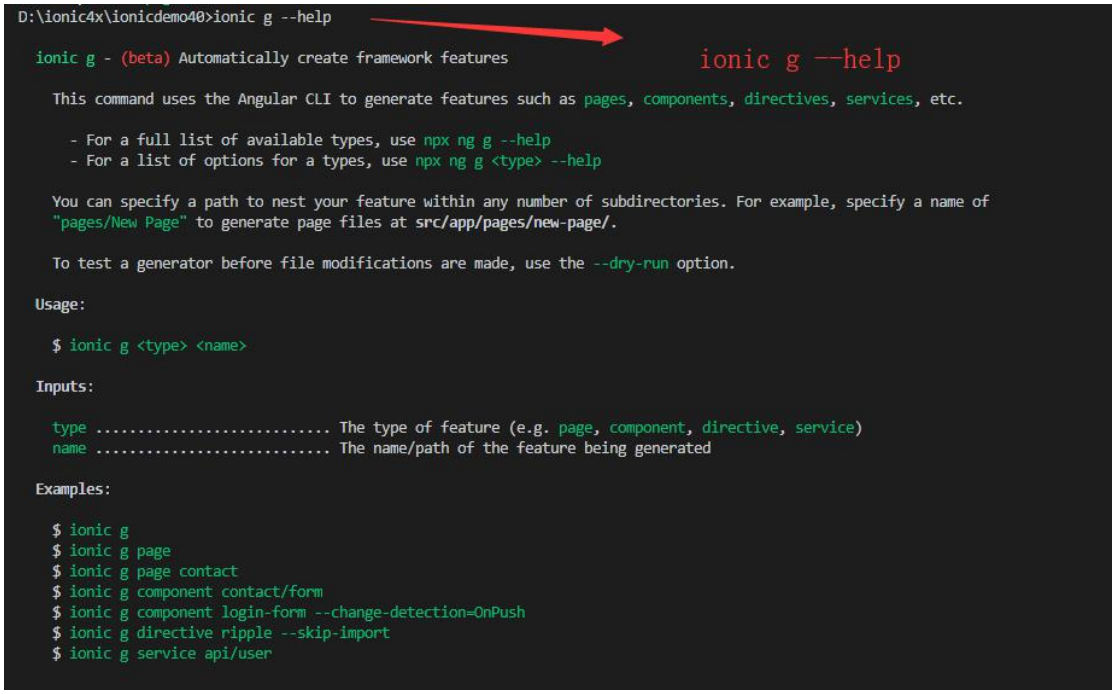
Ionic4.x 创建页面以及页面跳转、Ionic4.x 新增 tabs 页面、Ionic4.x 自定义公共模块、数据渲染等

一、Ionic4.x 创建页面以及页面跳转

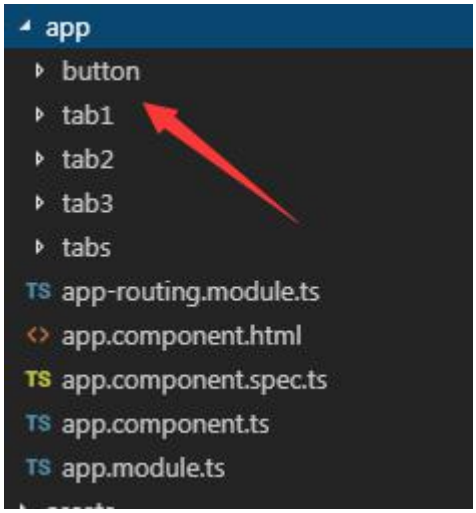
Ionic4.x 中创建页面和 ionic3 中是一样的，下面我们一步一步给大家看看在 ionic4.x 中如何创建页面以及进行页面跳转

1、cd 到我们的项目目录
2、通过 ionic g page 页面名称　如下图所示

```
D:\ionic4x\ionicdemo40>ionic g page button
> ng generate page button
CREATE src/app/button/button.module.ts (543 bytes)
CREATE src/app/button/button.page.html (133 bytes)
CREATE src/app/button/button.page.spec.ts (691 bytes)
CREATE src/app/button/button.page.ts (256 bytes)
CREATE src/app/button/button.page.scss (0 bytes)
UPDATE src/app/app-routing.module.ts (495 bytes)
[OK] Generated page!
```



3、创建完成组件以后会在 **src** 目录下面多一个 **button** 的目录，它既是一个页面也是一个模块。

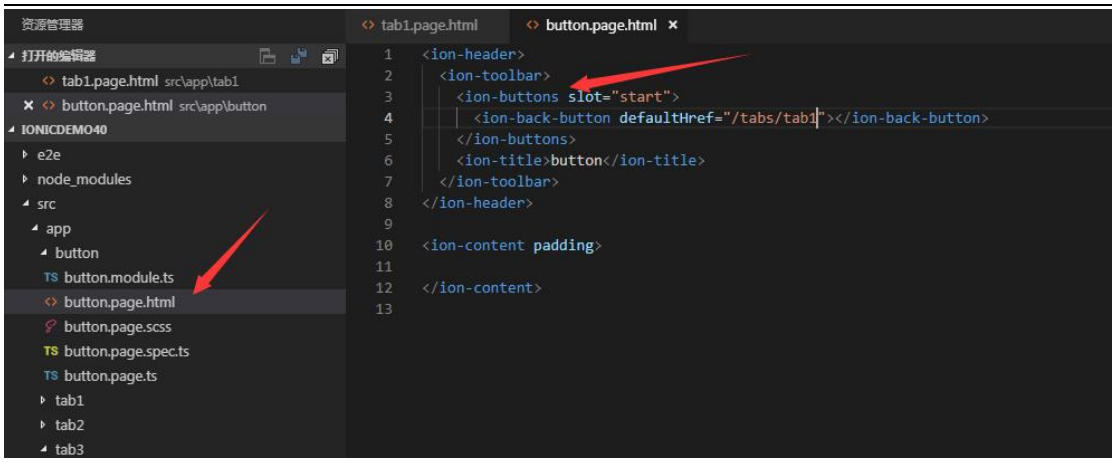


4、如果我们想在 tab1 里面写一个按钮点击跳转到 button 页面的话我们可以通过使用 Angular 的路由跳转。在 ionic4.x 中路由是完全基于 angular，用法几乎和 angular 一样。

