



FPT POLYTECHNIC



THỰC HỌC – THỰC NGHIỆP



Conceive Design Implement Operate

LẬP TRÌNH FRONT-END FRAMEWORK 1

DATABINDING

- ⊙ Biết sử dụng module core và share module
- ⊙ Hiểu và cài đặt Lazy Loading
- ⊙ Thực hiện deploy ứng dụng



📖 Biết sử dụng module core và share module

📖 Cài đặt Feature module

📖 Hiểu và cài đặt Lazy Loading

📖 Debugging ứng dụng

📖 Thực hiện deploy ứng dụng





PHẦN 1: MODULE

Module là gì?

AppModule

Angular dùng NgModule để quản lý ứng dụng và các tính năng của ứng dụng

AppComponent

Angular Module định nghĩa tất cả các thành phần được dùng trong ứng dụng như: component, Directive, service

ProductComponent

Một ứng dụng yêu cầu duy nhất 1 module, nhưng có thể tách thành nhiều module

Service

Core Angular module là các module đã có sẵn (như FormsModule) để dùng khi cần

Tất cả các thành phần trong ứng dụng phải được thêm (khai báo) trong một module

Các Module đặc trưng

Module được tạo sẵn

Là các module trong ứng dụng, được tạo ra để thực hiện tính năng nào đó

Lazy loaded module

Là module thường được sử dụng để cải thiện hiệu năng và trải nghiệm của người dùng

Routing module

Module này dùng để định tiến
Thêm guard

Shared module

Là tập các directive hay pipe

Cách sử dụng Module được xây dựng sẵn

Thường là các module được bên thứ ba xây dựng sẵn, nhằm thực hiện một chức năng nào đó mà trong ứng dụng ta cần

1. Các bước sử dụng module:
2. Cài đặt module
3. Import module vào `appModule`
4. Sử dụng

Cách sử dụng Module được xây dựng sẵn, module phân trang

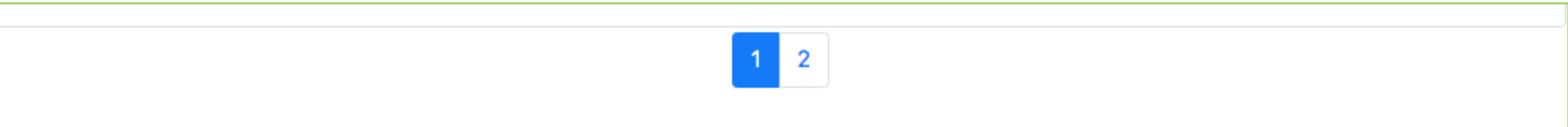
```
npm install ngx-pagination --save
```

```
// app.module.ts
```

```
import {NgxPaginationModule} from 'ngx-pagination'
```

```
<ul>  
  <li *ngFor="let item of collection | paginate: { itemsPerPage: 10, currentPage: p }"> ... </li>  
</ul>
```


Cách sử dụng Module được xây dựng sẵn, module phân trang – tạo nút chuyển trang với class bootstrap



```
<div class="row">
  <div class="col-12">
    <ul class="pagination justify-content-center mt-1">
      <li class="page-item" *ngFor="let page of pages"
        (click)="onPageIndexClicked(page.pageIndex)" [ngClass]=" { 'active':
        page.pageIndex == currentPageIndex}">
        <a href="#" onclick="return false" class="page-link">
          {{page.pageIndex}}
        </a>
      </li>
    </ul>
  </div>
</div>
```

Là dạng module do lập trình viên tự tạo

Gom các component hoặc service có liên quan đến nhau
hoặc cùng nằm trong một feature nào đó thành một nhóm

Tạo module, ví dụ như tạo module có tên Admin

ng generate module **Admin**

Cài đặt và sử dụng feature module

Thêm module vừa tạo vào appModule

App.Module.ts

```
import { AdminModule } from './admin/admin.module';  
imports: [  
  BrowserModule,  
  AdminModule,  
  AppRoutingModule,  
  FormsModule,  
  NgxPaginationModule,  
]
```

