Part III — Angular JS

Based on lectures by Brian

Notes taken by Dexter Chua

Lent 2017-2018

Contents

1 Introduction to Angular

1.1 Introduction

Angular is an open source JavaScript framework maintained by Google. We use Angular to create single page applications (SPA).

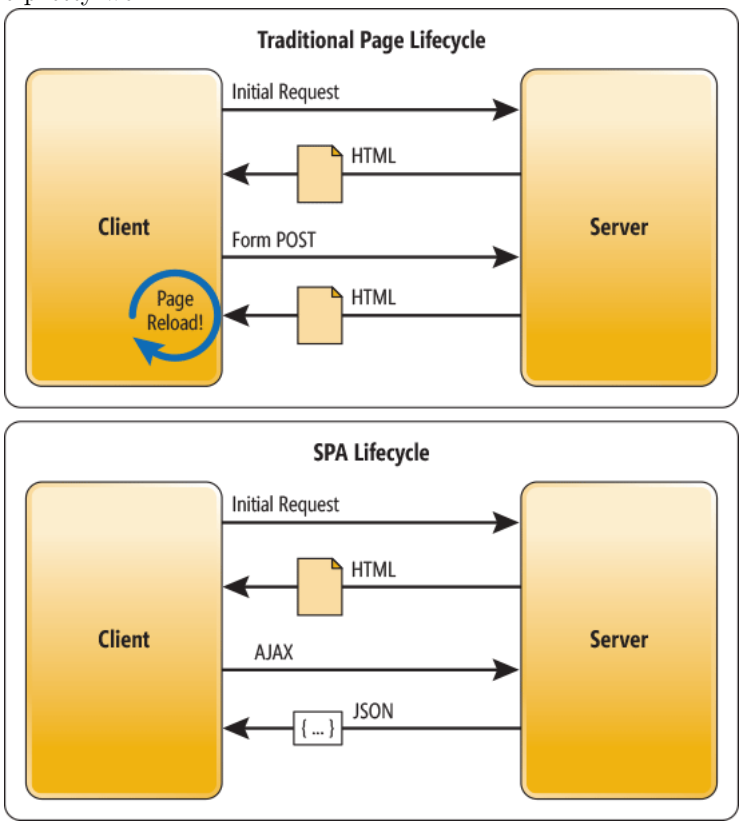
1.1.1 Single Page Application

SPA are web applications that work like a desktop application. That means upon first loading the application the user downloads all the JavaScript, HTML and CSS to render any part of the application. Any extra data needed will be fetched dynamically when the user is using the application. However a total refresh is not necessary upon most circumstances.

