FPT Education

FPT POLYTECHNIC

cdio™

Conceive Design Implement Operate

THỰC HỌC – THỰC NGHIỆP

LẬP TRÌNH FRONT-END FRAMEWORK 1

ROUTING

http://www.poly.edu.vn

Mục tiêu

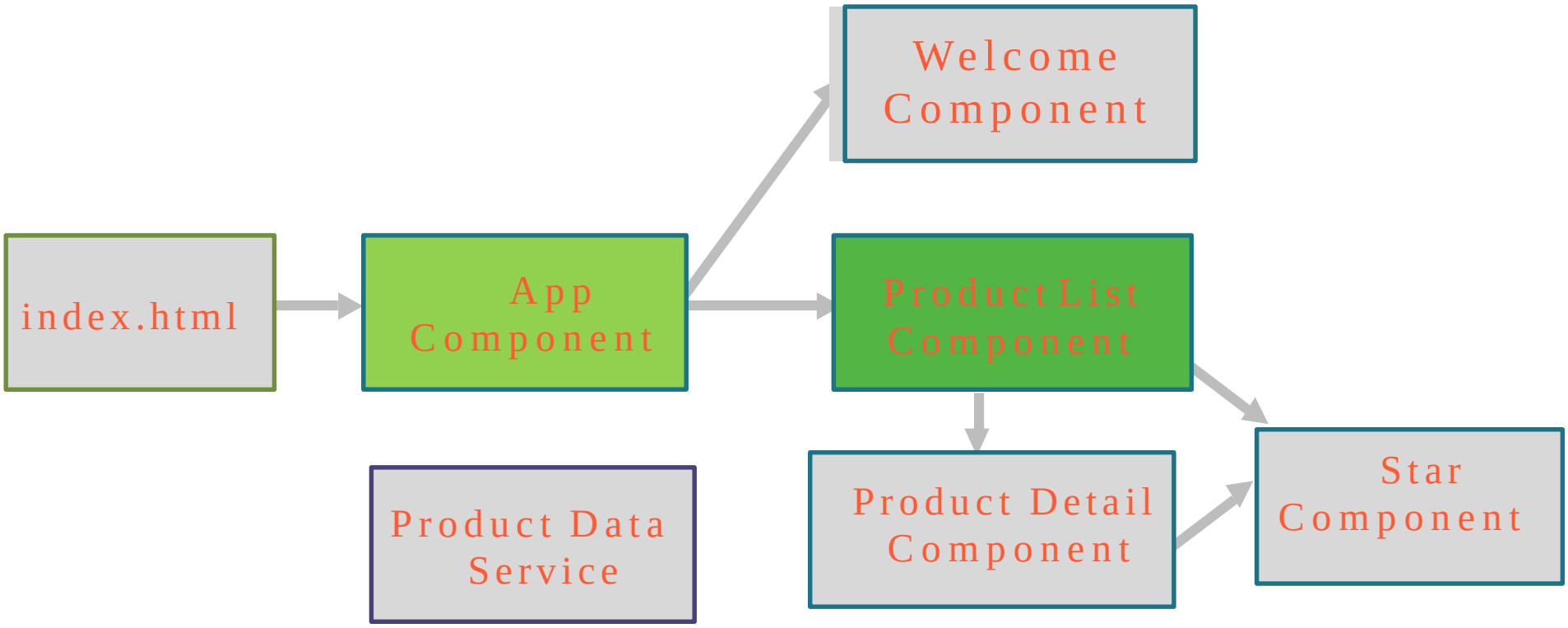
- Hiểu và cài đặt Routing
- Truyền và nhận tham số trong Router
- Bảo vệ route, điều khiển điều hướng với canActivate

NỘI DUNG

- Routing là gì? Sự cần thiết của nó
- Cài đặt routing
- Định dạng router
- Xử lý tham số trong routing
- Bảo vệ route với canActivate

Phần 1: Binding

KIẾN TRÚC CỦA ANGULAR



ROUTING LÀ GÌ?

