

大体流程：在app目录下创建独立的子模块 **Module**，在新创建的模块下创建组件，

写注册或登录的页面（各自组件的html中）

写各自的样式（各自组件的css中）

全局样式：写在src下的styles.css文件中

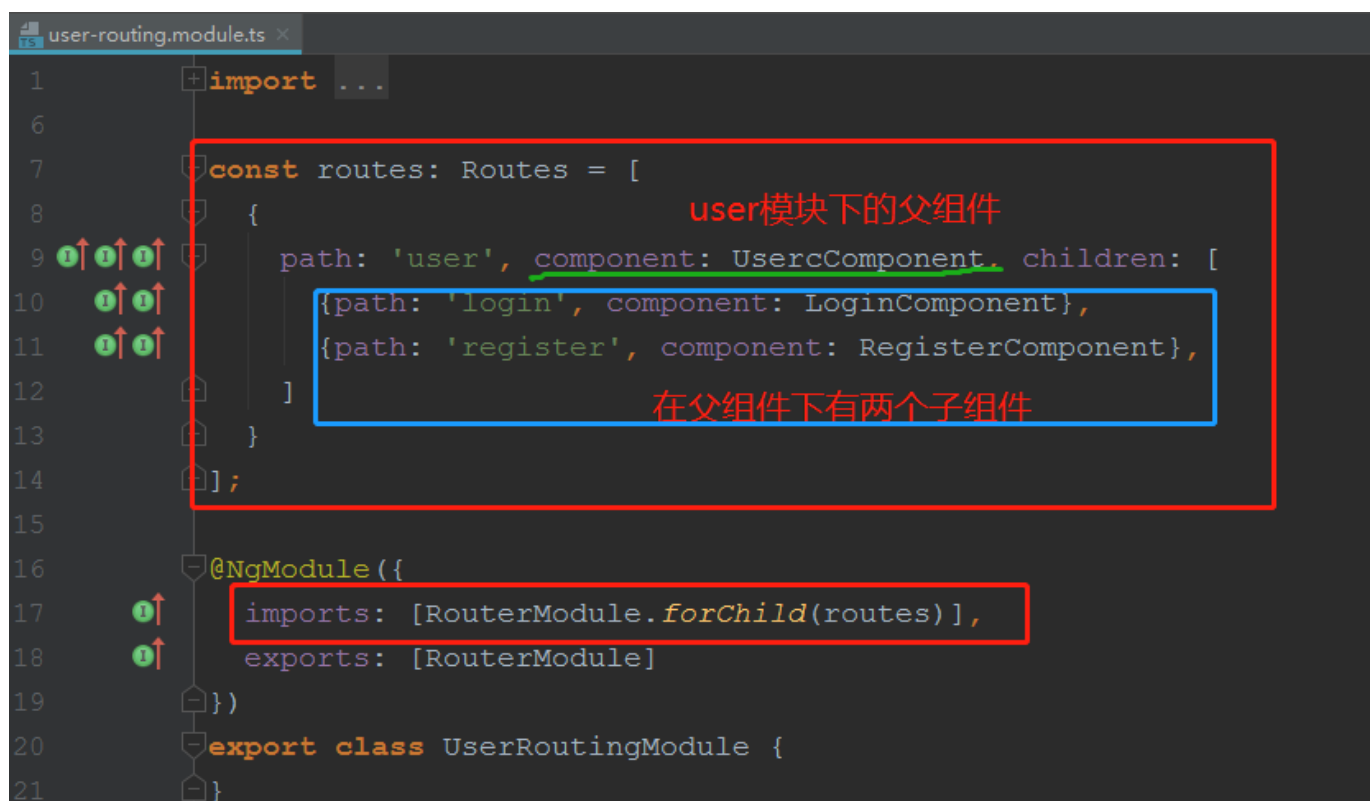
一、创建独立的子模块 **Module** 路由模块主要用于管理组件跳转的路径。

可视化选module 输入：模块名 --routing 如：user --routing

命令：ng g m 模块名 --routing 如：ng g m user --routing

--routing ： 的作用是创建模块的时候同时创建一个路由模块user-routing.module.ts，

定义模块路由：



```
1  import ...
6
7  const routes: Routes = [
8    {
9      path: 'user', component: UserComponent, children: [
10     {path: 'login', component: LoginComponent},
11     {path: 'register', component: RegisterComponent},
12   ]
13 }
14 ];
15
16 @NgModule({
17   imports: [RouterModule.forChild(routes)],
18   exports: [RouterModule]
19 })
20 export class UserRoutingModule {
21 }
```

Annotations in the image:

- Red box around lines 7-14: **user模块下的父组件**
- Blue box around lines 9-12: **在父组件下有两个子组件**
- Red box around lines 17-18: **imports: [RouterModule.forChild(routes)],**

将模块下生成的路由模块user-routing.module.ts

复制一份到app根模块下改名为 **app-routing.module.ts** 修改根路由模块的根路径及导出的名称：

```

1  import { NgModule } from '@angular/core';
2  import { Routes, RouterModule } from '@angular/router';
3  const routes: Routes = [
4      // 处理没有匹配的地址重定向到指定的页面      根路由，如果需要处理则处理
5      {path: '**', redirectTo: '/user/login'},
6      // 如果path为空时 要加一个pathMatch: 'full', 表示确定地址就是空后面什么都没有
7      {path: '', pathMatch: 'full', redirectTo: '/user/login'},
8  ];
9
10 @NgModule({
11     // 创建根路由
12     imports: [RouterModule.forRoot(routes)],      引入为根路由
13     exports: [RouterModule]
14 })
15 // 导出      改路由模块导出名称，这个名称将会在
16 export class AppRoutingModule { }      app.module.ts中引用

```

引入app根路由模块和新建子模块到app.module.ts根模块中

```

7  @NgModule({
8      // 组件注册
9      declarations: [
10         AppComponent,
11     ],
12      imports: [
13         BrowserModule,      导入新建模块
14         UserModule,
15         // 导入路由配置模块
16         AppRoutingModule      导入根路由模块
17     ],

```

二、创建新模块的组件：如先创建一个子模块叫user完成后在user模块下创建一个user的组件用于管理嵌套其它组件

三、在新模块下创建其它组件，如 登录、注册组件 参见第五篇笔记

四、新建模块的user.module.ts用于管理该模块下的所有组件的引用

```

user.module.ts x
8  import {HttpClientModule} from '@angular/common/http';
9
10 @NgModule({
11   declarations: [
12     UsercComponent,
13     LoginComponent, // 管理注册组件
14     RegisterComponent, // 管理登录组件
15   ],
16   imports: [
17     CommonModule,
18     FormsModule, // 导入注册组件所的表单模板驱动
19     HttpClientModule, // 导入注册组件所需要的HttpClient模块
20     UserRoutingModule // 导入user模块的路由模块
21   ]
22 })
23 export class UserModule { }

```

五、新建模块的user新组件的userc.component.html用于管理引入登录，注册 组件的路由

```

userc.component.html x
1
2  <!--路由导航链接-->
3  <div class="logo">
4    <a href="http://www.kaikeba.com/">
5      
6    </a>
7  </div>
8  <div class="nav">
9    <a routerLink="login" routerLinkActive="active">登录</a>
10   <a routerLink="register" routerLinkActive="active">注册</a>
11 </div>
12
13 <!--想要看到路由内容，需要放置路由出口-->
14 <router-outlet></router-outlet>

```

最后将**user**模块下所有组件要用到的组件，服务，类等等都放到**user**模块下进行管理

并在**user**模块中导入引用