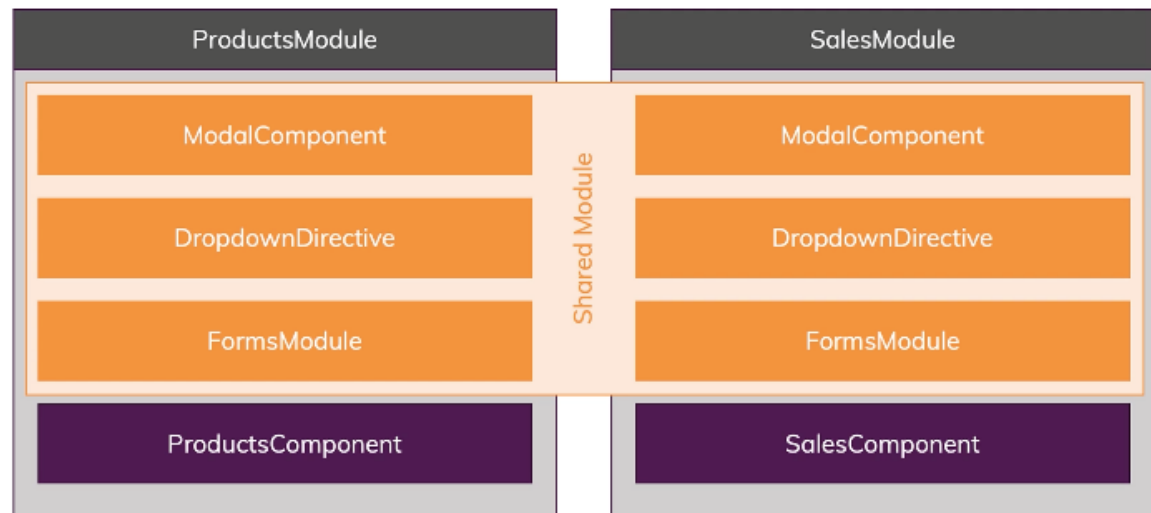
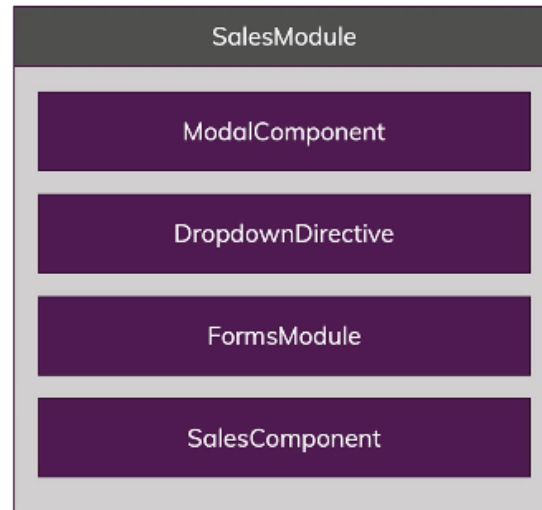
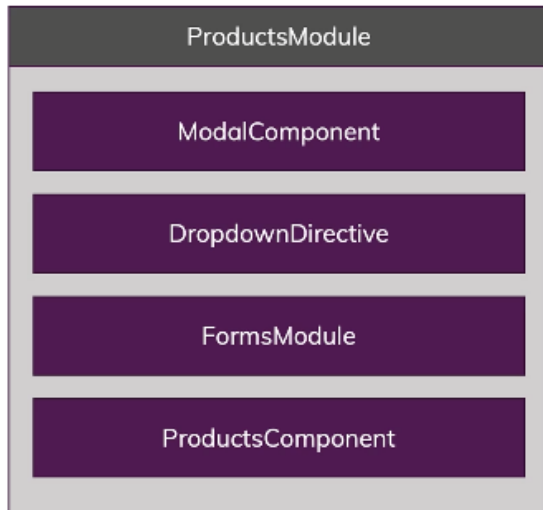
Cài đặt và sử dụng feature module

Thêm routing module cho feature module

```
const routes: Routes = [{
  path:'admin',component:DashboardComponent,
  children:[{path:'products',component:ProductsComponent},
    {path:'products/add',component:AddnewComponent},
    {path:'products/:id/edit',component>EditComponent},
  ]});
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule])
  export class AdminRoutingModule { }
```

