



关注他

需求如标题所示，下面是详细步骤介绍。

首先，你需要确保你的环境中已经安装了Angular CLI。如果没有，可以通过以下命令安装：

```
npm install -g @angular/cli
```

然后你可以创建一个新的Angular项目:

```
ng new product-app
cd product-app
```

创建一个名为 `product` 的组件来显示产品列表：

ng generate component product

在 `product.component.html` 文件中，我们可以添加一个搜索框和一个产品列表：



```
<input [(ngModel)]="searchText" (input)="search()" placeholder="Search products">

<ul>
  <li *ngFor="let product of products" (click)="goToDetail(product.id)">
    {{product.name}}
  </li>
</ul>
```

在 `product.component.ts` 文件中, 我们需要定义 `searchText` 和 `products` 属性以及 `search` 和 `goToDetail` 方法:

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

@Component({
  selector: 'app-product',
  templateUrl: './product.component.html',
  styleUrls: ['./product.component.css']
})
export class ProductComponent {
  searchText = '';
  products = [];

  constructor(private router: Router) {}

  search() {
    // TODO: Call API to search products by name
    // This is just a mock
    this.products = [
      {id: 1, name: 'Product 1'},
      {id: 2, name: 'Product 2'},
      // ...
    ];
  }

  goToDetail(productId) {
    this.router.navigate(['/product', productId]);
  }
}
```

接下来, 我们需要为产品详情页面创建一个组件:

```
ng generate component product-detail
```

在 `product-detail.component.ts` 文件中, 我们需要获取路由参数中的产品ID, 并调用API来获取产品详情:

```
@Component({
  selector: 'app-product-detail',
  templateUrl: './product-detail.component.html',
  styleUrls: ['./product-detail.component.css']
})
export class ProductDetailComponent implements OnInit {
  product = null;

  constructor(private route: ActivatedRoute) {}

  ngOnInit() {
    const productId = this.route.snapshot.paramMap.get('id');
    // TODO: Call API to get product detail by id
    // This is just a mock
    this.product = {id: productId, name: 'Product ' + productId};
  }
}
```

在 product-detail.component.html 文件中, 我们可以显示产品详情:

```
<h1>{{product.name}}</h1>
<!-- More product details here -->
```

最后, 我们需要在 app-routing.module.ts 文件中定义路由:

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { ProductComponent } from '../product/product.component';
import { ProductDetailComponent } from './product-detail/product-detail.component';

const routes: Routes = [
  { path: 'product', component: ProductComponent },
  { path: 'product/:id', component: ProductDetailComponent },
  // Redirect to product list by default
  { path: '', redirectTo: '/product', pathMatch: 'full' },
];

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

这只是一个非常基础的示例。在实际项目中, 你可能需要用到更多的特性, 如表单验证, 错误处理, 状态管理等。



JerryWang
164 次咨询 5.0
电子科技大学 计算机系统结构硕士
16510 次赞同

去咨询 >

发布于 2023-07-07 22:32 · IP 属地上海

Angular JavaScript 产品规划



欢迎参与讨论



还没有评论，发表第一个评论吧

文章被以下专栏收录



Jerry的非SAP技术学习笔记

推荐阅读



如何优雅的使用 Angular 表单验证

why520crazy

打造自己的angular通用组件库

前言 在项目开发过程中，经常会抽出一些通用性的组件，在angular项目中通常放于shared-module目录下，这样在该项目就可以引入shared-module并使用。进一步地，如果我们希望shared-...

Angular 如何使用第三方库

Angular 的组件与模块看似好像与现有各种第三方类库（例如：lodash、moment 等）使用上有点格格不入，这很大的原因是TypeScript 造成的假象。三足鼎立的前端其实都是雷同的，不管是...

React Hooks工具库 aHooks 解析之usePersistFn

前言 aHooks 是阿里巴巴开源的一个React Hooks库，其中有很多hooks实现得很巧妙，一起来看看吧，本文的主角是usePersistFn。这里是 usePersistFn文档
iePersistFn 解决...

发表于Just ...

