Single Page Reactive Application using Angular Spring MVC and Rest API

Shashank Jain, Sudha R Karbari

Abstract: Single Page Application (SPA) in web technology has the potential to compete with the seamless performance of app. Single Page Applications are highly reactive as they load the JavaScript bundles just once and then according to the required DOM changes the elements are downloaded. So, it is very fast and consumes less resources. In this paper Angular and Spring MVC are discussed to provide a methodology for creating SPA.

Keywords: Single Page Application (SPA), Angular 8, Spring MVC, REST API

I. INTRODUCTION

TILL the start of 20th century, most of the websites on the internet only had static data, text and images. In order to get current information, we had to either mail or contact them over phone. After this the websites were developed with dynamic data and were very successful. But with introduction of mobiles and the seamless performance on different app. So, web market started facing tough competition and, in this regard, they came up with Single Page Application (SPA) technique.SPA (Single Page Application) is a latest technology in web development in which loading of HTML page happens only once. After this, whenever user clicks a button for other information then only required resources are loaded which needs to be changed for that request, and the route is change in URL. In SPA the Javascript bundles are loaded only once and when a request comes from frontend then only the part that changes in DOM are loaded. This makes SPA very fast and also the consumption of resources are very optimized. Angular is a Typescript framework which gets type-compiled into Javascript because browser only understands Javascript. SPA is mainly achieved by using two main techniques in Angular that is, data binding and routing.

II. FRAMEWORK

A framework is a software bundle which has many inbuilt packages and tools that can be directly used by the software developer. The use of framework ensures that, a person as software engineer need not waste time on writing boiler plate code. This code comes in-built in frameworks and a person can focus on developing core business logic. Angular and Spring MVC are two popular frameworks used for frontend and backend respectively.

A. Angular 8

Angular 8 is an open source frontend framework for developing Single Page Applications (SPA). It is a package of languages like HTML, CSS and Typescript. There are many features like data binding, routing, components, services, dependency injection, etc. which makes Angular very popular among recent technologies. Fig. 1 shows a comparative analysis of the popularity of different Javascript frameworks based on the number of stack overflow question related to that framework.
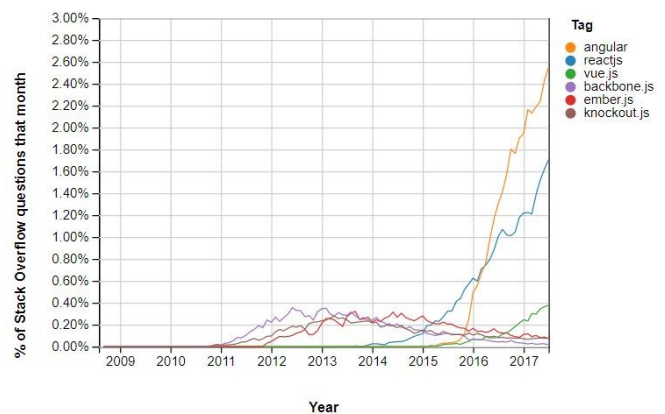

Fig. 1 Comparison of Different Javascript Frameworks [4]

B. Spring MVC (Java Framework)

Spring MVC is a Java based framework which uses the MVC design pattern. MVC stands for Model, View and Controller. MVC architecture is used to for proper segregation of principle. Model is used for defining POJO (Plain Old Java Object) class. In this, we define Java classes based on our database design. View is responsible for data presentation and interaction with the user. Controller is used to route the request coming from frontend to proper service and fetch the required results from database. Spring MVC has two core features Dependency Injection and Inversion of Control.

III. METHODOLOGY

The development of website is not just mere coding. It requires proper action plan and thought process. Coding is just 15% of the overall process. So, for the initial setup process, we need to design wireframes.

Retrieval Number: E9494069520/2020©BEIESP
DOI: 10.35940/ijeat.E9494.069520
Journal Website: www.ijeat.org

478

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication
© Copyright: All rights reserved.

Single Page Reactive Application using Angular Spring MVC and Rest API

Wireframe is a tool which is used to design how the website will look. We can have layout design and proper size for each and every element present on that page. Designing of wireframe also helps us in understanding the user perspective, and this helps us in proper designing of database. A sample wireframe is depicted in Fig. 2.
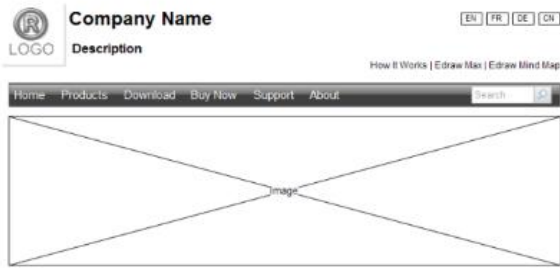

Fig. 2. A wireframe

Once we design the wireframes of all the possible screens, then we can accordingly design the database. The database should be normalized to 3NF in order to avoid redundancy and have atomicity of data. This concludes the planning phase and we move to the implementation phase.