Admin-routing.module.ts

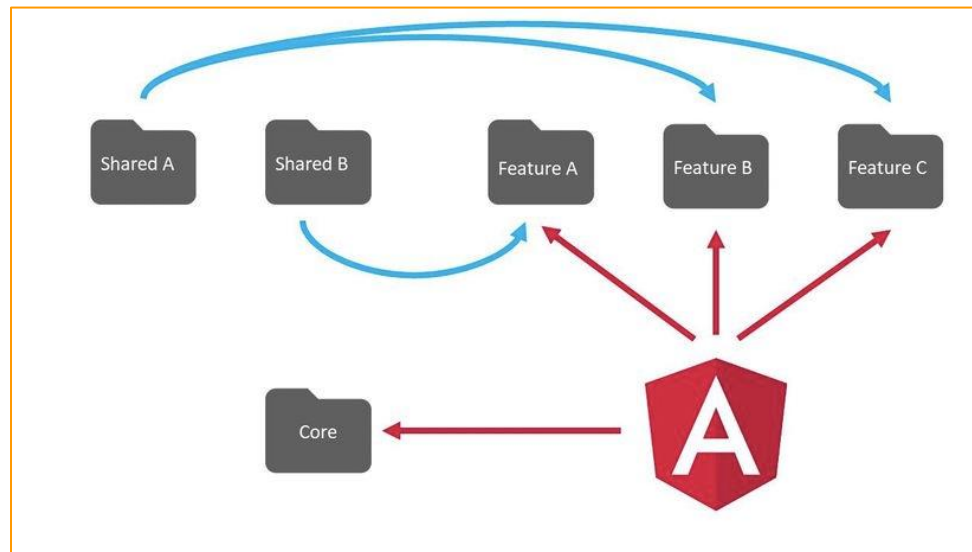
Shared module



Shared module

Thực hiện:

1. Tìm những module, directive, pip dùng chung cho các module
2. Tạo shared module chứa các module, directive, pipe dùng chung trên
3. Sử dụng share module cho các module cần dùng





