

### 这是一个典型的[Cordova](http://ionicframework.com/docs/v1/what-is/" \l "cordova) 项目结构，可以在里面安装native插件。

### ./src/index.html 是app的主入口

src/index.html is the main entry point for the app, though its purpose is to set up script and CSS includes and bootstrap, or start running our app. We won’t spend much of our time in this file.

For your app to function, Ionic looks for the <ion-app> tag in your HTML.

为了程序能够运行，Ionic要寻找<ion-app></ion-app> 标签。

底部<script src="build/main.js"></script>里的build/main.js是个联合文件，包括Ionic, Angular and your app’s JavaScript.

cordova.js 在本地开发过程中将会是404状态，在Cordova 创建应用的过程中才会注入项目中。

**./src/**

src文件夹中有一些原始的、未编译的代码，构建Ionic App的大部分工作都在这里发生。当我们运行ionic serve时，src/内的代码将会被转成浏览器能识别的相应版本的JavaScript。这意味着我们可以使用TypeScript开发，然后转成js。

src/app/app.module.ts是app的入口点。在这个文件里面，有下列代码：

@NgModule({

declarations: [MyApp, HelloIonicPage, ItemDetailsPage, ListPage],

imports: [ BrowserModule, IonicModule.forRoot(MyApp)],

bootstrap: [IonicApp],

entryComponents: [MyApp, HelloIonicPage, ItemDetailsPage, ListPage],

providers: []

})

export class AppModule {}

每个app都必须有一个根模块（root module）来控制其他的应用（类似于Angular中的ng-app），并用ionicBootstrap来引导app（bootstrap our app）。

在src/app/app.component.ts中，我们可以为MyApp设置根组件（它是在app中第一个被加载的组件），通常它是一个空壳，以供其他组件加载进来。在app.component.ts中，我们为src/app/app.html设置模板。

**./src/app/app.html**

打开app.html，模板中包括如下两句：

<ion-menu [content]="content">

<ion-nav [root]="rootPage" #content swipeBackEnabled="false"></ion-nav>

ion-menu定义了侧栏菜单，ion-nav定义了主体内容区。[content]与来自ion-nav的nav本地变量绑定，这样侧栏菜单就知道了它应该围绕着谁滑动。

当ion-nav加载时，rootPage变量所引用的部分将会是根页面。

在src/app/app.component.ts中，

import { HelloIonicPage } from '../pages/hello-ionic/hello-ionic';

...

rootPage = HelloIonicPage;

我们看到rootPage设置为HelloIonicPage，所以该页将是在nav控制器中第一个被加载的。

### 创建一个页面

打开src/pages可以看到每个页面都有自己的文件夹，里面有同名的.html / .scss / .ts文件，虽然这种结构并不是必须的，但是这样可以使项目条理清晰。

打开src/pages/hello-ionic/hello-ionic.ts可以看到@Component里面有个selector，页面都是动态加载的，本不需要选择器，但是如果想要覆盖.scss默认样式就用得到了。

打开src/pages/hello-ionic/hello-ionic.html，<ion-navbar>是导航条的模板，当我们导航到本页时，导航条的button和title也会过渡过来。

其余的是主体内容的标准Ionic代码。

**创建额外页面**，我们只需要正确配置title和导航条需要展示的东西。

src/pages/list/list.ts创建了一个包含若干条目的列表页，该页与HelloIonicPage页面类似。里面有段代码：

itemTapped(event, item) {

this.navCtrl.push(ItemDetailsPage, {

item: item });

}

ItemDetailsPage一开始就存在，在list.ts中引入：

import { ItemDetailsPage } from '../item-details/item-details';

### BUTTON

<button ion-button color="secondary">Secondary</button>



<button ion-button color="danger" outline>Danger</button>



<button ion-button color="dark" clear>Dark</button>



img文件夹要和icon在同一个父文件夹下并列。