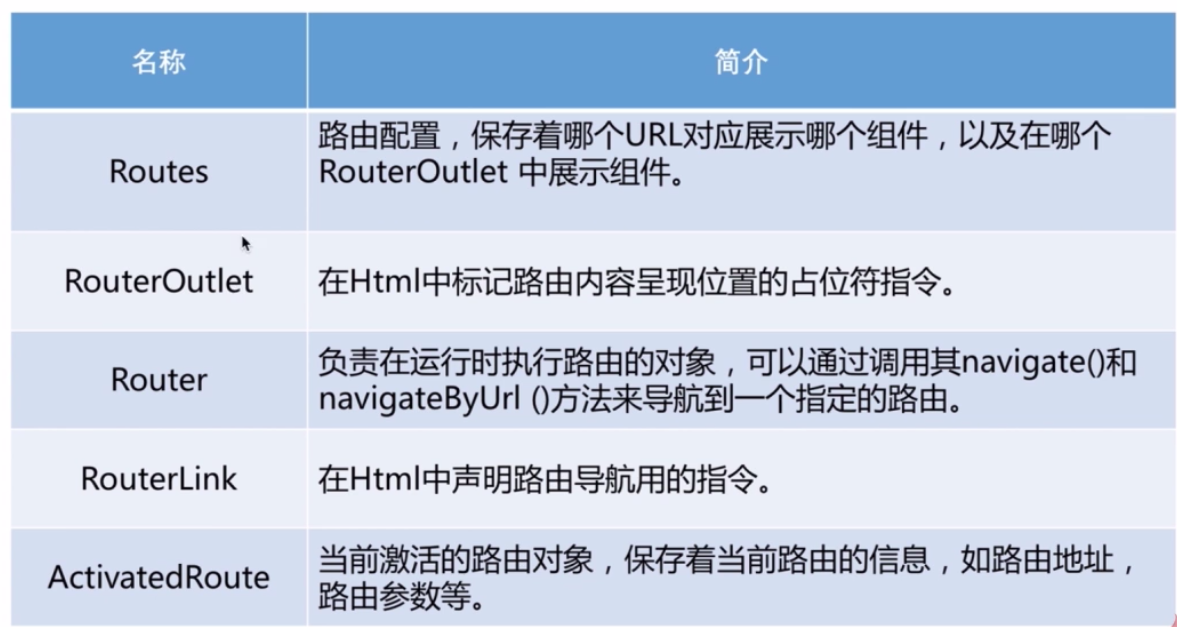
了解路由的基础知识

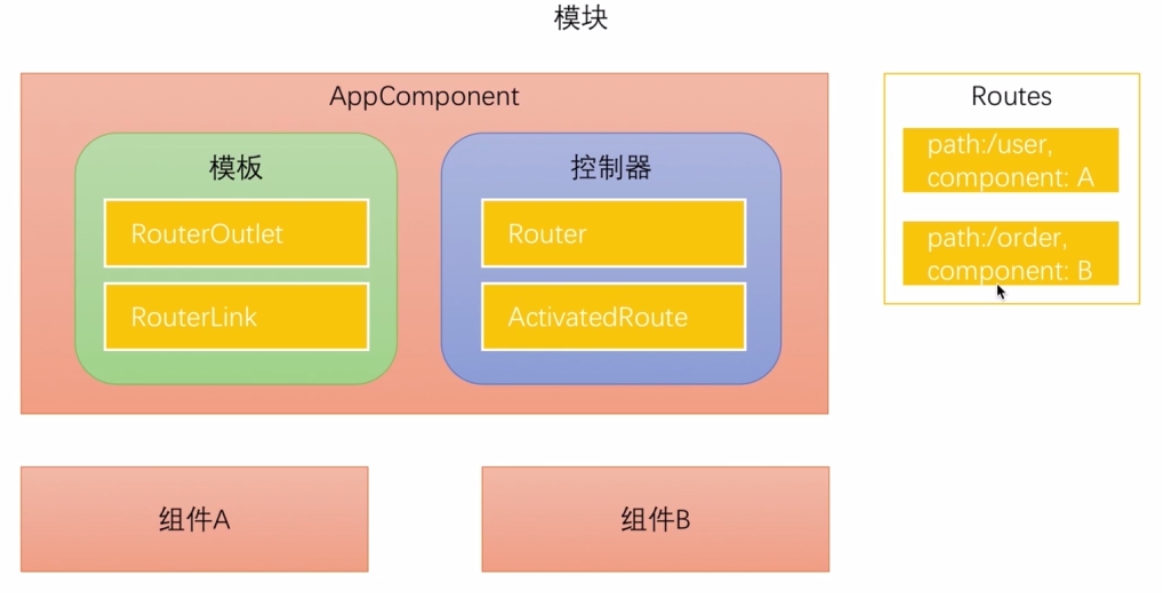
子路由, 保护路由和辅助路由

SPA [Single Page Application](http://kiwi.csie.chu.edu.tw/blog/archives/156" \l "toc-single-page-application)

1. 路由的基础知识

路由相关的对象





当地址是/user 是,展示A组件, 地址是/order, 展示B组件

**const** routes: Routes = [  
 {  
 **path**: **''**, **component** : HomeComponent  
 },  
 {  
 *// path里面千万不能/ 开头  
 // 路由对象路径 路由对象组件* **path**: **'product'**, **component** : ProductComponent  
 },  
 {  
 **path**: **'\*\*'**, **component** : Code404Component  
 },  
];

1. 重定向路由

用户访问一个特定的地址时, 将其重定向到另一个指定的地址

[www.aaa.com](http://www.aaa.com) => [www.aaa.com/products](http://www.aaa.com/products)

进行添加

{  
 path: '', redirectTo:'/home',pathMatch: 'full'  
},  
// 此时首页加载页面不存在  
{  
 path: 'home', component : HomeComponent  
},

重定向, 解决

改造股票项目

App.module.ts文件中

// 1.点击不同导航,显示不同页面  
**const** routeConfig: Routes = [  
 {path: '', redirectTo:'/dashboard',pathMatch: 'full'},  
 {path: 'dashboard', component: DashboardComponent},  
 {path: 'stock', component: StockmangeComponent}  