- Routing là một tính năng cho phép chúng ta điều hướng các URL tới các trang nào đó trong ứng dụng
- Đây là tính năng có trong hầu hết các web framework phổ biến ngày nay.

Trong Angular thì một URL sẽ được điều hướng tới một Component, tức là khi người dùng trỏ URL nào vào trong trình duyệt thì Angular sẽ hiển thị template của lớp Component được điều hướng tương ứng.

TẠI SAO CẦN TỚI ROUTER?

- Router giúp người dùng dễ dàng chuyển tới các trang trong ứng dụng

CÀI ĐẶT ROUTING

Tạo module router: Trong Angular, tốt nhất ta nên tách biệt chức năng cấu hình và cài đặt routing thành module riêng rẽ

ng generate module app-routing --flat --module=app

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

const routes: Routes = [];

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

CÀI ĐẶT ROUTING

Định nghĩa các url cho ứng dụng tại module app-routing

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { HomeComponent } from './home/home.component';
import { ProductlistComponent } from './productlist/productlist.component';

const routes: Routes = [
{path:'',component:HomeComponent},
{path:'home',component:HomeComponent},
{path:'products',component:ProductlistComponent}
];

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

CÀI ĐẶT ROUTING

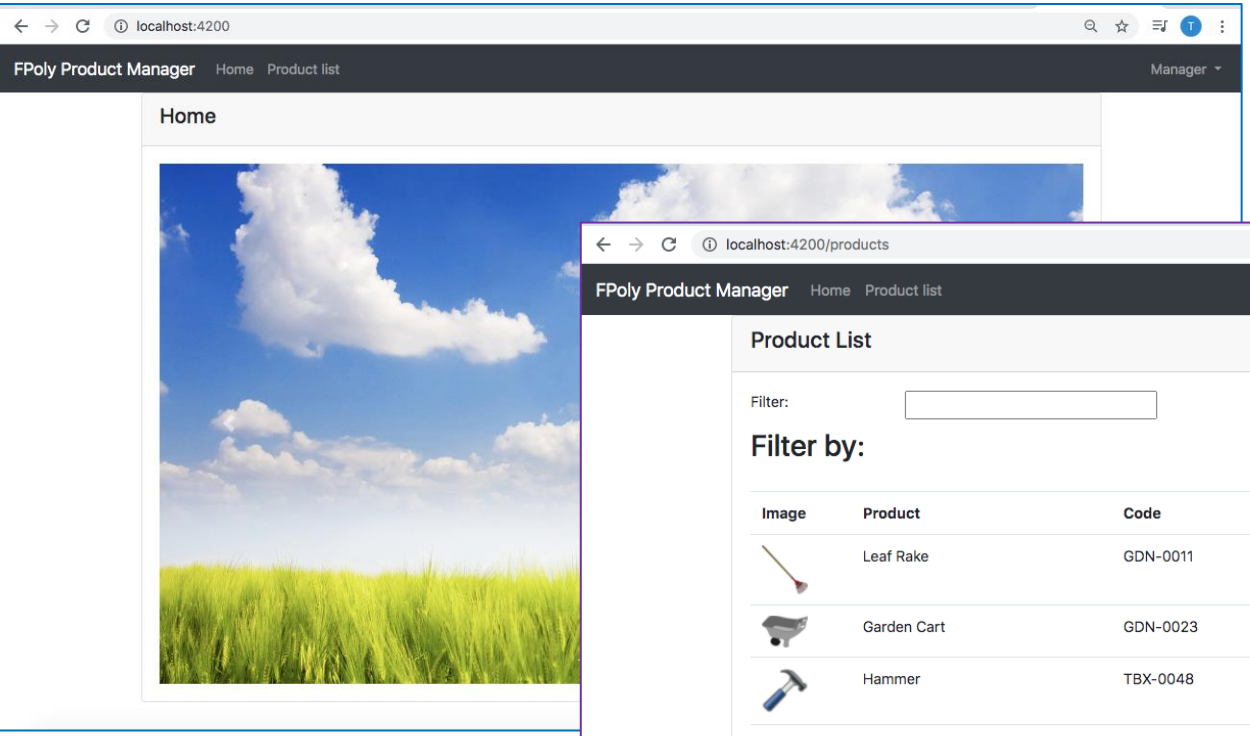
Tại app component khai báo vị trí load component tương ứng với path đã tạo