First of all, frontend is designed in Angular 8. The concept of SPA (Single Page Application) is utilized in angular by using two main properties i.e. data binding and routing. Data binding is one of the core features of Angular. It helps in directly using the data from Typescript file and render the content in HTML or vice-versa. There is also two way binding available which can be used for form filling activities. Four types of data binding techniques are shown in Fig. 3.
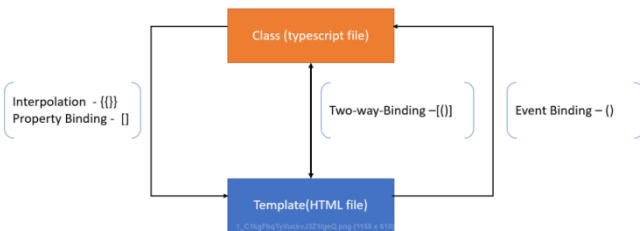

Fig. 3. Databinding in Angular [3]

Routing is the property which is used by angular to navigate from one page to another. Using navigation, there is only one URL and the rest pages are opened by using "/" and appropriate name as configured. Request parameters and query parameters can also be passed using routing. This is the main feature that make the web applications seamless as mobile apps.

The requests made at frontend (User Interface) are passed to the backend by using REST API HTTP methods namely Post, Get, Put, Delete, Patch, etc. The Post method is used to send data, Get method is used to retrieve data, Put method is used to update data and Delete is used to delete data.

When the request reached from frontend to backend, then job of backend is to properly manipulate database and backend code should be properly structured. For this, we should use Spring MVC framework which is especially designed Java framework for web applications. In Spring MVC we have three layers, Controller, Service and DAO. Controller is the

layer in which we make API endpoints to handle request coming from frontend. Service is the layer in which we write most of the core business logic that is manipulating the requests from frontend sending appropriate responses, etc. DAO (Data Access Object) is the layer which is majorly responsible for interacting with the database.

The flow now shifts from backend to database that we have designed with appropriate normalization. Now CRUD (Create, Read, Update, Delete) operations are performed according to the request coming from all the way through frontend. A schematic for the overall methodology is shown in Fig. 4.
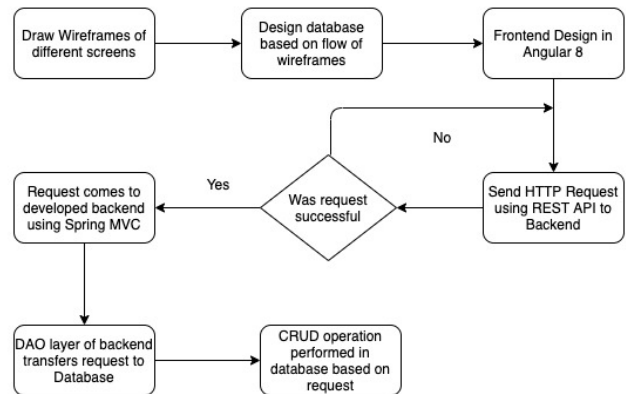

Fig. 4 Schematic flow for Web Development

IV. COCLUSION

Single Page Application (SPA) are the way ahead in the field of web development and Angular will be one of the vital Typescript framework in this regard. The main thing in web development is not only coding but proper planning and deciding the best suited technologies according to our requirement.

REFERENCES

1. V. Okanovic, "Web application development with component frameworks," 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, 2014, pp. 889-892.
2. W. Chansuwath and T. Senivongse, "A model-driven development of web applications using AngularJS framework," 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, 2016, pp. 1-6.
3. Omerbegovic, N., Omerbegovic, N., & Subasi, A. (2014). Effects of Two-way Data Binding on Better User Experience and Easier Development of Clinical Information Systems. SWAT4LS.
4. Kaluža, Marin & Vukelic, Bernard. (2018). Comparison of front-end frameworks for web applications development. Zbornik Veleučilišta u Rijeci. 6. 261-282. 10.31784/zvr.6.1.19.
5. S. M. Sohan, F. Maurer, C. Anslow and M. P. Robillard, "A study of the effectiveness of usage examples in REST API documentation," 2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Raleigh, NC, 2017, pp. 53-61
6. Haupt, Florian & Leymann, Frank & Pautasso, Cesare. (2015). A conversation based approach for modeling REST APIs. 165-174. 10.1109/WICSA.2015.20.
7. K. Boonchuay, Y. Intasorn and K. Rattanaopas, "Design and implementation a REST API for association rule mining," 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Phuket, 2017, pp. 668-671.

8. H. Rissanen, T. Mecklin and M. Opsenica, "Design and Implementation of a RESTful IMS API," 2010 6th International Conference on Wireless and Mobile Communications, Valencia, 2010, pp. 86-91.
9. F. Yang and S. Manoharan, "A security analysis of the OAuth protocol," 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), Victoria, BC, 2013, pp. 271-276.

AUTHORS PROFILE

Shashank Jain is currently pursuing his Bachelor of Engineering Degree as a student of 8th Semester from Electronics and Communication Department, R.V. College of Engineering, Bengaluru. He is also working as an Intern Software Developer Engineer with Accolite Software India Pvt. Ltd, Bengaluru since January 2020. His 1 paper is under publication with a National Conference. He has worked on some of the technologies and frameworks listed – Spring MVC, Angular 8, Flutter. His areas of interest include Software Development and Web Designing.

Sudha R Karbari received B.E degree and MTech with MEMS as specialization honors from Visveswaraya Technological University, Belgaum, Karnataka in 2009 and 2013 respectively. Her research interests include MEMS and sensor development. She has a work experience of 1.4 years for an IT firm and teaching experience of 6 years in different institutions as ASSISTANT PROFESSOR Currently She is working as an Assistant professor in R V College of Engineering, Electronics and Communication Department, Bangalore, Karnataka, India.