];

imports: [  
 BrowserModule,  
 FormsModule,  
 HttpModule,  
 // 2.将模块添加进来  
 RouterModule.*forRoot*(routeConfig)  
],

更改content.html

<!--3.使用插座替换-->  
<router-outlet></router-outlet>

更改menu,html

<!--4.更改menu-->  
<!--<li class="active"><a href="javascript:;" (click)="nav('dashboard')"><i class="fa fa-link"></i> <span>首页</span></a></li>-->  
<!--<li><a href="javascript:;" (click)="nav('stock')"><i class="fa fa-link"></i> <span>股票管理</span></a></li>-->

Menu.ts

// // // 5.设置nav方法  
// nav(url: string){  
// this.router.navigateByUrl(url);  
// }

constructor(**public** router: Router) { }

生成dashboard 模块, 重定向路由

**const** routeConfig: Routes = [  
 {path: '', redirectTo:'/dashboard',pathMatch: 'full'},  
 {path: 'dashboard', component: DashboardComponent},  
 {path: 'stock', component: StockmangeComponent}  
];

换种方法, 改变

Menu.ts文件中

// 6 设置menu类  
**export class** Menu {  
 **constructor**(  
 **public** id: **number**,  
 **public** name: **string**,  
 **public** link: **string** ){  
  
 }  
}

// 7.创建对象  
menus: Array<Menu>;

