ISSN:0975-9646

Single Page Application using AngularJS

Madhuri A. Jadhav[#1], Balkrishna R. Sawant[#2], Anushree Deshmukh[*3]

[#]*Master of Computer Application*
Lokmanya Tilak College of Engineering
Koparkhairane, Navi Mumbai

[*]*Department of Computer Engineering*
Lokmanya Tilak College of Engineering
Koparkhairane, Navi Mumbai

Abstract— **Single Page Application (SPA) is composed of individual component that can be replaced or updated independently, without refreshing whole page so that the entire page does not need to be reloaded on each user action, which saves bandwidth as well as no loading of external files every time when page is loaded, such as images or CSS files, etc. The purpose behind this is to make the subsequent page loads very fast as compared to traditional Request-Response cycle. SPA's written by using JavaScript, HTML5, AJAX are getting the likes of developers to build their web applications and some frameworks like AngularJS which are built on top of the JavaScript are making the life of developers very easy. The idea behind using AngularJS in web application is to make your web application modular and easy to maintain. AngularJS brings MVC (Model View Controller) capability to your application. After using minified and compressed files in your application, the size reduces to some KBs which will results in faster loading of pages.**

Keywords— **Single Page Application, AngularJS, Web Application.**

I. Introduction

The traditional websites which were introduced in early days of Internet had the sheer purpose of serving static pages (content) to its client, which included content like images, CSS, JavaScript, etc. as the years went people started using websites to publish their business and explore it to world, by the introduction of e-commerce, the need of providing current status of business did rise and that requirement lead to serving the dynamic pages which were the mirror image of live status of business. Technologies like Asynchronous JavaScript and XML (AJAX) were used to dynamically load the data from database. AJAX helped to introduce dynamic content to web pages, However, using AJAX we can built desktop applications, So to achieve same user experience on mobile applications, Single Page Application have been developed. We can build Single Page Application with JavaScript too, but development and maintenance of front end using JavaScript is more complex. Frontends with AngularJS is easy to develop and maintain, since AngularJS frontend is broken down into 3 components Model, View and Controller (MVC).

In the traditional approach, when user dose any interaction with a page, like button click then new page is loaded and generated from scratch to display a response on user's window. This requires more bandwidth as well as time.

In modern era of web technology most of the websites are using SPA which is a web application that sits on one single page like any other desktop application. In SPA all the component like CSS, images, scripts and any other required resources are loaded at one time at the initial page load and then appropriate content/components gets loaded dynamically depending on the user interaction. Once the user has loaded an initial slim version, after that every subsequent request will take very less amount of time because it is now refreshing that particular part or region of a page rather than reloading an entire page. The control remains on one page till the user is on that website and that single page communicate with the server behind the curtain.

II. Single Page Application (SPA)

The SPA is a web application that fully loads all of the resources in initial request and then the page components are replaced by other component depending on user interaction. When we compare SPA with traditional web application, we can see there exist an analogy between 'states' of SPA and 'web pages', navigation of 'web pages' in traditional web application is analogous to 'state' navigation in SPA.

A. Definition

"Single Page Application (SPA) is composed of individual components that can be replaced/updated independently, without refreshing/reloading whole page so that the entire page needs not to be reloaded on every user action."

1) Individual Components: Entire page is divided into smaller components that interact with each other.

2) Replaced/Updated: A component/region/part of a page replaced/updated with any changes on user requests.

3) *Refreshing/Reloading:* The entire page never reloaded/refreshed, rather new content is loaded in some part or section as per new data.

4) User actions: Single Page Application is responsible for handling all user inter actions such as button click, input from keyboard etc. very fast and hence leads to very fluid user interface.

SPA allows more flexible and elegant way of dealing with data. Refreshing particular part or a section of a page without refreshing an entire page is main goal that SPA is serving, but all this flexibility requires more interactive interface and this leads to better user experience.

When we build SPA with JavaScript front end becomes very complex. In bigger projects many developers work collaboratively to build front end. If the code is very difficult to understand because of no separation of layouts and business logic then maintenance of that code becomes very difficult. This issue of maintenance of code of bigger projects is solved if that Single Page Application is build with AngularJS. Since AngularJS divides the front end into 3 parts.

- Model
- View
- Controller

Code that is easy to read and understand becomes a key to achieve high efficiency and good quality as well as leads to fluid user interface.

B. *Client-Server Request-Response Cycle*

In a traditional approach,

- Client makes initial request to open a particular website or application.
- Server respond with HTML pages along with images, CSS, script files, and other external resources.
- When client does interactions with a page like button click or input from keyboard, it makes a new request to server.
- Server again responds with a whole HTML page not a particular part of a page.
- At the client side that page is reloaded and hence requires more time for these Client-Server interactions. This leads to more bandwidth and unsatisfied user experience.