tab1.page.html

```
<ion-header>
  <ion-toolbar>
    <ion-title>
      Tab One
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>

  <ion-button color="primary" [routerLink]="['/button']"> 跳转到按钮页面 </ion-button>


  <ion-button color="primary" routerLink="/button"> 跳转到按钮页面 </ion-button>

</ion-content>

```

5、ionic4.x 中跳转到其他页面不会默认加返回按钮，如果我们想给 button 页面加返回的话需要找到 button 对应的 button.page.html，然后在再头部加入 ion-back-button。如下图



二、Ionic4.x 新增底部 tabs 页面

1、创建 tab4 模块

```
ionic g page tab4
```

2、修改根目录里 app-routing.module.ts 文件里面的路由配置，去掉默认增加的路由

3、tabs.router.module.ts 中新增路由

```
{
        path: 'tab4', loadChildren: '../tab4/tab4.module#Tab4PageModule'
}
```

4、tabs.page.html 中新增底部 tab 切换按钮

```
<ion-tabs>

  <ion-tab-bar slot="bottom">
    <ion-tab-button tab="tab1">
      <ion-icon name="flash"></ion-icon>
      <ion-label>Tab One</ion-label>
    </ion-tab-button>

    <ion-tab-button tab="tab2">
      <ion-icon name="apps"></ion-icon>
      <ion-label>Tab Two</ion-label>
    </ion-tab-button>

    <ion-tab-button tab="tab3">
```

```
        <ion-icon name="send"></ion-icon>
        <ion-label>Tab Three</ion-label>
      </ion-tab-button>


      <ion-tab-button tab="tab4">
          <ion-icon name="settings"></ion-icon>
          <ion-label>Tab four</ion-label>
        </ion-tab-button>
  </ion-tab-bar>

</ion-tabs>
```

三、Ionic4.x 中自定义公共模块

1、创建公共模块以及组件

```
ionic g module module/slide

ionic g component module/slide
```

2、公共模块 slide.module.ts 中暴露对应的组件

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { SlideComponent } from './slide.component';

@NgModule({
  declarations: [SlideComponent],
  imports: [
    CommonModule
  ],
  exports:[SlideComponent]
})
export class SlideModule { }
```

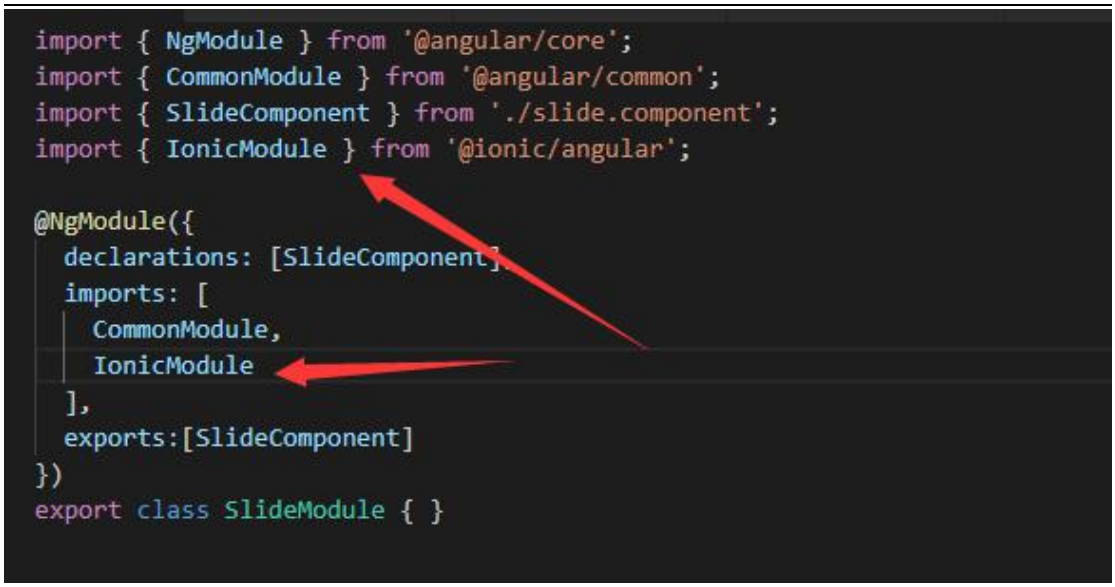
3、用到的地方引入自定义模块，并依赖注入自定义模块

```
import { SlideModule } from '../module/slide/slide.module';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    SlideModule,
    RouterModule.forChild(routes)
  ],
  declarations: [Tab4Page]
})
```

4、使用自定义模块

```
<app-slide></app-slide>
```

四、Ionic4.x 自定义公共模块中使用 ionic 内置组件


import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { SlideComponent } from './slide.component';
import { IonicModule } from '@ionic/angular';
@NgModule({
declarations: [SlideComponent],
imports: [
CommonModule,
IonicModule
],
exports:[SlideComponent]
})
export class SlideModule { }