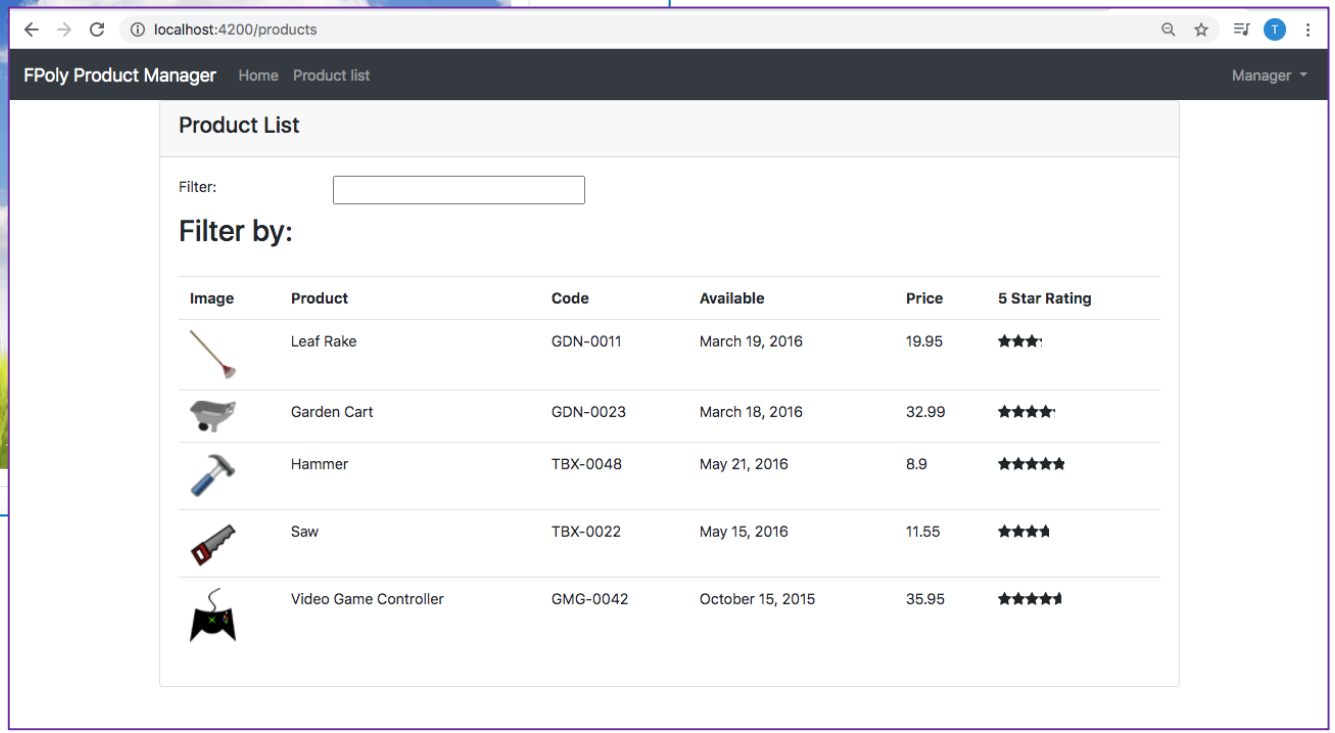
app.component.html

```
<app-header></app-header>
<div class="container">
  <router-outlet></router-outlet>
</div>
```

CÀI ĐẶT ROUTING

Star server và nhập url tương ứng với router đã cài đặt





ĐIỀU HƯỚNG (CHUYỂN TRANG)

Thực hiện liên kết với thẻ a và thuộc tính href

```
<a class="navbar-brand" href="">FPoly Product Manager</a>
<!-- Links -->
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link" href="home">Home</a>
</li>
<li class="nav-item">
<a class="nav-link" href="products">Product list</a>
</li>
</ul>
```

