

## [CGA-JAVA-WFDA] Web Front-end Development Angular 2.1

[Dashboard](#) / [My courses](#) / [CGA-JAVA-WFDA-2.1](#) / [7. Service và Router](#) / [\[Thực hành\] Xây dựng ứng dụng quản lý sản phẩm](#)

# [Thực hành] Xây dựng ứng dụng quản lý sản phẩm

## Mục tiêu

Sử dụng được service và router trong Angular

## Mô tả

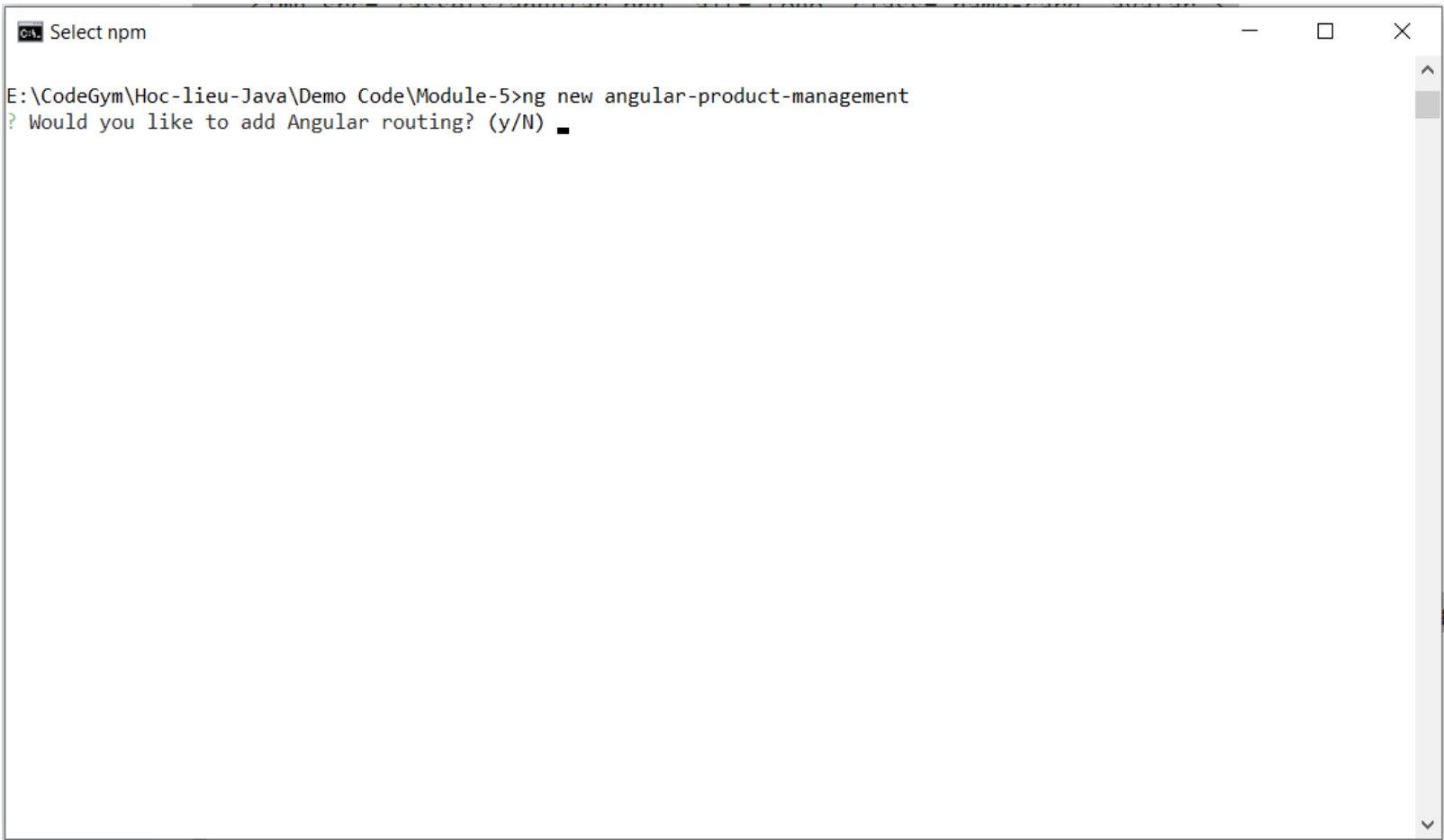
Xây dựng ứng dụng quản lý sản phẩm với 2 chức năng thêm và hiển thị sản phẩm.