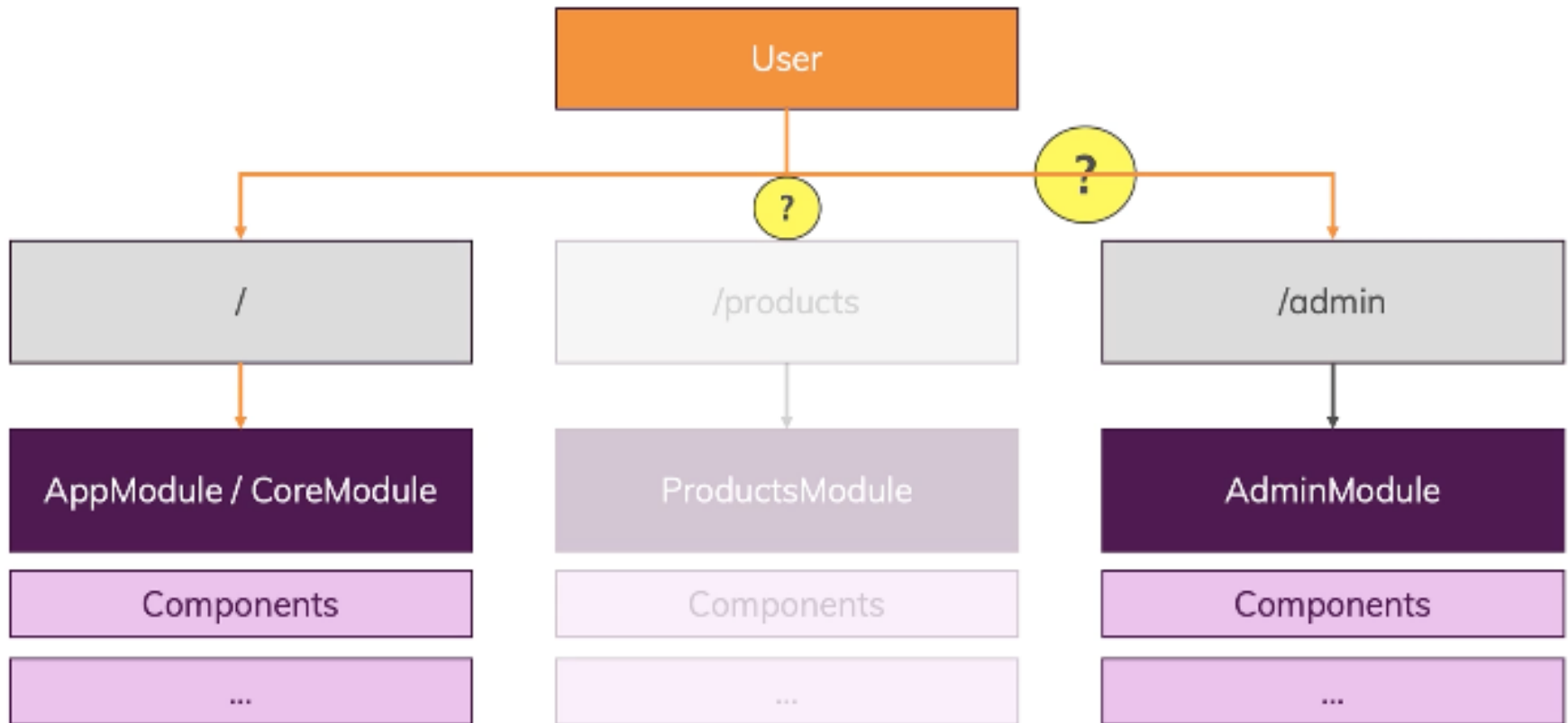
DOMAIN MODULE
SHARED MODULE



PHẦN 2: LAZY LOADED MODULE

- Việc load tất cả các module cùng lúc khi tải trang sẽ có thể gây chậm trễ từ đó làm giảm trải nghiệm người dùng.
- Việc viết tất cả các route vào AppRoutingModuleModule cũng sẽ gây khó trong quá trình phát triển file quá dài cũng gây ức chế trong quá trình phát triển.
- Vì vậy thường khi chia các FeatureModule rồi thì chúng ta cũng chia route đi kèm các FeatureModule đó.

LAZY LOADED MODULE



Cài đặt lazy loaded module

- Để dùng lazy loading, ta sử dụng `loadChildren` trong `AppRoutingModule` như sau:

AppRoutingModule

```
const routes: Routes = [  
  {  
    path: 'admin',  
    loadChildren: () => import('./admin/admin.module').then(m => m.AdminModule)  
  },  
]
```

Cài đặt lazy loaded module

- Tại route của lazy module, ta sử dụng thay đổi cấu hình:

