

# Angular4.x 中的路由

主讲教师：(大地)

合作网站: [www.itying.com](http://www.itying.com) (IT 营)  
合作网站: [www.ionic.wang](http://www.ionic.wang) (Ionic 中文网)

## 目录

一、Angular 命令创建一个配置好路由的项目 .....	1
二、Angular4.x 在已有的项目中配置路由 .....	3
三、Angular routerLink 页面跳转 默认跳转路由 .....	5
四、Angular routerLinkActive 设置 routerLink 默认选中路由.....	6
五、路由的动态传值.....	6
六、路由的 js 跳转.....	7
七、路由的 js 跳转 get 传值.....	8
八、父子路由.....	9

## 一、Angular 命令创建一个配置好路由的项目

### 1. 命令创建项目

```
ng new demo02 --routing
```

```
G:\angularjs\demo>ng new demo02 --routing
installing ng
  create .editorconfig
  create README.md
  create src\app\app-routing.module.ts
  create src\app\app.component.css
  create src\app\app.component.html
  create src\app\app.component.spec.ts
  create src\app\app.component.ts
  create src\app\app.module.ts
  create src\assets\.gitkeep
  create src\environments\environment.prod.ts
  create src\environments\environment.ts
  create src\favicon.ico
  create src\index.html
  create src\main.ts
  create src\polyfills.ts
```

## 2. 创建需要的组件

```
ng g component home
ng g component news
ng g component newscontent
```

## 3. 找到 app-routing.module.ts 配置路由

引入组件

```
import { HomeComponent } from './home/home.component';
import { NewsComponent } from './news/news.component';
import { NewscontentComponent } from './newscontent/newscontent.component';
```

配置路由

```
const routes: Routes = [
  {path: 'home', component: HomeComponent},
  {path: 'news', component: NewsComponent},
  {path: 'newscontent/:id', component: NewscontentComponent},
  {
    path: '',
    redirectTo: '/home',
```

```
    pathMatch: 'full'
  }
];
```

4. 找到 `app.component.html` 根组件模板，配置 `router-outlet` 显示动态加载的路由

```
<h1>
  <a routerLink="/home">首页</a>
  <a routerLink="/news">新闻</a>
</h1>

<router-outlet></router-outlet>
```

## 二、Angular4.x 在已有的项目中配置路由

1. 新建组件

```
ng g component home
ng g component news
ng g component newscontent
```

2. 新建 `app-routing.module.ts` ,`app-routing.module.ts` 中引入模块

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
```

3. `app-routing.module.ts` 中引入组件

```
import { HomeComponent } from './home/home.component';
import { NewsComponent } from './news/news.component';

import { NewscontentComponent } from './newscontent/newscontent.component';
```

3. `app-routing.module.ts` 中配置组件

```
const routes: Routes = [  
  
  {path: 'home', component: HomeComponent},  
  {path: 'news', component: NewsComponent},  
  {path: 'newscontent/:id', component: NewscontentComponent},  
  {  
    path: '',  
    redirectTo: '/home',  
    pathMatch: 'full'  
  }  
  
];
```

#### 4. app-routing.module.ts 中配置模块 暴露模块

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

#### 5. 在 app.module.ts 引入刚才定义的路由

```
import { AppRoutingModule } from './app-routing.module';
```

#### 6.app.module.ts 里面的 import 注册这个路由模块

```
imports: [  
  BrowserModule,  
  AppRoutingModule  
]
```

#### 7.找到 app.component.html 根组件模板，配置 router-outlet 显示动态加载的路由

```
<h1>
  <a routerLink="/home">首页</a>
  <a routerLink="/news">新闻</a>
</h1>

<router-outlet></router-outlet>
```

### 三、Angular routerLink 页面跳转 默认跳转路由

```
<a routerLink="/home">首页</a>
<a routerLink="/news">新闻</a>
```

```
//刚进来路由为空跳转的路由
{
  path: '',
  redirectTo: 'home',
  pathMatch: "full"
}

//匹配不到路由的时候加载的组件 或者跳转的路由
{
  path: '**', /*任意的路由*/
  // component: HomeComponent
  redirectTo: 'home'
}
```

## 四、Angular routerLinkActive 设置 routerLink 默认选中路由

```
<h1>
  <a routerLink="/home" routerLinkActive="active">首页</a>
  <a routerLink="/news" routerLinkActive="active">新闻</a>
</h1>
```

```
.active{
  color:red;
}
```

## 五、路由的动态传值

### 1.配置动态路由

```
const routes: Routes = [
  {path: 'home', component: HomeComponent},
  {path: 'news', component: NewsComponent},
  {path: 'newscontent/:id', component: NewscontentComponent},
  {
    path: '',
    redirectTo: '/home',
    pathMatch: 'full'
  }
];
```

### 2.获取动态路由的值

```
import { Router, ActivatedRoute, Params } from '@angular/router';
```

```
constructor( private route: ActivatedRoute) {
```

```
}
```

```
ngOnInit() {  
  
    console.log(this.route.params); //  
  
    this.route.params.subscribe(data=>this.id=data.id);  
  
}
```

## 六、路由的 js 跳转

### 1. 引入

```
import { Router } from '@angular/router';
```

### 2. 初始化

```
export class HomeComponent implements OnInit {  
    constructor(private router: Router) {  
  
    }  
  
    ngOnInit() {  
    }  
  
    goNews(){  
        // this.router.navigate(['/news', hero.id]);  
  
        this.router.navigate(['/news']);  
    }  
}
```

### 3. 路由跳转

```
this.router.navigate(['/news', hero.id]);
```

## 七、路由的 js 跳转 get 传值

### 1. 引入 NavigationExtras

```
import { Router, NavigationExtras, ActivatedRoute } from '@angular/router';
```

### 2. 定义一个 goNewsContent 方法执行跳转，用 NavigationExtras 配置传参。

```
goNewsContent(){  
  
    let navigationExtras: NavigationExtras = {  
        queryParams: { 'session_id': '123' },  
        fragment: 'anchor'  
    };  
  
    this.router.navigate(['/news'], navigationExtras);  
  
}
```

### 3. 获取 get 传值

```
constructor(private route: ActivatedRoute) {  
    console.log(this.route.queryParams);  
}
```



## 八、父子路由

### 1. 创建组件引入组件

```
import { NewsaddComponent } from './components/newsadd/newsadd.component';  
import { NewslistComponent } from './components/newslist/newslist.component';
```

### 2. 配置路由

```
{  
  path: 'news',  
  component: NewsComponent,  
  
  children: [  
  
    {  
      path: 'newslist',  
  
      component: NewslistComponent  
    },  
    {  
      path: 'newsadd',  
  
      component: NewsaddComponent  
    }  
  ]  
}
```

### 3. 父组件中定义 router-outlet

```
<router-outlet></router-outlet>
```