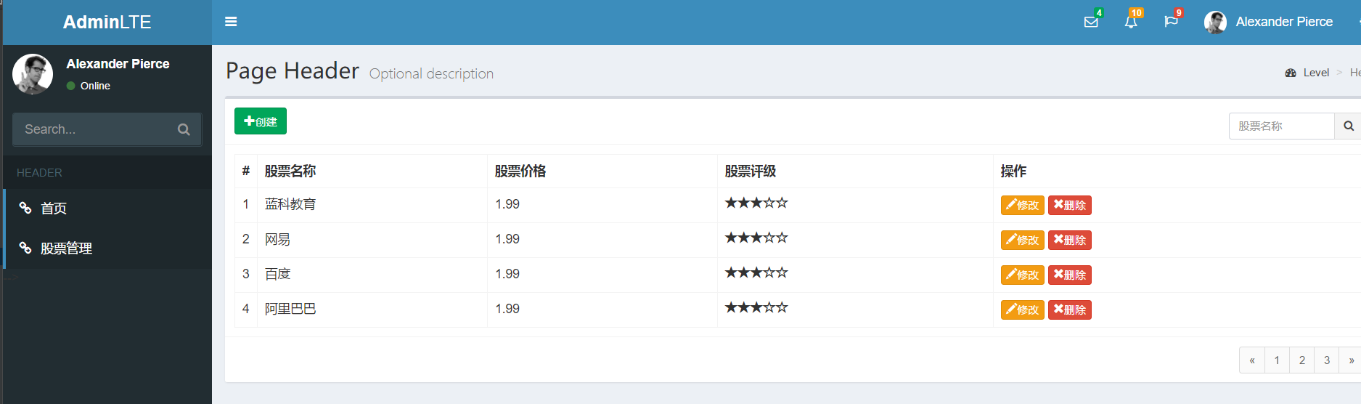
ngOnInit() {  
 //8.初始化对象  
 **this**.menus = [  
 **new** Menu(1,'首页','dashboard'),  
 **new** Menu(2,'股票管理','stock'),  
 ];  
}

menu.html文件中

<!--9.更改-->  
<li \*ngFor="let menu of menus" class="active"><a href="javascript:;" (click)="nav(menu)"><i class="fa fa-link"></i> <span>{{menu.name}}</span></a></li>-->

Menu.ts文件中

// 10 更改传递方法  
nav(menu: Menu) {  
 **this**.router.navigateByUrl(menu.link);  
}



现在首页和股票管理都处于激活, 处理激活样式

Menu.html绑定相应的样式

<!--9.更改-->  
<li \*ngFor="let menu of menus" [class.active]=""><a href="javascript:;" (click)="nav(menu)"><i class="fa fa-link"></i> <span>{{menu.name}}</span></a></li>-->

Menu.ts

// 11.定义页面id  
currentMenuId: **number**;

// 12. 传递id  
**this**.currentMenuId = menu.id;

<!--13.更改-->  
<li \*ngFor="let menu of menus" [class.active]="currentMenuId==menu.id"><a href="javascript:;" (click)="nav(menu)"><i class="fa fa-link"></i> <span>{{menu.name}}</span></a></li>-->

完成激活时的切换

更改页面文字

Content.html

<h1>  
 <!--14.更改页面文字-->  
 {{pageTitle}}  
 <small>{{pageDesc}}</small>  
</h1>

Content.ts

**export class** ContentComponent **implements** OnInit {  
 // 15.设置属性  
 pageTitle='';  
 pageDesc = '';  
 **constructor**(**public** router: Router) {  
 router.events.filter(event => event **instanceof** NavigationEnd).subscribe((event:NavigationEnd)=>{  
 **if**(event.url == '/dashboard'){  
 **this**.pageTitle ='这里是首页';  
 **this**.pageDesc = '';  
 }**else if** (event.url.startsWith('/stock')){  
 **this**.pageTitle = "股票信息管理";  
 **this**.pageDesc = "股票信息增删改查";  
 }  
 });  
 }

// 16.引入头文件  
**import** 'rxjs/add/operator/filter';

创建 + 修改按钮

ng g component stock/stockform

创建股票表单模块

Stockform.ts

**export class** StockformComponent **implements** OnInit {  
 // 17 创建对象  
 stock: Stock;  
 **constructor**() { }  
  
 ngOnInit() {  
 // 18.进行初始化  
 **this**.stock = **new** Stock(1,"蓝科教育",1.99,3.5,"蓝科教育的第一支股票",["IT","教育"]);  
 }  
  
}

stockform.html

<!--19.页面显示股票名称-->  
{{stock.name}}

Stockmanage.html

<!--20.创建按钮事件-->  
<a class="btn btn-success btn-sm" (click)="create()"> <span class="glyphicon glyphicon-plus" >创建</span></a>

<!--21. 修改按钮事件-->  
<a class="btn btn-warning btn-xs " (click)="update(stock)"><span class="glyphicon glyphicon-pencil"></span>修改</a>

Stockmanage.ts

// 22.实现方法  
create(){  
 **this**.router.navigateByUrl('/stock/0');  
}  
update(stock:Stock){  
 **this**.router.navigateByUrl('/stock/'+stock.id);  
}

// 23.更改路由  
**constructor**(**public** router: Router) { }