Để tạo mới sản phẩm chúng ta sẽ tạo một component CreateProduct và để routes là 'products/create'

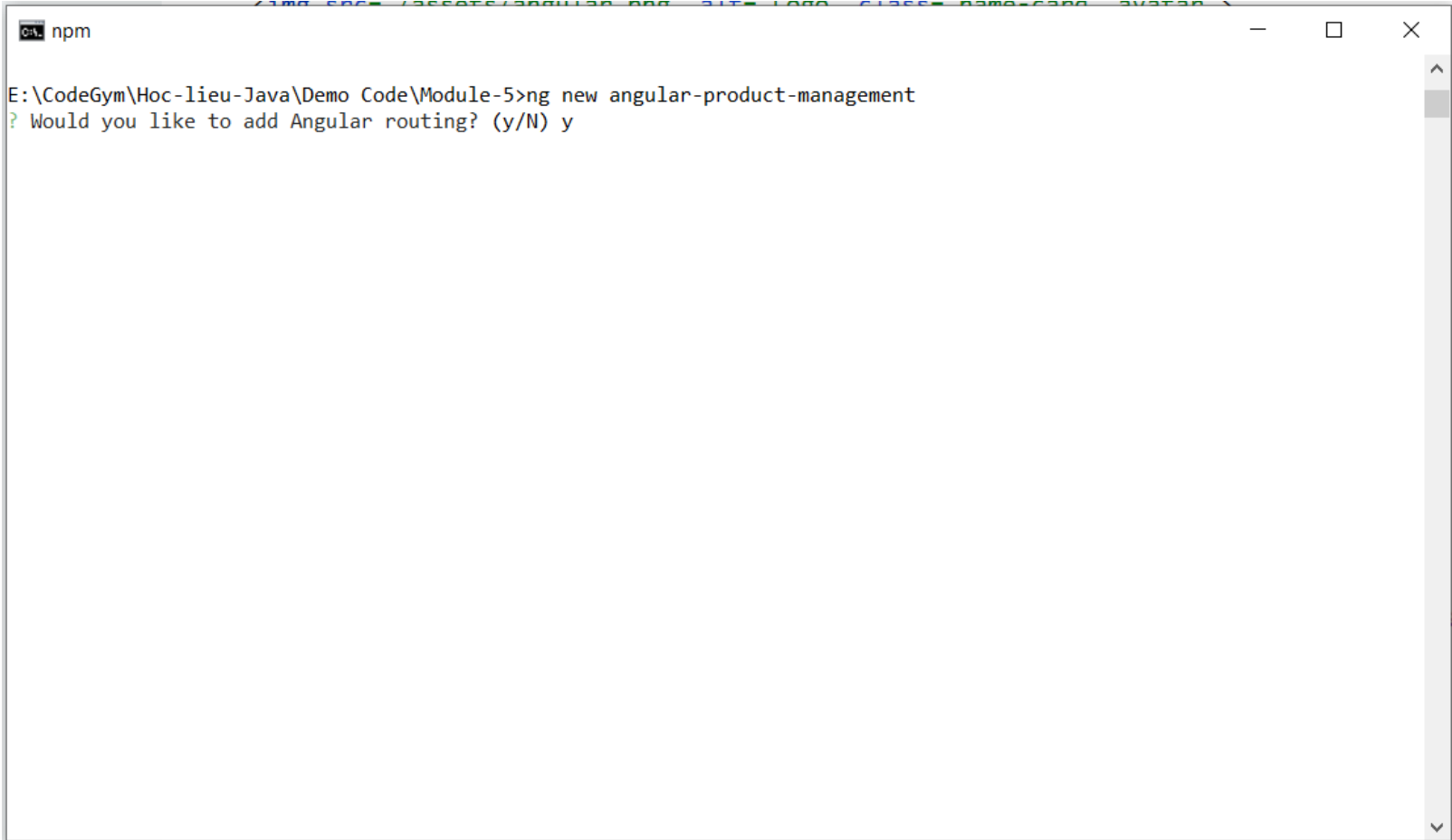
Để hiển thị danh sách sản phẩm chúng ta sẽ tạo component ListProduct và để routes là 'products'