ĐIỀU HƯỚNG (CHUYỂN TRANG)

Thực hiện liên kết router link

```
<!-- Brand -->
<a class="navbar-brand" routerLink="/">FPoly Product Manager</a>
<!-- Links -->
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link" routerLink="/home">Home</a>
</li>
<li class="nav-item">
<a class="nav-link" routerLink="products">Product list</a>
</li>
</ul>
```

THỰC HIỆN ROUTER

PHẦN 2: QUẢN LÝ THAM SỐ

TRUYỀN THAM SỐ TỚI ROUTE

http://localhost:4200/product/2

Khi nhận request với url như trên, ứng dụng angular sẽ tìm route tương ứng:

{path:'product/:id',**component:ProductdetailComponent**}

TRUYỀN THAM SỐ TỚI ROUTE

http://localhost:4200/product/2

Khi nhận request với url như trên, ứng dung angular sẽ tìm route tương ứng:

{path:'product/:id',**component:ProductdetailComponent**}

TRUYỀN THAM SỐ TỚI ROUTE

Nhận tham số

Sử dụng lớp ActivateRoute:

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

constructor(private route:ActivatedRoute) { }
```

Đọc giá trị tham số:

```
let id=this.route.snapshot.params['id'];
```

QUERY PARAMETER VÀ FRAGMENT

Angular hổ trợ request có dạng: url/edit-product/2/edit?alowedit=1#loading

Sử dung query paramas và fragment để tạo link:

```
<a
    href=""
    [routerLink]="['/edit-product',2,'edit']"
    [queryParams]="{allowEdit:'1'}"
    fragment="loading">
    Chỉnh Sửa
</a>
```

QUERY PARAMETER VÀ FRAGMENT

Truy xuất tới query paramas và fragment

Sử dụng đối tượng ActivateRoute để truy xuất, cú pháp:

```
constructor(private route:ActivatedRoute) { }

ngOnInit(): void {
  console.log(this.route.snapshot.queryParams)
  console.log(this.route.snapshot.fragment);
}
```

FPT Education
FPT POLYTECHNIC

ROUTE CON – CHILD ROUTE

Thường sử dụng để gom nhóm router, ví dụ như:

```
const routes: Routes = [
{path:'',component:HomeComponent},
{path:'home',component:HomeComponent},
{path:'products',component:ProductlistComponent},
{path:'products/:id/edit',component:EditproductComponent},
{path:'products/:id',component:ProductdetailComponent}
];
```

```
const routes: Routes = [
{path:'',component:HomeComponent},
{path:'home',component:HomeComponent},
{path:'products',component:ProductlistComponent, children:[
  {path:':id/edit',component:EditproductComponent},
  {path:':id',component:ProductdetailComponent}
]}
];
```

CHÚ Ý

Router xác định không tồn tại url, ta định nghĩa như sau

```
{ path: '**', component: PageNotFoundComponent }
```

Router định nghĩa component mặc định

```
{ path: '', redirectTo: '/home', pathMatch: 'full' },
```

BẢO VỆ ROUTE VỚI CANACTIVATE

- Hiện tại, bất kỳ người dùng nào cũng có thể điều hướng đến bất kỳ đâu trong ứng dụng bất kỳ lúc nào.
- Đôi khi bạn cần kiểm soát quyền truy cập vào các phần khác nhau của ứng dụng vì nhiều lý do khác nhau, như:
  - Có lẽ người dùng không được phép điều hướng đến component.
  - Có thể người dùng phải đăng nhập (xác thực) trước.
  - Có lẽ bạn cần số dữ liệu trước khi hiển thị component.

BẢO VỆ ROUTE VỚI CANACTIVATE

Tạo class authservice

```
export class AuthService {
loggedIn = false;
isAuthenticated() {
  const promise = new Promise(
   (resolve, reject) => {
       setTimeout(() => {
          resolve(this.loggedIn);
        }, 800);
    }
  );
  return promise;
}
login() {
  this.loggedIn = true;
}
logout() {
  this.loggedIn = false;
}
}
```

BẢO VỆ ROUTE VỚI CANACTIVATE

Tạo class bảo vệ với canActivate

