Angular 中的路由

主讲教师：（大地）
合作网站：www.itying.com （IT 营）
合作网站：www.ionic.wang （Ionic 中文网 ）

一、Angular 创建一个默认带路由的项目

1. 命令创建项目

```
ng new angualrdemo08 --skip-install
```


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、Angular routerLink 跳转页面 默认路由

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

```
//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'
}
```

三、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>
```

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

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

四、动态路由

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.跳转传值

```
<a [routerLink]="[ '/newscontent/',aid]">跳转到详情</a>
<a routerLink="/newscontent/{{aid}}">跳转到详情</a>
```

3.获取动态路由的值

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

```
  constructor(  private route: ActivatedRoute) {

  }
```

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

五、动态路由的 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]);
```

六、路由 get 传值 js 跳转

1. 引入 **NavigationExtras**

```
import { Router ,NavigationExtras} 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);
}
```

七、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>
```