Angular 2.0

tutorialspoint

SIMPLYEASYLEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia

https://twitter.com/tutorialspoint

About the Tutorial

Angular 2 is an open source JavaScript framework to build web applications in HTML and JavaScript, and has been conceived as a mobile first approach.

Audience

This tutorial is designed for software professionals who want to learn the basics of AngularJS 2 and its programming concepts in simple and easy steps. It describes the components of AngularJS 2 with suitable examples.

Prerequisites

You should have a basic understanding of JavaScript and any text editor. As we are going to develop web-based applications using Angular 2, it will helpful if you have an understanding of other web technologies such as HTML, CSS, AJAX, AngularJS, etc.

Copyright & Disclaimer

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

1. ANGULAR 2 — OVERVIEW

Angular 2 is an open source JavaScript framework to build web applications in HTML and JavaScript, and has been conceived as a mobile first approach. The beta version of Angular 2 was released in March 2014.

Why use Angular 2?

- Angular 2 is simpler than Angular 1. The concepts here are easier to understand.
- You can update the large data sets with minimal memory overhead.
- It will speed up the initial load through server side rendering.

Features of Angular 2

- Angular 2 is faster and easier than Angular 1.
- It supports the latest version of browsers and also supports old browsers including IE9+ and Android 4.1+.
- It is a cross-platform framework.
- Angular 2 is mainly focused on mobile apps.
- Code structure is more simplified than the previous version of Angular.

Advantages of Angular 2

- If an application is heavy, then Angular 2 keeps it fully UI (User Interface) responsive.
- It uses the server side rendering for fast views on mobile.
- It works well with ECMAScript and other languages that compile with JavaScript.
- It uses dependency injection to maintain applications without writing lengthy codes.
- The applications here have a component-based approach.

Disadvantages of Angular 2

- Since Angular 2 is a newly introduced framework, there is less online community support.
- It takes time to learn if you are new to Angular 2.

2. ANGULAR 2 — ENVIRONMENT

In this chapter, let us discuss the Angular 2 development environment in detail.

- Angular uses TypeScript, which is a primary language for developing Angular applications.

- TypeScript is a super set of JavaScript, which is migrated to TypeScript. Here, the code written in TypeScript makes it less prone to runtime errors.

To set up the development environment, follow these steps:

Step 1: Create a project folder in your local drive by typing the commands in the command prompt as given below.

```
mkdir angular2-demo
cd angular2-demo
```

Creating Configuration Files

The creation of configuration files follows the step mentioned above.

Step 2: You need to create **tsconfig.json** which is the TypeScript compiler configuration file. It guides the compiler to generate JavaScript files.

```
{
  "compilerOptions": {
    "target": "es5",
    "module": "system",
    "moduleResolution": "node",
    "sourceMap": true,
    "emitDecoratorMetadata": true,
    "experimentalDecorators": true,
    "removeComments": false,
    "noImplicitAny": false
  },
  "exclude": [
    "node_modules",
    "typings/main",
    "typings/main.d.ts"
  ]
}
```

Step 3: Create a **typings.json** file in your project folder *angular2-demo* as shown below:

typings.json

```
{
  "globalDependencies": {
    "core-js": "registry:dt/core-js#0.0.0+20160602141332",
    "jasmine": "registry:dt/jasmine#2.2.0+20160621224255",
    "node": "registry:dt/node#6.0.0+20160621231320"
  }
}
```

A large number of libraries of the JavaScript extends JavaScript environment with features and syntax which is not natively recognized by the TypeScript compiler. The **typings.json** file is used to identify TypeScript definition files in your Angular application.

In the above code, there are three typing files as shown below:

- **core-js**: It brings ES2015/ES6 capabilities to our ES5 browsers.
- **jasmine**: It is the typing for Jasmine test framework.
- **node**: It is used for the code that references objects in the **nodejs** environment.

These typing files are used in the development of larger Angular applications.

Step 4: Add the **package.json** file to your angular2-demo project folder with the code given below:

package.json

```
{
  "name": "angular2-demo",
  "version": "1.0.0",
  "scripts": {
    "start": "concurrent \"npm run tsc:w\" \"npm run lite\" ",
    "tsc": "tsc",
    "tsc:w": "tsc -w",
    "lite": "lite-server",
    "typings": "typings",
    "postinstall": "typings install"
  },

  "license": "ISC",
  "dependencies": {
    "angular2": "2.0.0-beta.7",
    "systemjs": "0.19.22",
    "es6-promise": "^3.0.2",
    "es6-shim": "^0.33.3",
    "reflect-metadata": "0.1.2",
    "rxjs": "5.0.0-beta.2",
    "zone.js": "0.5.15"
  },
  "devDependencies": {
    "concurrently": "^2.0.0",
    "lite-server": "^2.1.0",
    "typescript": "^1.7.5",
    "typings":"^0.6.8"
  }
}
```

The **package.json** will contain the packages that our apps require. These packages are installed and maintained with **npm** (Node Package Manager). To install *npm* click here.

Step 5: To install packages, run the **npm** command in the command prompt as given below.

```
npm install
```

Error messages in red may appear while installing npm. These messages have to be ignored.

Creating Our First Angular Component

A component is the fundamental concept of Angular. A component is a class that controls a view template - a part of a web page where information to the user is displayed and user feedback is responded to. Components are required to build Angular apps.

Step 6: Create a sub-folder called *app/* inside your project folder to place the Angular app components. You can use the following command to create the folder:

```
mkdir app

cd app
```

Step 7: The files which you create need to be saved with the **.ts** extension. Create a file called **environment_app.component.ts** in your *app/* folder with the below code:

environment_app.component.ts

```
import {Component, View} from "angular2/core";


@Component({

   selector: 'my-app'

})


@View({

  template: '<h2>My First Angular 2 App</h2>'

})


export class AppComponent {


}
```

- The above code will import the ***Component*** and the ***View*** package from ***angular2/core***.
- The ***@Component*** is an Angular 2 ***decorator*** that allows you to associate metadata with the component class.
- The ***my-app*** can be used as HTML tag and also as a component.
- The ***@view*** contains a ***template*** that tells Angular how to render a view.
- The ***export*** specifies that, this component will be available outside the file.

Step 8: Next, create the **environment_main.ts** file with the following code:

environment_main.ts

```
import {bootstrap} from "angular2/platform/browser"
import {AppComponent} from "./environment_app.component"

bootstrap(AppComponent);
```