# Angular入门基础

## 第1节AngularJs概述

### 1.1什么是AngularJs

AngularJs是一个Javascript framework，也就是一个js框架，它能够帮助我们创建Web、Mobile、Desktop Application。

AngularJS是由谷歌公司创建，它遵循的是MIT协议，开源开放。

它的官网地址是：

<https://angularjs.org/> 1.X系列，2009年发布；

<https://angularjs.io/> 2.X、4.X系列，2016年9月发布2系列，2017年3月发布4系列。

AngularJs这套js框架针对于SPA（Single-Page-Application）应用以及商业业务流程性应用的开发具有特别的优势。

目前已经有许许多多的个人与企业利用AngularJs进行项目开发，也已经有许多成熟的案例，这里有一个网站进行了项目罗列。

<https://www.madewithangular.com/>

注：SPA--> <http://blog.csdn.net/u012974787/article/details/74856870/>

### 1.2AngularJs的优势

我们已经拥有了许多的JavaScript Framework来帮助我们进行Web Application应用的开发，那么我们为什么还要选择AngularJs呢？

1. Dependency Injection依赖注入；
2. Two Way Data-Binding双向数据绑定；
3. Test测试；
4. Model-View Controller控制；
5. Directive（指令），Pipe（管道操作），service（服务），animate（动画）等。

M：model（数据）

V：view（显示）

C：controller（控制器）

### 1.3angular版本之间的比较

Angular1=AngularJs：一个应用非常广泛的JS框架，发布时间久远（2009年google发布）。

Angular2=Angular：相对于Angular1，是一个完全重构的JS框架，虽然有一些概念得以沿用，但与Angular1基本没有什么关系了（2016年9月发布正式版）。

Angular4=Angular2+：是对Angular2的向下兼容性做了一些更新与改进（由于为了解决版本冲突问题，官方跳过了Angular3这个版本）（2017年3月正式版）。

### 1.4建立第一个angular4应用

从CLI开始Angular4项目（CLI，Command-Line-Interface是一个非常好的工具，特别是对于越来越复杂的Angular项目来说，它能够让我们更多的关注流程，而不需要去关注代码内容）。

<https://cli.angular.io/>

1.安装angular的CLI需要NodeJs的环境基础，<https://nodejs.org/>；

2.运行：npm install -g @angular/cli，进行angular的CLI安装；

3.运行：ng new angular-project-name，进行angular项目的创建（注意：因为需要安装node\_modules，所以会比较慢）；

4.运行：cd angular-project-name，进入到创建的项目目录中；

5.运行：ng serve （注意：不是server，是serve），运行刚刚创建的angular项目，运行地址为： <http://localhost:4200>。

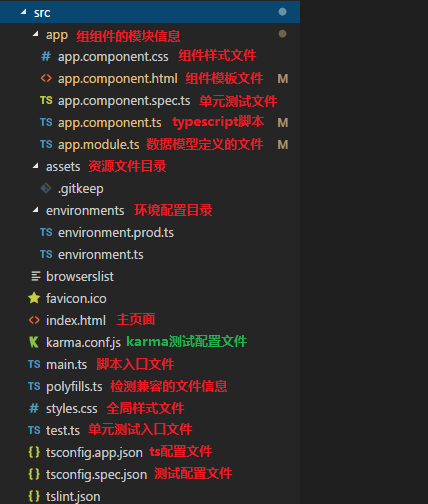
注意：如果CMD命令窗口新建项目编译失败，首先使用Git Bash打开进行新建，再编译运行。

### 1.5项目的文件目录及结构

1）主文件目录



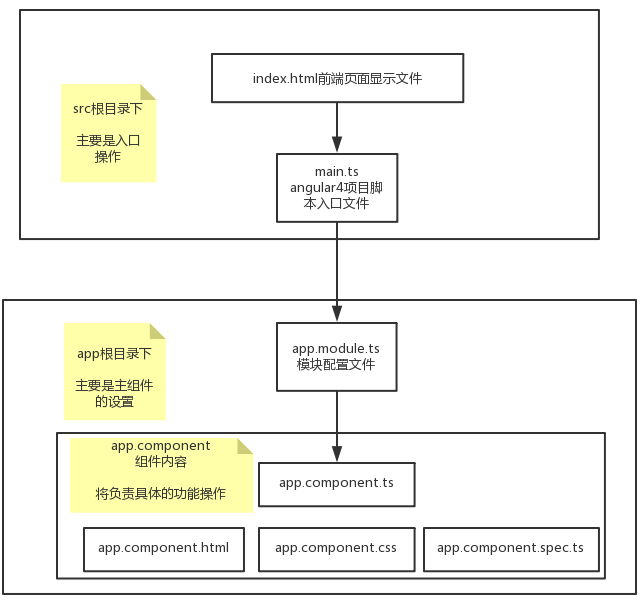
1. src文件目录



1. 其他

Typings.d.ts 系统模块定义文件

1. angular入口结构



### 1.6建立一个输入文本框angular4应用

1. app.module.ts引入表单依赖

|  |
| --- |
| import { FormsModule } from '@angular/forms';  import { BrowserModule } from '@angular/platform-browser';  import { NgModule } from '@angular/core';  import { AppComponent } from './app.component';  @NgModule({  declarations: [  AppComponent  ],  imports: [  BrowserModule,  FormsModule  ],  providers: [],  bootstrap: [AppComponent]  })  export class AppModule { } |