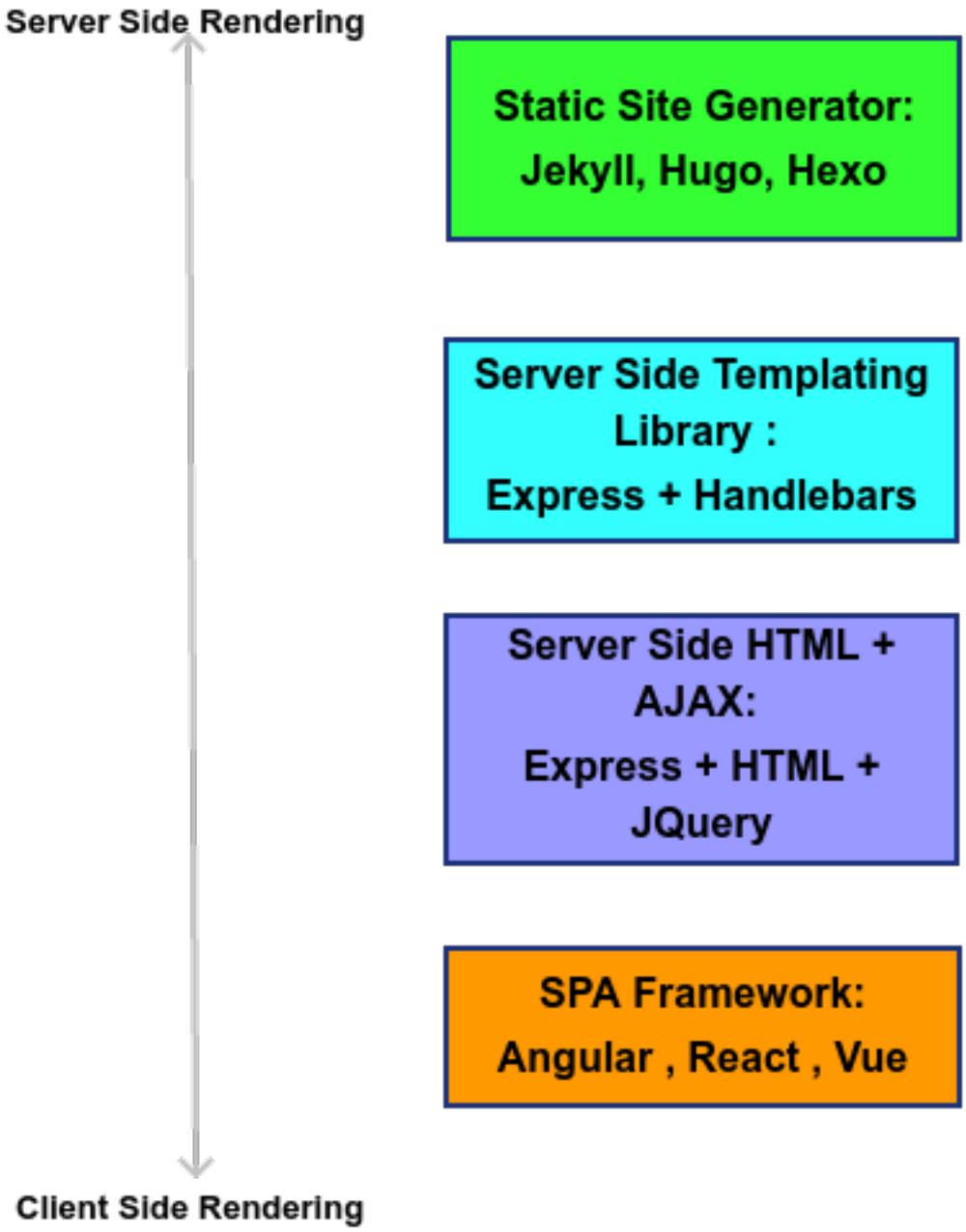
We can summarize the difference between a SPA and a normal website as follow: normal website reloads on every click. SPA loads most HTML/CSS from the beginning and loads the necessary JSON while needed.

Therefore , you can regard the applications you created before as Multiple page applications. The application we can create with Angular is thus the contrary, Single Page Application.

Here is a diagram from Microsoft Documentation that summarize the difference pretty well

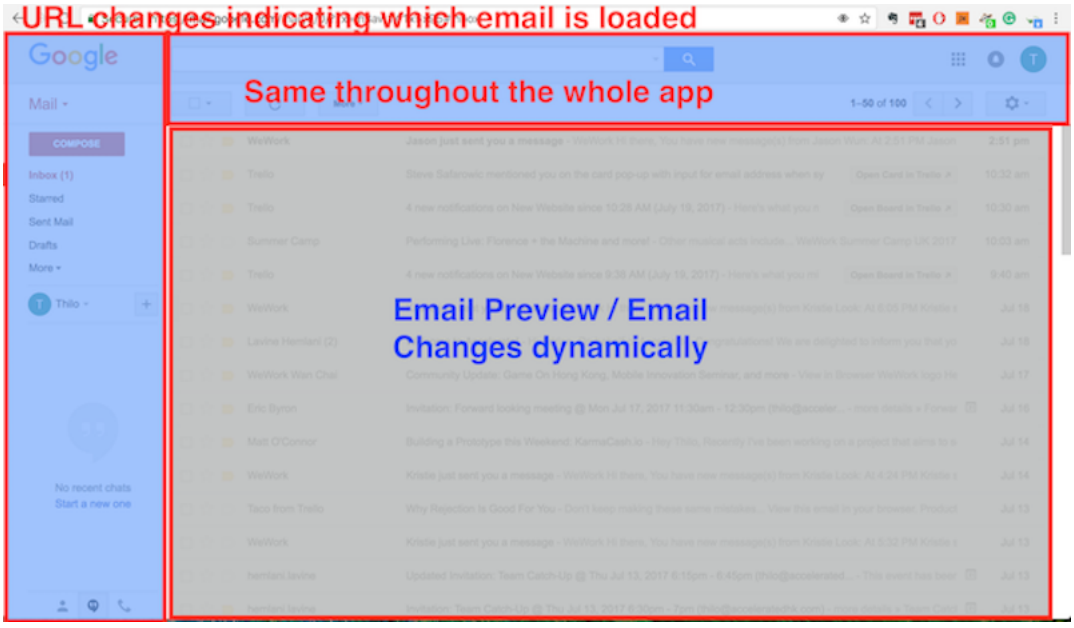


The rendering of SPA and MPA also differs. Here is a diagram showing the difference in the rendering process.



