ATTUNE

Angular 1 Vs.Angular 2

By Juned Laliwala

Short description of Angular's Version Conversion

- Basic Difference between An gular 1 and Angular 2
- Installation Section
- Syntax Differences

ATTUNE

Contents

Contents

1. Preface
    1.1. About This Guide
    1.2. Intended Audience
    1.3. Revision History

2. Introduction
    2.1. Angular 1
        2.1.1 Installation of Angular 1
    2.2 Angular 2
        2.2.1 Installation of Angular 2

3. Changes in Angular 1 to Angular 2
        3.1 Bindings
        3.2 Filters and Pipes Concept
        3.3 Local and Input Variables

4. Concept of Template Directives
    4.1 Bootstrapping
    4.2 Stylesheet
    4.3 Click Event
    4.4 Controller

5. More changes

6. Conclusion

1. Preface

1.1. About This Guide

This guide will help the individual or the developer who will see the basic scenario of the difference that have been monitored for the angular 1 and angular 2. We will also see the syntax difference that have been occurred for the conversion from the Angular 1 to Angular 2.

1.2. Intended Audience

This guide is particularly intended for the developers who are having the basic knowledge of the AngularJS and Angular 2. The developers must also have the knowledge of the Syntax occurring in both of the JS.

1.3. Revision History

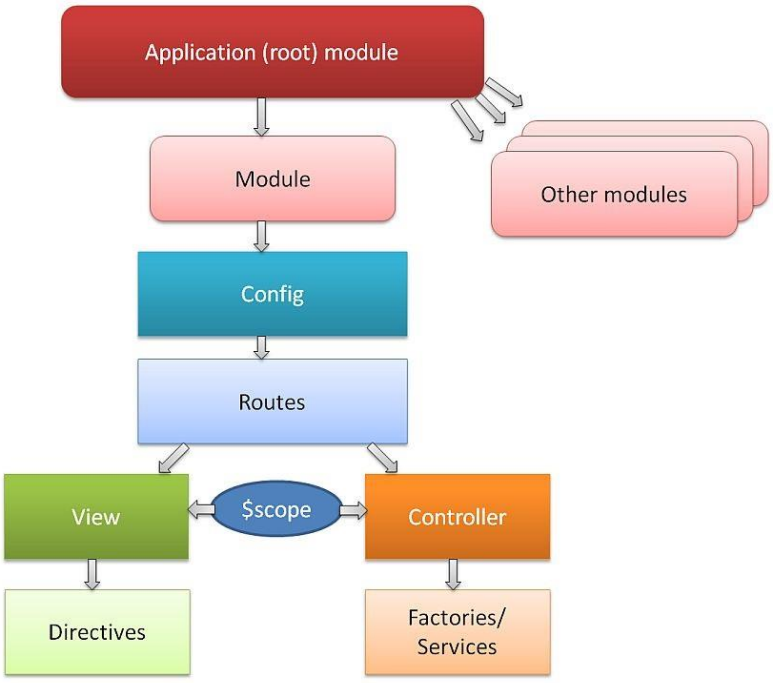
Getting Started with AngularJS and Getting Started with Angular 2 are the first version for the reference to this document. This is the Second version of the AngularJS and Angular 2 getting started.

2. Introduction

In this introduction part, we are now going to see the basic overview of the Angular 1 and Angular 2. In the coming section, we will come across the basic changes that have come for the angular versions.

2.1. Angular 1

Angular 1 is also well known by AngularJS. With the help of Angular 1, we can easily make any of the web or mobile based application using MVC (Model-View-Controller) type. We have already seen the various examples and the working of the application using MVC in the sections that we have already published in the previous articles.



2.1.1 Installation of Angular 1

Installation of the Angular 1 is very simple. There are usually two different ways to make our application to be Angular 1 dependent.

Way 1: We can download the zip file from the official website of the AngularJs (angularjs.org). Keep the extracted folder to the folder where your application resides.

Way-2: We can also use the CDN with which we can directly download the appropriate files necessary for any of the particular option.

We can also use the combination of both of the ways defined above.

2.2 Angular 2

Angular 2 is the enhanced version of AngularJS. Angular 2 is equipped with the new thing namely "typescript". We can easily work with angular 2 using typescript. As we know that this is the advanced concept of Angular, so there are various concepts as compared to AngularJS.