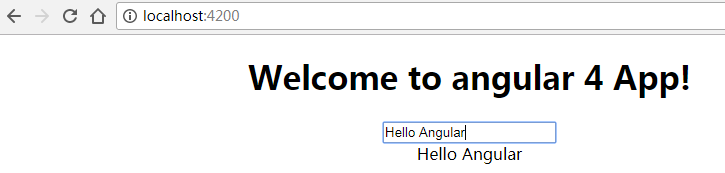
2）app.component.ts修改title

|  |
| --- |
| import { Component } from '@angular/core';  @Component({  selector: 'app-root',  templateUrl: './app.component.html',  styleUrls: ['./app.component.css']  })  export class AppComponent {  title = 'angular 4 App';  } |

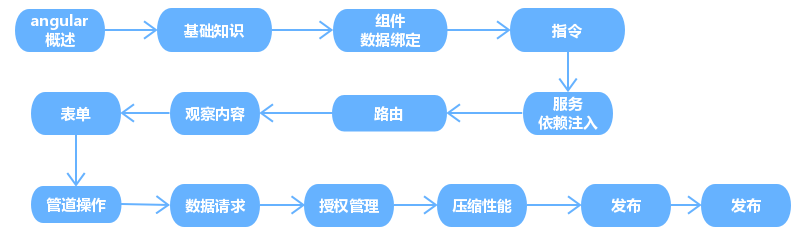
3）增加一个imput输入框并显示输入内容

|  |
| --- |
| <!--The content below is only a placeholder and can be replaced.-->  <div style="text-align:center">  <h1>  Welcome to {{ title }}!  </h1>  <input type="text" [(ngModel)]="name"/>  <br/>  {{ name }}  </div> |

4）效果展示



### 1.7angular总结结构



### 1.8TypeScript

TypeScript比普通js功能更为强大，比如数据类型、类的定义、接口操作等；

TypeScript最终编译为JavaScript。

### 1.9bootstrap框架安装

1）方式一

<http://www.bootcdn.cn/bootstrap/>

引入CDN链接依赖

<link href="https://cdn.bootcss.com/bootstrap/4.1.1/css/bootstrap-grid.css" rel="stylesheet">

添加按钮组：

<div class="btn-group" role="group" aria-label="...">

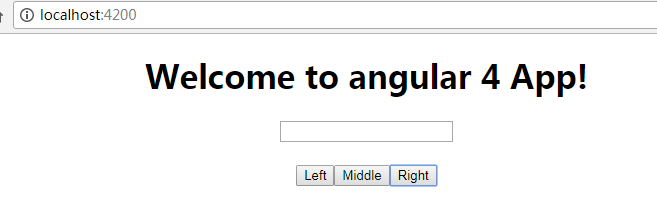
<button type="button" class="btn btn-default">Left</button>

<button type="button" class="btn btn-default">Middle</button>

<button type="button" class="btn btn-default">Right</button>

</div>

保存后自动编译



1. 方式二

安装bootstrap资源到本地

my-first-angular-app>npm i bootstrap -S

+ bootstrap@4.1.1

added 1 package and updated 1 package in 53.542s

angular.json引入bootstrap依赖

"styles": [

"src/styles.css",

"../node\_modules/bootstrap/dist/css/bootstrap.min.css"

],

## 第2节基础知识

### 2.1如何搭建一个应用

现象描述：在运行ng serve以后，打开<http://localhost:4200/>运行程序，根据上一节的知识延续，将显示一个带有title及部分带bootstra框架页面的内容，在第一次打开项目的时候，会显示Loading...的内容，而之后刷新页面的时候，将不会再出现Loading...的内容。

问题思路：查看index.html源码，会看到一些引入的js脚本代码，例如：

<script type="text/javascript" src="runtime.js"></script>

<script type="text/javascript" src="polyfills.js"></script>

<script type="text/javascript" src="styles.js"></script>

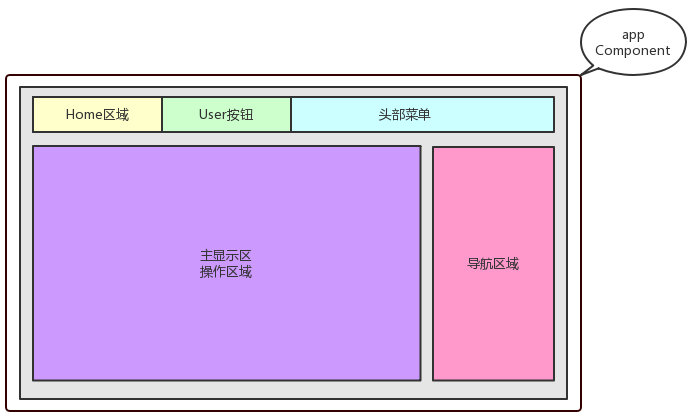
<script type="text/javascript" src="vendor.js"></script>

<script type="text/javascript" src="main.js"></script>

Angular已经将我们的项目通过webpack进行了脚本打包，而再次刷新操作以后，代码已经被缓存，执行速度将不再影响，所以我们就看不到Loading...进度操作了？思考一下，这种说明是否正确？（不正确）运作的原理并非如此，我们需要从Angular4的启动原理来理解这一现象的原因。

回想一下之前的内容，我们有提到Loading...到底去了哪里，我们通过入口文件及主文件等内容的渲染操作会发现以下结果：Angular4最终还是一个js的框架，它是对HTML的DOM节点进行操作的，我们的项目在加载完成以后，其实已经将<app-root>Loading...</app-root>的代码进行了DOM渲染的替换，所以Loading...再次刷新页面时看不到，所以，这与脚本缓存的问题无关，而是DOM对象的渲染替换问题。

### 2.2组件的重要性



头部菜单、按钮、主显示区、导航区域等内容都可以是单独的或是嵌套的组件，组件操作在Angular4开发中项目的核心及基础。

### 2.3创建一个组件