```
export class AuthGuard implements CanActivate, CanActivateChild {
constructor(private authService: AuthService, private router: Router)
{}

canActivate(route: ActivatedRouteSnapshot,
state: RouterStateSnapshot): Observable<boolean> |
Promise<boolean> | boolean {
  return this.authService.isAuthenticated()
          .then((authenticated: boolean) => {
                  if (authenticated) {
                     return true;
                  } else {
                      this.router.navigate(['/']);
                  }
              });
}
canActivateChild(route: ActivatedRouteSnapshot,
state: RouterStateSnapshot): Observable<boolean> |
Promise<boolean> | boolean {
    return this.canActivate(route, state);
}
}
```

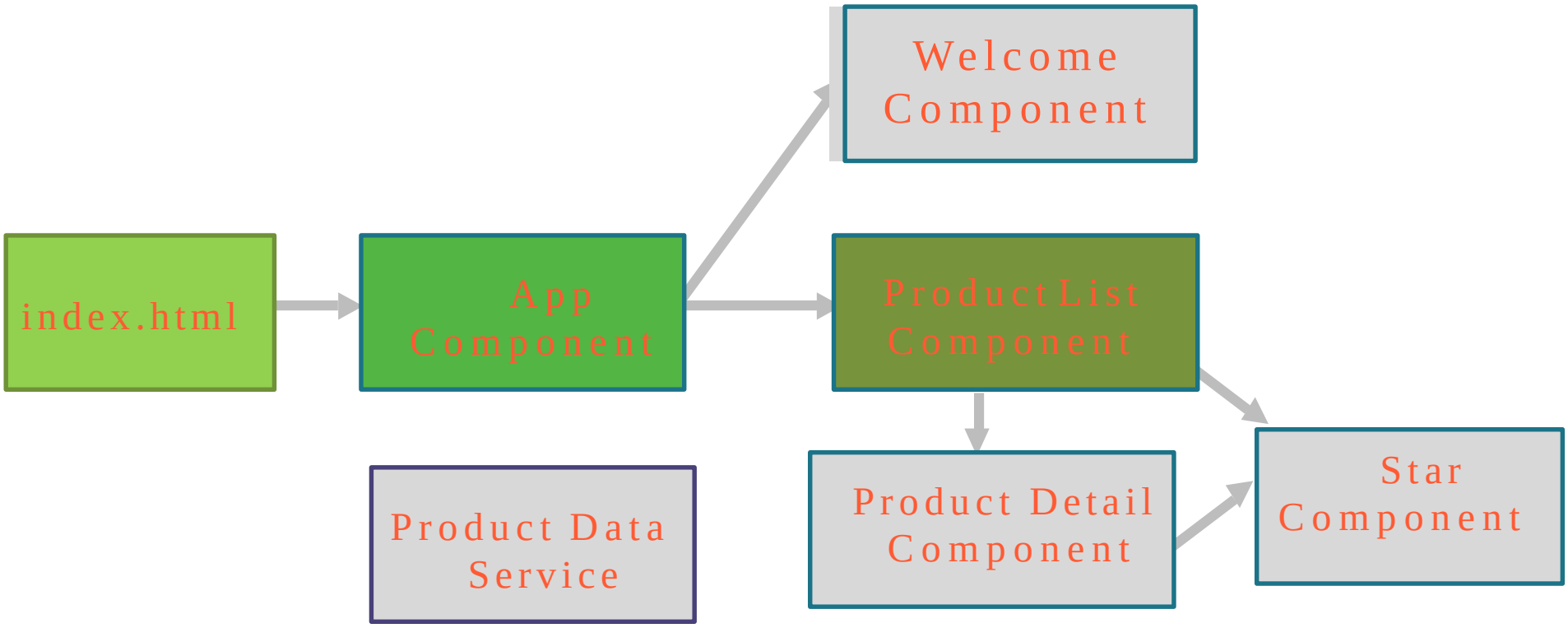