## Hướng dẫn

Bước 1: Mở cửa sổ terminal và gõ lệnh **ng new angular-product-management** để tạo mới một dự án Angular



Tại cửa sổ tiếp theo gõ chữ **y** và ấn enter để tạo routing cho dự án



Tiếp theo chọn CSS và ấn enter và chờ ứng dụng được khởi tạo

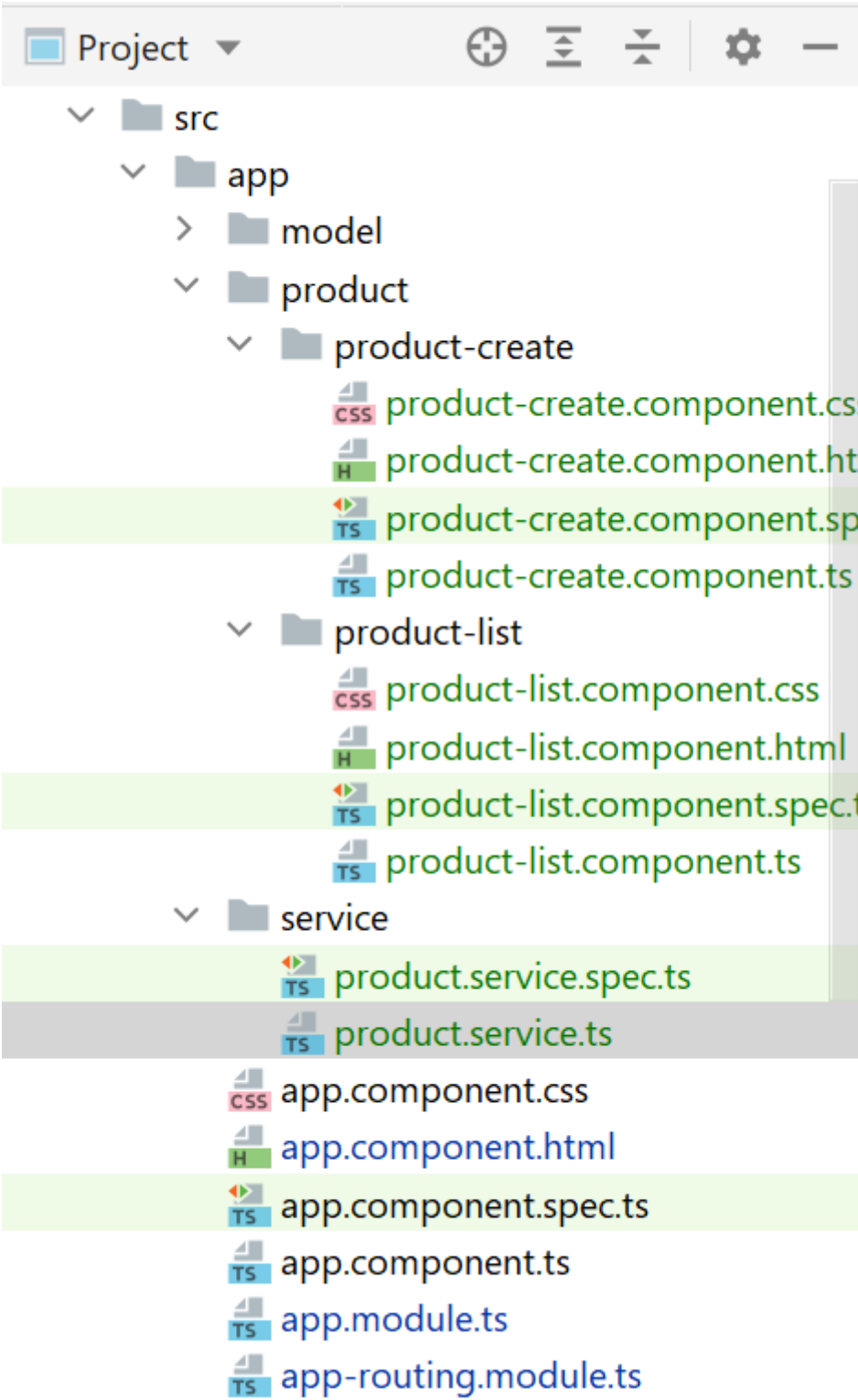
```
C:\Windows\System32\cmd.exe

The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/app.component.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/app.module.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/environments/environment.prod.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/environments/environment.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/index.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/main.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/polyfills.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/styles.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/test.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.app.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.spec.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tslint.json.
The file will have its original line endings in your working directory
Successfully initialized git.

E:\CodeGym\Hoc-lieu-Java\Demo Code\Module-5>
```