2.2.1 Installation of Angular 2

Installation of Angular 2 is typically complex as compared to simple AngularJS. We currently need to make some of the basic or the core files necessary to make our application Angular 2 dependent.

1. Package.json file for the making our application dependent on the npm package manager
2. Tsconfig.json which is nothing but the json file for the typescripting purpose.
3. Systemjs.config.js which is merely responsible for handling all the functionalities of the application.

After making the above three files, we need to install the above three files using npm.

To start the application, we will use the command “npm start”.

3. Changes in Angular 1 to Angular 2

As we are now moving towards the main concept of our book. We are going to see some of the basic changes like:

1. Bindings
2. Directives in Templates
3. Filtering
4. Modules/Controllers/Components
5. Style Sheets.

3.1 Bindings

In this section, we will come across the bindings that we are usually doing in the Angular. We will show the basic usage in both the aspects (Angular 1 and Angular2). In both of the aspects, we will have the same concept of binding.
Binding normally occurs within the two curly brackets. This will bind the value of the element to a property inside the controller associated with the particular template.
Let us first see the simple example of binding with the Angular 1 Application.

```
<!DOCTYPE html>
<html ng-app="example">
<head>
    <title>AngularJS</title>
    <script type="text/javascript" src="angular/angular.min.js"></script>
</head>

<body ng-controller="TestCtrl">
        <h1>Employee Details</h1>
            {{emp}}
</body>

<script>
    var app = angular.module('example',[]);

    app.controller('TestCtrl', function($scope){
        $scope.emp ="AttuneWW!!!";
    });
</script>

</html>
```

Fig: Angular 1 Binding Concept

In the example shown above, we can see that we have included the “angular.min.js” file which is nothing but the mandatory dependency for the angularjs application. In the next line, we can see that we have defined the controller named “TestCtrl” for body part and inside that we have applied the {{ }} which is nothing but the binding part.
Now inside the scripting part, we can see that we are applying the controller named “TestCtrl” and inside that we are applying the binding concept with $scope parameter to call the value.

Angular 2

As we can see from the diagram shown below that we are using “@Component” directive instead of the controller that we were using for the AngularJS Application. Inside the component part, we are using the template section which is normally used for the view section of the application.

```
import {Component} from '@angular/core';

@Component({
    selector:'my-app',
    template: `<h1>Student Details</h1><div>{{stud_name.name}}</div>`
})

export class AppComponent{
    stud_name = {name: 'Juned',desg:'trainer'};
}
```

Fig: Angular 2 Binding Concept

3.2 Filters and Pipes Concept

In Angular 1, we are using the word "filter". As the name suggests, filter is used for doing some of the specifying tasks like uppercase, lowercase, currency and so on as we have seen the whole basic concepts in one of our e-book named "Getting started with AngularJS". We will see some simple example to demonstrate you the use of filters in angularjs.

1. {{emp.uname | uppercase }} : Which will display "emp.uname" in Capital or uppercase letter.
2. {{emp.salary | currency }}: Which will display the value with "$" before it.

As we can see that we are using pipe (|) character to represent the filter. The word "pipe" is being used in Angular 2. Many (but not all) of the built-in filters from Angular 1 are built-in pipes in Angular 2.

3.3 Local and Input Variables

We all are aware about the concept of "ng-repeat" in Angular 1. In Angular 2, we are going to use the same concept but with the different name "*ngFor".
ngFor abbreviates for "For Angular". Let us see the example to demonstrate you the usage of the *ngFor.

```
<tr ng-repeat=”name in emp.name”>
<td> {{name.lname}} </td>
</tr>
```

As we can see that, we are using the ng-repeat directive for all the last names of the employees. This same concept can be represented in Angular 2 using the syntax as:

```
<tr *ngFor="let name of emp.name">
  <td>{{name.lname}}</td>
</tr>
```

so , we have seen that “ng-repeat” is used in Angular 1 whereas “*ngFor” is used in Angular 2.

4. Concept of Template Directives

In this section, we will come across the various directives used for the template section. We have seen various directives in Angular application, but in the angular 2 the same concept will be used but with different syntax. Let us take a simple example to demonstrate the use of various template directives.

4.1 Bootstrapping

In Angular 1, we are using “ng-app” for defining the module for the application. As we know that there is a unique module for each of the angular application.