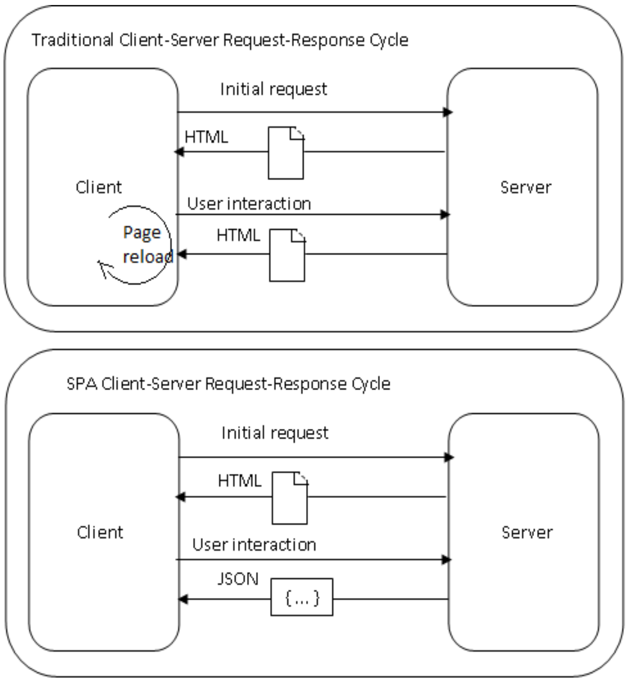

Fig. 1 Client-Server Request-Response Cycle

In SPA AngularJS,

- Client makes initial request to open a particular website or application.
- Server respond with HTML pages along with images, CSS, script files, and other external resources.
- Now when client does any interaction like button click on input from keyboard with a page, it makes a new request to a server.
- Server now responds with only user's action result not the entire HTML page again. That is server responds with a JSON (JavaScript Object Notation) which includes only result of user's action.
- At client side that page is not reloaded rather a particular part of a page is replaced/updated with that response. Hence very less bandwidth as well as time is required for this Client-Server Request-Response Cycle. This leads to very fluid user interface.

C. *SEO in SPA:*

Search Engine Optimization (S.E.O.) is the process of increasing your web site's page rank so that more and more traffic can come to your web site. The content build by SPA's is dynamic and search engines have difficult time processing them this can result into poor page rank. To make your SPA S.E.O. friendly you should create HTML snapshots and regular maintenance is required. Following is the basic overview of how search engine optimization works,

- Crawler will look at some URL.
- (http://mydomain.com/myseo#key=value)
- The crawler request web server to fetch the content of that URL.
- Web server returns the HTML snapshot to crawler.
- HTML snapshot is processed by the crawler.
- Search result shows original URL.

III. ANGULARJS

AngularJS is an open source JavaScript framework maintained by Google and community which can help developers to create single page applications. Its purpose is to help developing the web applications with model-view-controller (MVC) capability in an effort to make development, maintaining and testing easier. SPA is getting popular nowadays and the technology like AngularJS aids to create such applications.

AngularJS helps to create web applications which are based on HTML, CSS, JavaScript. AngularJS brings the MVC capability to the web application and hence making it more modular and easy to develop, maintain and test. AngularJS introduces additional tags which are called as directives. These directives are prefix with 'ng-' and the goal of directive is to bind the data to the view/templates through controller. AngularJS controllers are written in JavaScript which add the business logic to views which are nothing but HTML pages.

SPA is fully loaded in initial load and only page regions/sections are replaced or updated with new page fragments loaded from server on demand. AngularJS helps to create such applications easily. AngularJS is fully client side library. AngularJS is based on full Bidirectional Data-Binding. Bidirectional-Data-Binding Is an automatic way of updating a view whenever model changes, and updating a model whenever view changes.

AngularJS is the fastest road for us to implement the simplest website as well as most complicated web applications. In AngularJS main file is index.html that functions as a base for our entire application or a website. Whatever files we required like CSS or scripts, include that file in index.html and no need to repeat that anywhere else. There is another file menu.html; our site's navigation goes here. We could change site's wide element for ex. Navigation icon, just by making changes in this file, not required to make changes anywhere else.

A. AngularJS Controllers

AngularJS Controllers are nothing more than plain JavaScript functions, which are limited to a particular scope that is for one view there is one controller, which helps in maintenance of code. Controllers are used to add business logic to your view. Views are HTML pages. These HTML pages simply show the data that we bind to them using Bidirectional data binding. Basically it is the controller's responsibility to glue the Model (data) with the View.