Vậy là dự án đã được tạo xong

Bước 2: Mở dự án vừa tạo bằng Webstorm và cấu trúc dự án sẽ như sau:



Bước 3: Mở cửa sổ terminal trong Webstorm và gõ lệnh **ng g i model/product** để tạo một interface mới trong thư mục model

Trong interface Product thêm đoạn code như sau để khai báo các thuộc tính:

```
export interface Product {
  id?: number;
  name?: string;
  price?: number;
  description?: number;
}
```

Bước 4: Gõ lệnh **ng g s service/product** để tạo ProductService

Bước 5: Trong file product-service.component.ts, khai báo một danh sách các sản phẩm:

```
products: Product[] = [{
  id: 1,
  name: 'IPhone 12',
  price: 2400000,
  description: 'New'
}, {
  id: 2,
  name: 'IPhone 11',
  price: 1560000,
  description: 'Like new'
}, {
  id: 3,
  name: 'IPhone X',
  price: 968000,
  description: '97%'
}, {
  id: 4,
  name: 'IPhone 8',
  price: 7540000,
  description: '98%'
}, {
  id: 5,
  name: 'IPhone 11 Pro',
  price: 1895000,
  description: 'Like new'
}];
```

Bước 6: Khai báo phương thức getAll() trong product-service.component.ts để lấy ra danh sách sản phẩm:

```
getAll() {
  return this.products;
}
```

Bước 7: Sử dụng lệnh **ng g c product/product-list** để tạo ProductListComponent trong thư mục product:

Trong file product-list.component.ts, khai báo một đối tượng productService trong constructor như sau:

```
constructor(private productService: ProductService) { }
```

Và khai báo một đối tượng products và một phương thức getAll() để lấy ra danh sách sản phẩm ở ProductService vừa khai báo ở trên và gọi phương thức getAll() trong ngOnInit:

```
import {Component, OnInit} from '@angular/core';
import {ProductService} from '../../service/product.service';
import {Product} from '../../model/product';

@Component({
  selector: 'app-product-list',
  templateUrl: './product-list.component.html',
  styleUrls: ['./product-list.component.css']
})
export class ProductListComponent implements OnInit {
  products: Product[] = [];

  constructor(private productService: ProductService) {
  }

  ngOnInit() {
    this.getAll();
  }

  getAll() {
    this.products = this.productService.getAll();
  }
}
```

Bước 8: Trong file product-list.component.html sử dụng đoạn template sau để hiển thị danh sách sản phẩm:

```
<h1>Danh sách sản phẩm</h1>
<table>
  <tr>
    <td>STT</td>
    <td>Tên</td>
    <td>Giá</td>
    <td>Mô tả</td>
  </tr>
  <tr *ngFor="let product of products; index as i">
    <td>{{i + 1}}</td>
    <td>{{product.name}}</td>
    <td>{{product.price}}</td>
    <td>{{product.description}}</td>
  </tr>
</table>
```

Bước 9: Cấu hình routing cho ProductListComponent:

Truy cập file app-routing-module.ts để cấu hình route truy cập đến ProductListComponent:

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

const routes: Routes = [{
  path: 'product/list',
  component: ProductListComponent
}];

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

```
  })  
  export class AppRoutingModuleModule { }
```

Trong file app.component.html sử dụng cặp thẻ <router-outlet></router-outlet> để hiển thị nội dung của component tương ứng và thêm đoạn code như sau:

```
<a routerLink="/product/list">Danh sách sản phẩm</a>  
<router-outlet></router-outlet>
```