BẢO VỆ ROUTE VỚI CANACTIVATE

Thêm AuthGuard vào phần addmin để bảo vệ các route con

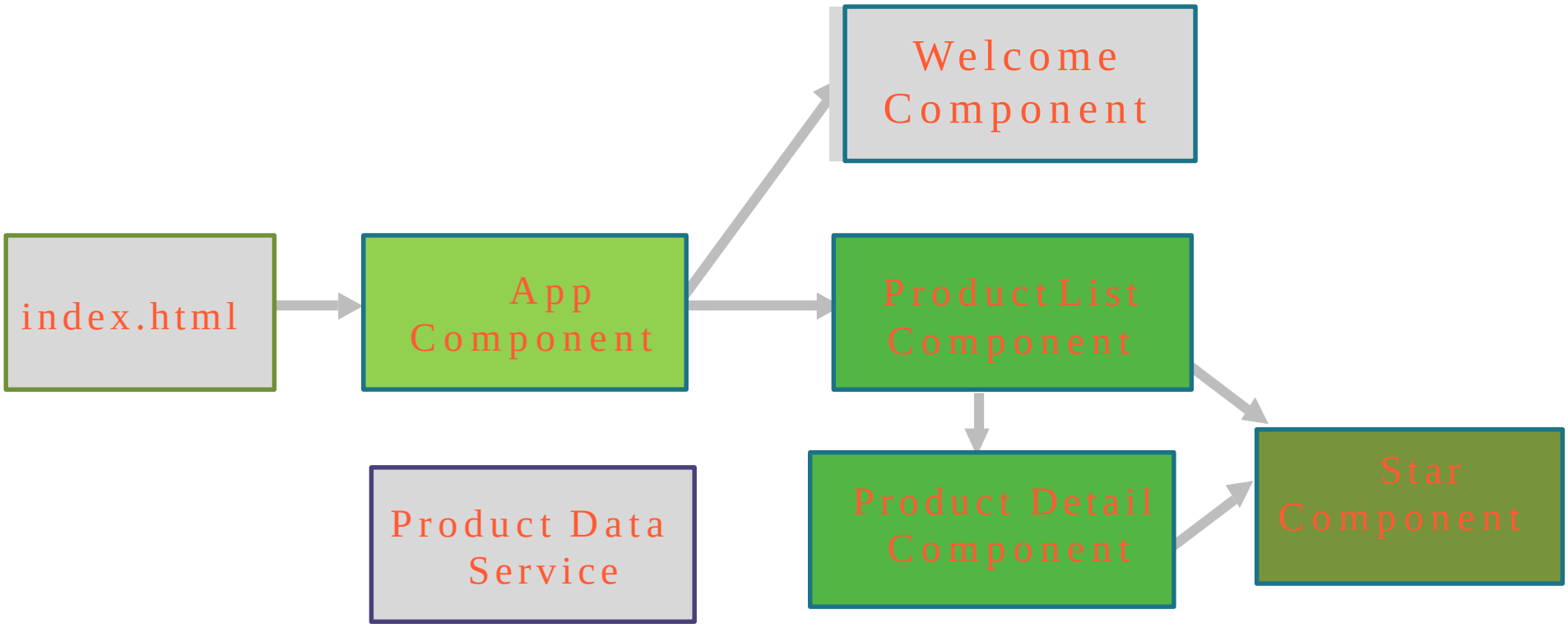
```
const adminRoutes: Routes = [
{ path: 'admin',
 component: AdminComponent,
 canActivate: [AuthGuard],
 children: [
   { path: '', canActivateChild: [AuthGuard], children: [
     { path: products', component: ProductsComponent },
     { path: catalogs', component: CatalogsComponent },
     { path: '', component: AdminDashboardComponent }
   ]
  }
 ]
}
];
```

XỬ LÝ THAM SỐ
BẢO VỆ ROUTE

KIẾN TRÚC CỦA ANGULAR



KIẾN TRÚC CỦA ANGULAR



SUMARRY

- ☑ Routing là gì? Sự cần thiết của nó
- ☑ Cài đặt routing
- ☑ Định dạng router
- ☑ Xử lý tham số trong routing
- ☑ Bảo vệ route với canActivate


thank
you!