B. Benefits of AngularJS

1) *MVC:*
   AngularJS breaks down your application into Model (data), Views (HTML page), and Controllers (logic). This separation helps to easily develop your web application.
2) Two way binding:
   AngularJS provide a powerful technique called two way data binding which allow data to get updated whenever there is change in the Views.
3) Ease in UI development:
   Since AngularJS separates your web application into MVC pattern, web designers can now create Views separately without much of worrying about business logic behind the View.
4) Server Communication:
   AngularJS controllers communicate with server behind the scene and they are completely responsible for controlling the behavior of SPA.
5) Testing:
   AngularJS provides various modules by which testing your web application becomes easier than before.

C. Sample Program in AngularJS

home.html

```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Single Page Application in AngularJS </title>
<script
src="//ajax.googleapis.com/ajax/libs/angularjs/1.3.3/angular
.min.js"></script>
<script
src="//cdnjs.cloudflare.com/ajax/libs/angular.js/1.3.3/angul
ar-route.min.js"></script>
<script src="app.js"></script>
</head>
<body ng-app="myapp" ng-controller="myctrl">
<div ng-show="showme">
<label>Enter your Name : <input type="text"
placeholder="joe" ng-model="username"/>
</label><br/>
<label><button ng-click="submit()">Click Me</button>
</label><br/></div>
<div ng-show="!showme">
<p>Welcome to our website {{ user }}.</p>
</div>
</body>
</html>
```

app.js

```
var app=angular.module('myapp',['ngRoute']);
app.config(function($routeProvider){
        $routeProvider
        .when('/',{
        templateUrl: 'home.html'
        });
});
app.controller('myctrl',function($scope){
        $scope.showme = true;
        $scope.submit = function(){
                $scope.showme = false;
                $scope.user = $scope.username;
        }
});
```

D. Sample Program Explained

In our AngularJS sample we start our application by creating HTML page which is presentation part of our sample program. AngularJS libraries are not available to us until unless we link them by using following script tags, first and second line will bring AngularJS core libraries into our application and the last line will attach AngularJS module that we have created for our sample.

```
<script
src="//ajax.googleapis.com/ajax/libs/angularjs/1.3.3/angular
.min.js"></script>
<script
src="//cdnjs.cloudflare.com/ajax/libs/angular.js/1.3.3/angul
ar-route.min.js"></script><script src="app.js"></script>
```

We use ng-app and ng-controller directive to inject our 'myapp' module and 'myctrl' controller to our page.

```
<body ng-app="myapp" ng-controller="myctrl">
```

ng-model directive is used to bind the data between controller and views,

<input type=”text” placeholder=”joe” ng-model=”username”/>
and ng-show directive takes the Boolean value ‘true’ or ‘false’ which allows us to display and hide the content of web page.
<div ng-show=”showme”>

Expressions in AngularJS are written inside {{}} curly brasses which are evaluated on page load or some event.

<p>Welcome to our website {{ user }}.</p>

We then create our app.js file which have the module and controller for our sample program, to create module and configure it we use following line of code.

var app=angular.module('myapp',['ngRoute']);

app.config(function($routeProvider){

We also inject the ‘ngRoute’ dependency to our module inside [] square brackets. Configuring routes in AngularJS is done in following way,

```
$routeProvider
        .when('/',{
        templateUrl:'home.html'
        });
```

Controllers are the backbone of AngularJS and they are defined in following code snippet.

```
app.controller('myctrl',function($scope){
 /* business logic goes here. */ });
```

IV. Conclusion

Single Page Application is very easy to build. It loads entirely for the first time and then every subsequent request, only a part of particular page is goes on updating or changing as per server’s response on client’s interactions without refreshing the entire page. This saves bandwidth as well as time required for this Client-Server Request-Response Cycle. Single Page Application built with AngularJS structured their frontend into Model, View, Controller pattern and hence very easy to maintain a code. Code that is very easy to read and understand is a crucial part for achieving high efficiency and leads to very fluid user interface.

References

1) http://www.knowarth.com/single-page-application-angular-js/
2) https://docs.angularjs.org/api
3) http://codeforgeek.com/2014/12/single-page-web-app-angularjs/
4) .http://www.diva-portal.org/smash/get/diva2:571223/FULLTEXT01.pdf
5) .http://en.wikipedia.org/wiki/Single-page_application
6) .https://scotch.io/tutorials/single-page-apps-with-angularjs-routing-and-templating
7) https://www.airpair.com/angularjs/building-angularjs-app-tutorial
8)].http://www.codeproject.com/Articles/808213/Developing-a-Large-Scale-Application-with-a-Single
9) .http://code.tutsplus.com/tutorials/3-reasons-to-choose-angularjs-for-your-next-project--net-28457