Trong đó routerLink được sử dụng thay cho href để điều hướng sang trang product/list

Bước 10: Tạo phương thức saveProduct trong ProductService như sau:

```
saveProduct(product) {  
  this.products.push(product);  
}
```

Bước 11: Tạo component ProductCreateComponent bằng câu lệnh sau: **ng g c product/product-create**

Trong file product-create.component.ts khai báo một productForm và khai báo đối tượng productService như sau:

```
productForm: FormGroup = new FormGroup({  
  id: new FormControl(),  
  name: new FormControl(),  
  price: new FormControl(),  
  description: new FormControl(),  
});  
constructor(private productService: ProductService) { }
```

Và xây dựng một phương thức submit để thêm sản phẩm mới vào danh sách sản phẩm đang có:

```
submit() {  
  const product = this.productForm.value;  
  this.productService.saveProduct(product);  
  this.productForm.reset();  
}
```

Sau đó trong file product-create.component.html sử dụng đoạn template như sau để tạo một form thêm mới sản phẩm:

```
<h1>Tạo sản phẩm mới</h1>  
<form [formGroup]="productForm" (ngSubmit)="submit()">  
  <table>  
    <tr>  
      <td>Mã sản phẩm:</td>  
      <td><input type="number" formControlName="id"></td>  
    </tr>  
    <tr>  
      <td>Tên sản phẩm:</td>  
      <td><input type="text" formControlName="name"></td>  
    </tr>  
    <tr>  
      <td>Giá sản phẩm</td>  
      <td><input type="number" formControlName="price"></td>  
    </tr>  
    <tr>  
      <td>Mô tả về sản phẩm</td>  
      <td><input type="text" formControlName="description"></td>  
    </tr>  
    <tr>  
      <td></td>  
      <td><button>Tạo mới</button></td>  
    </tr>  
  </table>  
</form>
```

Bước 12: Truy cập lại vào file app-routing-module.ts để khai báo thêm routes tương ứng với ProductCreateComponent:

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { ProductListComponent } from './product/product-list/product-list.component';
import { ProductCreateComponent } from './product/product-create/product-create.component';

const routes: Routes = [{
  path: 'product/list',
  component: ProductListComponent
}, {
  path: 'product/create',
  component: ProductCreateComponent
}];

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

Và trong file app.component.html bổ sung đoạn template như sau:

```
<a routerLink="/product/list">Danh sách sản phẩm</a>|<a routerLink="/product/create">Tạo mới sản phẩm</a>
<router-outlet></router-outlet>
```

Giao diện thêm mới sản phẩm

<

>

↺

ⓘ

http://localhost:4200/product/create

Danh sách sản phẩm

Tạo mới sản phẩm

Tạo sản phẩm mới

Mã sản phẩm:

6

Tên sản phẩm:

IPhone 7

Giá sản phẩm

540000

Mô tả về sản phẩm

99%

Tạo mới

Truy cập lại vào đường dẫn /product/list để xem danh sách sản phẩm lúc này:

[Danh sách sản phẩm](#)[|Tạo mới sản phẩm](#)

# Danh sách sản phẩm

STT	Tên	Giá	Mô tả
1	IPhone 12	2400000	New
2	IPhone 11	1560000	Like new
3	IPhone X	968000	97%
4	IPhone 8	7540000	98%
5	IPhone 11 Pro	1895000	Like new
6	IPhone 7	595000	99%

Checkout sang nhánh service-router để xem mã nguồn tham khảo: <https://github.com/codegym-vn/wfda-angular-product-management>

## Hướng dẫn nộp bài:

up bài tập lên github

paste link github vào phần nộp bài

### Submission status

Submission status	No attempt
Grading status	Not graded
Last modified	-
Submission comments	<div><div></div><div><a href="#">Comments(0)</a></div></div>

Add submission

You have not made a submission yet.

#### CHƯƠNG TRÌNH

- Career
- Premium
- Accelerator

#### TÀI NGUYÊN

- Blog
- Tạp chí Lập trình
- AgileBreakfast

Follow Us



CodeGym@2018. All rights reserved.

You are logged in as Dương Văn Thanh Sơn (Log out)