1.1.2 Gmail

One of the prominent example of a SPA is gmail. You have one page, which in its basic structure always looks the same. We can dynamically load any email but the basic structure doesnt change just like many desktop applications (like MS Office). This set the standard for modern web development. Almost all the big websites these days makes use of SPA. Just take a good look at YouTube or Facebook.

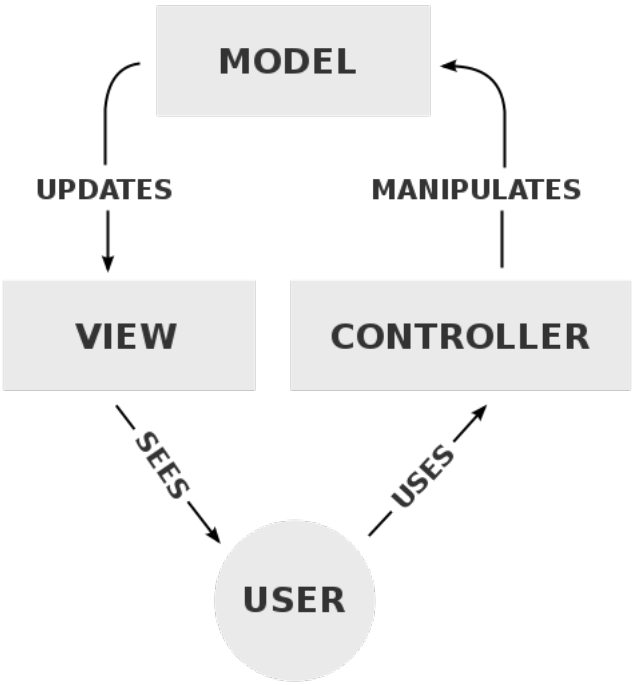


1.1.3 Different types of SPA Franework

1.1.4 MVVM vs MVS

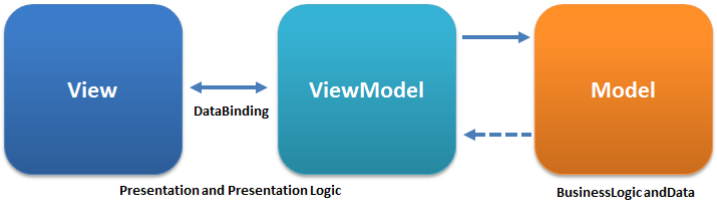
You may have heard the following two architectures. Model-View-Controller which specifies the three components in the architectures Model, View and Controller.

Here is the architecture for Model-View-Controller



In MVC, the users interacts with the controller which manipulates the model which is the data. The data is updated to view at the end.

Here is the architecture for Model-View-ViewModel



The biggest difference between MVVM and MVC lies in the reliance of Data Binding in MVVM. The data in ViewModel is bound to the view by the framework instead of updating through templates as in MVC.

For their application domains, nowadays MVC is mostly used in backend framework while MVVM is widely used in frontend framework.

1.2 Installation

1.3 TypeScript

1.4 TS node

2 Components

2.1 Components

2.2 creating component

2.3 Data Binding

2.4 Templates and Models

2.5 Breaking down Components

3 Template

3.1 Directives

3.2 Pipes

4 Forms

4.1 Template Driven Form

4.2 Model Driven Form

5 Assignment 1

6 Service

6.1 Services

6.2 Injectors

7 Routing

8 Http

8.1 HTTP

8.2 Authentication

9 Other Topics

9.1 Testing

9.2 Lifecycle Hooks

9.3 Full Stack Development Environment

9.4 JSON Web Token

10 Assignment 2

11 Introduction to Ionic

11.1 Installing

12 Ionic Components

12.1 Pages

12.2 Styling

13 Ionic Testing

13.1 Depoly on IOS

13.2 Depoly on Android

14 Ionic Native

15 Assignment 3

16 Extra Topics

16.1 TCP/IP Model

16.2 Angular Depolyment

16.3 OO Design Principles and Clean Code