<body ng-app= “TestApp”>

The above syntax represents the module “TestApp”. This is the simple task in Angular 1 but if we talk about the Angular 2, then we need to define it in two different files named “app.module.ts” and “main.ts”.

main.ts
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic'; import { AppModule } from './app.module'; platformBrowserDynamic().bootstrapModule(AppModule);

As we can see that we are importing two main important things particularly for the browser dependency and for the module of the application.

app.module.ts
import { NgModule } from '@angular/core'; import { BrowserModule } from '@angular/platform-browser';

```
import { AppComponent }  from './app.component';

@NgModule({
  imports: [ BrowserModule ],
  declarations: [ AppComponent ],
  bootstrap: [ AppComponent ]
})
export class AppModule { }
```

The above file shows us the example of the simple module that we have created in the main typescript file. We are importing all the necessary modules and the templates for the application. So we can easily see that the angular 2 application is much more tedious as compared to Angular 1.

We can see that we are importing “ngModule” from the core elements of the Angular Application. We are also importing the “AppComponent” from the app.module.ts file.

4.2 Stylesheet

If we talk about the Angular 1, then we are using “ng-class” for giving any of the section the style we want to apply for. This ng-class will be used as an expression often a key-value control object with each key of the object defined as a CSS class name, and each value defined as a template expression that evaluates to a Boolean value.

Angular 1

```
<div ng-class="{active: isActive}">
<div ng-class="{active: isActive,
                shazam: isImportant}">
```

As we can see that we have defined class as “Active”.

This same in angular 2, we can define using the below syntax:

Angular 2
<div [ngClass]="{active: isActive}"> <div [ngClass]="{active: isActive, shazam: isImportant}"> <div [class.active]="isActive">

As we can see that we have applied ngClass within the square bracket in angular 2. The rest of the expression will remain the same.

4.3 Click Event

As we know that in Angular 1, we were using the "ng-click" for getting the click event. There is a slight change in the Angular 2, we are going to use "(click)" syntax for triggering an event.

Angular 1
<button ng-click="SaveForm()"> <button ng-click="SaveForm($event)">

As we can see that we have used the button to demonstrate the example of the button click. Inside that we have applied ng-click directive for the triggering an event.

The same above demo can be illustrated using Angular 2 using the syntax shown below:

Angular 2
<button (click)="SaveForm()"> <button (click)="SaveForm($event)">

As we can see that we are using (click) for triggering an event action in Angular 2.

4.4 Controller

Controller is the main and core part of the Angular Application. Whatever the main functionality of the application is being handled by the controller. Controller as the name suggests is going to handle the model and the view part of the application.

Let us first take an example to illustrate the use of the controller for angular 1 application.

Angular 1

In Angular 2, we are not going to use the word “controller”. In fact, we are going to use the word named “component” for handling the controller function.

Angular 2
@Component({ moduleId: module.id, selector: 'my-list', templateUrl: 'homepage.html', styleUrls: ['app.component.css'], })

As we can see that we are using “@Component” directive for defining the controller. Inside that we are using three more parameters for defining the component. This section we have already discussed in the previous section of the e-book named “Getting Started with Angular 2”.

5. More changes

In this section, we will see the various changes similar to the one we have used in the previous sections.

Angular 1	Angular 2
Use of ng-switch ng-switch-when ng-switch-default	Use of ngSwitch *ngSwitchCase *ngSwitchDefault
Currency <td>{{pen.price | currency}}</td>	 <td>{{pen.price | currency: 'USD':true}}</td>
<tr ng-repeat="name in EmplList | **limitTo**:2:0">	<tr *ngFor="let name of EmplList| **slice**:0:2">
Number <td>{{student.RollNo | number}}</td>	<td>{{student.RollNo | number}}</td> <td>{{student.RollNo | number:'1.1-2'}}</td> <td>{{student.StudentRating | percent: '1.0-2'}}</td>

So, we have seen that the more changes that have being appeared into the Angular 2 Application as compared to AngularJS (Angular 1).

6. Conclusion

Throughout this guide, we have come across various functionalities and the syntax that are occurring for both the AngularJS and Angular 2. We have also seen that the Angular 2 is very much tedious as compared to Angular 1. The installation part is also some what complex. The overall functionality of the application is the same for both the Angular 1 and Angular 2.