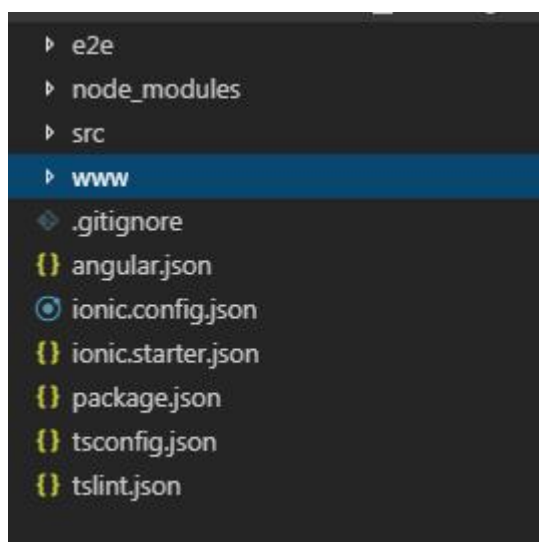

基于 Angular 的 Ionic 目录结构分析、核心文件分析、执行流程讲解

目录

一、 Ionic4.x 目录结构分析.....	1
二、 Ionic4.x src 下面文件分析.....	2
三、 app.module.ts 分析.....	3
四、 app-routing.module.ts 分析.....	4
五、 路由匹配以及项目执行流程.....	4

一、 Ionic4.x 目录结构分析



e2e: 端对端测试文件

node_modules : 项目所需要的依赖包

resources : android/ios 资源（更换图标和启动动画）

src: 开发工作目录，页面、样式、脚本和图片都放在这个目录下

www: 静态文件，ionic build --prod 生成的单页面静态资源文件

platforms: 生成 android 或者 ios 安装包需要的资源---(cordova platform add android 后会生成)

plugins: 插件文件夹，里面放置各种 cordova 安装的插件

config.xml: 打包成 app 的配置文件

package.json: 配置项目的元数据和管理项目所需要的依赖

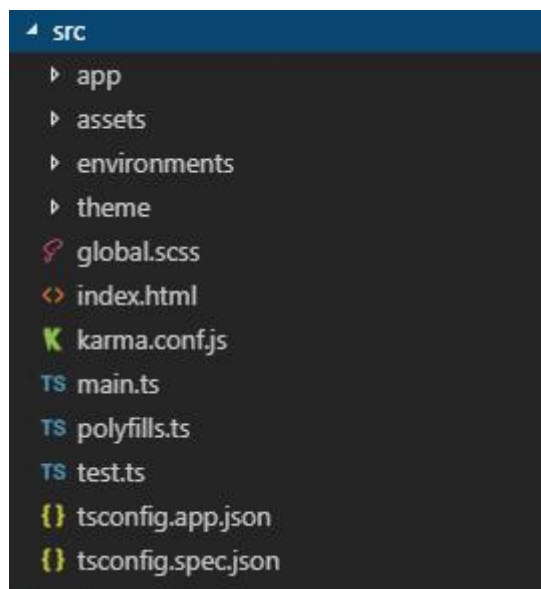
ionic.config.json、**ionic.starter.json**: ionic 配置文件

angular.json angular 配置文件

tsconfig.json: TypeScript 项目的根目录，指定用来编译这个项目的根文件和编译选项

tslint.json: 格式化和校验 typescript

二、Ionic4.x src 下面文件分析



app: 应用根目录（组件、页面、服务、模块...）

assets: 资源目录（静态文件（图片，js 框架...）

theme: 主题文件，里面有一个 scss 文件，设置主题信息。

global.scss: 全局 css 文件

index.html: index 入口文件

main.ts: 主入口文件

karma.conf.js/test.js: 测试相关的配置文件

polyfills.ts: 这个文件包含 Angular 需要的填充，并在应用程序之前加载

app 下面文件分析:



三、app.module.ts 分析

```
/*这个文件是Ionic 根模块，告诉Ionic如何组装应用*/

//angular核心
import { NgModule } from '@angular/core';
//BrowserModule, 浏览器解析的模块
import { BrowserModule } from '@angular/platform-browser';
//路由
import { RouteReuseStrategy } from '@angular/router';
//ionic核心模块
import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
//启动画面插件相关服务
import { SplashScreen } from '@ionic-native/splash-screen/ngx';
//导航条插件相关服务
import { StatusBar } from '@ionic-native/status-bar/ngx';
//路由模块
import { AppRoutingModule } from './app-routing.module';
//根组件
import { AppComponent } from './app.component';

@NgModule({
  declarations: [AppComponent], //配置当前项目运行的组件
  entryComponents: [],
  imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModule], //配置当前模块运行依赖的其他模块
  providers: [ //配置项目所需要的服务
    StatusBar,
    SplashScreen,
    { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }
  ],
  bootstrap: [AppComponent] //指定应用的主视图(称为根组件) 通过引导根AppModule来启动应用，这里一般写的是根组件
})
export class AppModule {}
```

四、app-routing.module.ts 分析

```
import { NgModule } from '@angular/core';
//路由相关模块
import { PreloadAllModules, RouterModule, Routes } from '@angular/router';
//配置路由
const routes: Routes = [
  { path: '', loadChildren: './tabs/tabs.module#TabsPageModule' }
];

@NgModule({
  imports: [
    RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })
  ],
  exports: [RouterModule]
})
export class AppRoutingModule {}
```

五、路由匹配以及项目执行流程