AppRoutingModule

```
{
  path:'',component:LayoutComponent,
  children:[
    {path:'',component:DashboardComponent},
    {path:'products',component:ProductsComponent},
  ],
},
},

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

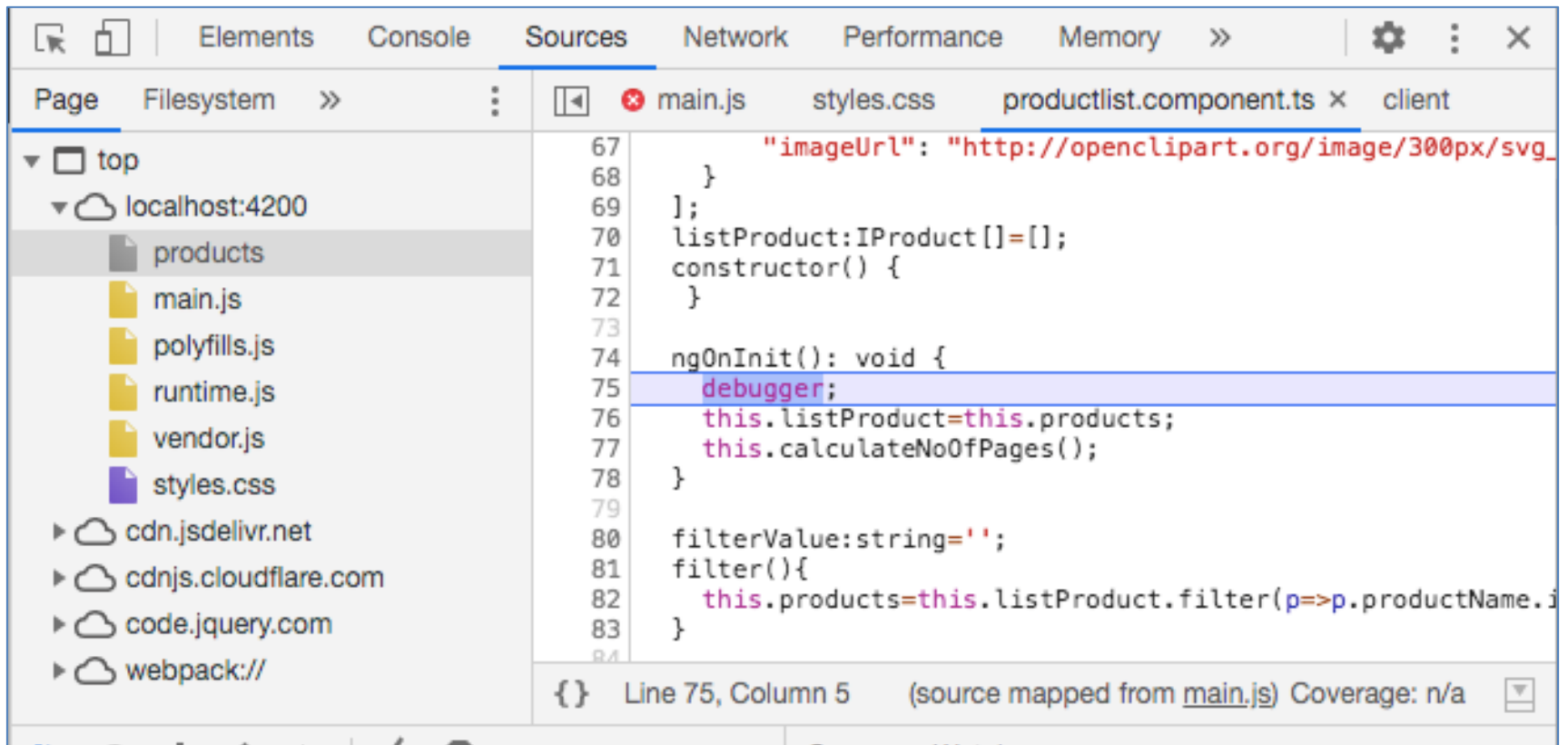
Sử dụng từ khoá debugger tạo các điểm dừng để debug

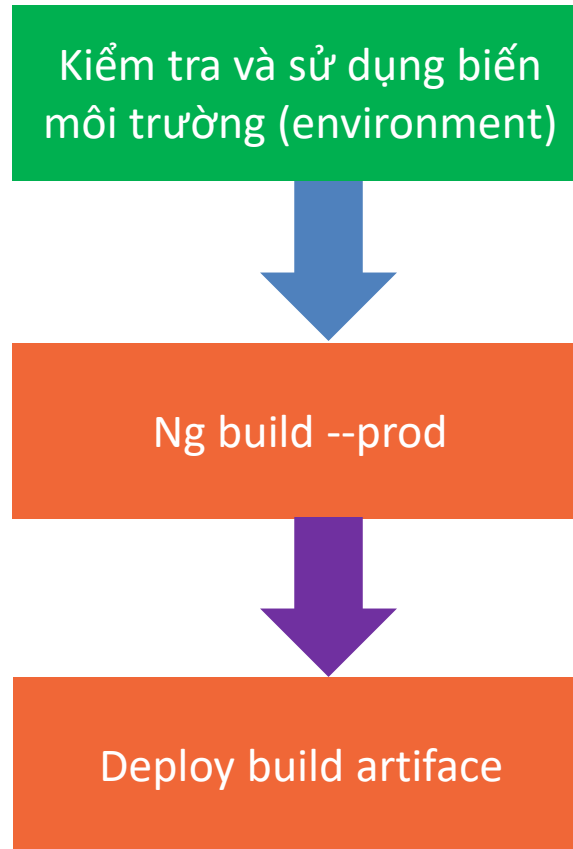
```
ngOnInit(): void {  
  debugger;  
  this.listProduct=this.products;  
  this.calculateNoOfPages();  
}
```

//phân trang

```
calculateNoOfPages()  
{  
  var noOfPages = Math.ceil(this.products.length / this.pageSize);  
  debugger;  
  this.pages = [];  
  for (let i = 1; i <= noOfPages; i++)  
  {  
    this.pages.push( { pageIndex: i });  
  }  
}
```

Khi thực thi ứng dụng ta có thể debug, F9: để chuyển đến các điểm đánh dấu debugger





Deploy ứng dụng lên Git pages

```
ng build --prod --base-href https://name.github.io/deploy-github-angular/
```



```
ng build --prod --base-href=/deploy-github-angular/
```



```
ngh --dir=dist/deploy-github-angular
```




LAZY LOADED MODULE
DEBUG
DEPLOYING

- ☑ Biết sử dụng module core và share module
- ☑ Cài đặt Feature module
- ☑ Hiểu và cài đặt Lazy Loading
- ☑ Debugging ứng dụng
- ☑ Thực hiện deploy ứng dụng



thank
you!