
- PUT — replaces data, like an existing blog article.
- DELETE — deletes data, like an existing blog article.

The most important part about a request of course is the URL you are sending the request to. Then there is another part which is headers and it includes the metadata. The last part is the body which contains the main data that is attached to the request.

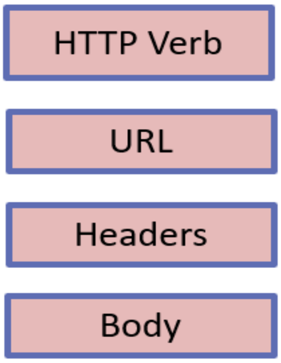


Fig -2: Anatomy of HTTP request

Steps to setup HttpClient module:

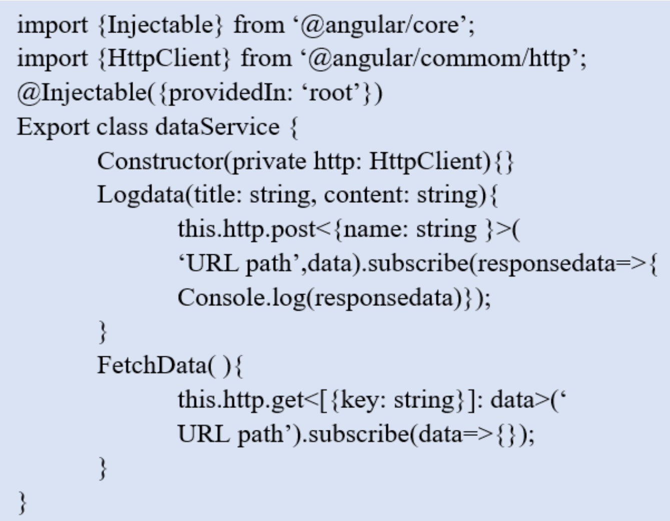
In the app module, a new core Angular module which is HttpClient Module has to be added. By simply importing HTTPClient module from @angular/common/http where package name is @angular/common/http, this can be achieved.

```
Import { HttpClientModule } from '@angular/common/http'
```

Now HttpClient has to be also added in the imports array as:

```
    imports: [
     BrowserModule,
     HttpClientModule
],
```

For logging data http requests are sent with the post method and we send then only when we subscribe to them. For outputting the data to the page that is for fetching the data we can use get method and send a get request to that same URL we used for storing the data. It is best to make services to deal with all the data retrieval parts. An example of such a service is:



```
import {Injectable} from '@angular/core';
import {HttpClient} from '@angular/commom/http';
@Injectable({providedIn: 'root'})
Export class dataService {
        Constructor(private http: HttpClient){}
        Logdata(title: string, content: string){
                this.http.post<{name: string }>(
                'URL path',data).subscribe(responsedata=>{
                Console.log(responsedata)});
        }
        FetchData( ){
                this.http.get<[{key: string}]: data>('
                URL path').subscribe(data=>{});
        }
}
```

Similarly, for deleting data first the URL which targets the data node has to be grabbed and then request has to be sent there. Now if there is a need to be informed about that deletion process in the component, we can return a observable here but the subscription will not be here instead it will be in the component. In this way the client is able to access database through a server in place of accessing it directly.

4. COMPARISION OF VARIOUS VERSIONS

To make the front-end development procedure easier to manage Google launched a superior framework in 2009 which was AngularJS. Though there were countless plugins and frameworks already there in the marketplaces, website owners chose to pick AngularJS development services because of its advanced features. Some of those features include the MVC framework which is a software design pattern to web applications development. MVC architecture helps in building separate client side applications. It has many other important features like data binding, templates, unit testing(helps to build a quality code), dependency injection, filters, handling of the directives. There is continuous updating of model and the view layers in order to maintain the sync between one another. It provides the feature to switch between views while maintaining the single page application. AngularJS has found its use in various fields like weather applications, e-commerce websites, any application that streams videos, and etc. AngularJS utilizes HTML as a template language. It extends HTML attributes with directives and binds data to HTML with expressions. AngularJS creates an environment that is extraordinarily expressive, readable, and quick to develop.

Angular is a revolutionary version and a complete rewrite of AngularJS 1.x frameworks. Angular is a term that refers to all angular 2+ versions. $scope and controllers were replaced by directives and components in angular. There were differences in CLI, performance, architecture, dependency injection, use of typescript and many more. Angular provides with a better structure to maintain and create bigger applications. There are not much major differences between angular 4 and angular 2. In angular 2 types of directives were introduced, one was attribute directives and the other was structural directives. Microsoft's TypeScript language is used by angular which is a superset of ECMAScript 6 (ES6). This helped in combining the benefits of ES6, like iterators and lambdas with the benefits and advantages of typeScript features, for example type declarations.

Angular 5 was released in 2017. This version had many useful updates which helped in removing unnecessary code, it allowed sharing the state of the application between the server side and client side very easily and also had compiler improvements and improved support for decorators.

Angular 6 was released in April 2018. It reduces bundle sizes for common use. It has an optional backward compatible generic type that supports typed. This release is focused less on the underlying framework, and more on tool-chain and on making it easier to move quickly with angular in the future. It helps to eliminate them if not required in the application. It uses a new method of connecting modules and services.

Angular 7 was released in October 2018. It provides a new ability to recover from malformed URLs and supports typescript 3.1. For starting and running Angular projects on your local machine, it also includes the downloadable console and added new interfaces and pipe like doBootstrap, canLoad interface and keyvaluepipe. It was a major release for the whole platform including CLI and the main framework.

Angular 8 was released in April 2019. It updated support for typescript 3.4. It includes many performance and workflow improvements. It added support for firebase and dynamic imports for lazy loaded modules. It also added ng new command and has improvement for ng upgrade. Ivy engine was a major part of this release, this was the new compiler for this version.

Angular 9 had Ivy compiler as the default for all applications and updated to typescript 3.7. It provides improved testing, better debugging and improved bundle size. It updated the feature of showing compiler error when a template variable is created that has not been defined in a component.

5. CONCLUSION

This paper presents a general study on building efficient single page web applications using angular web application framework for the frontend. It is most advantageous framework for building dynamic programming structure and makes it possible to synchronize data extremely efficiently, automatically. It gives a complete framework with elegant architecture and extensive documentation that allows developers to find all the necessary information quickly. The database interaction can take place through a REST API server and then HttpClient can be used to include observable APIs, typed request and response objects, testability features, and streamlined error handling.

REFERENCES

[1] Angular and the Trending Frameworks of Mobile and Web-based Platform Technologies A Comparative Analysis by Mohamed Sultan: Future Technologies Conference (FTC) 2017 29-30 November 2017| Vancouver, Canada

[2] https://angular.io/docs

[3] "Fielding talks about application states" Tech.groups.yahoo.com. Retrieved 2013-02-07.

[4] K. N. Ard, "Single page architecture as basis for web a pplications," 2015.

[5] S. A. Mousavi, "Maintainability Evaluation of Single Page Application," Ph.D. dissertation, 2016.

[6] P. Deeleman, Learning Angular 2, 2016.