OPENGIS

Ionic Framework cơ bản

Đỗ Thành Long

dtlong@opengis.vn

OPENGIS

Discover the world, Learn with maps

https://opengis.vn

OPENGIS

1. Mục tiêu

Angular là nền tảng chính của Ionic, giúp tổ chức ứng dụng hiệu quả

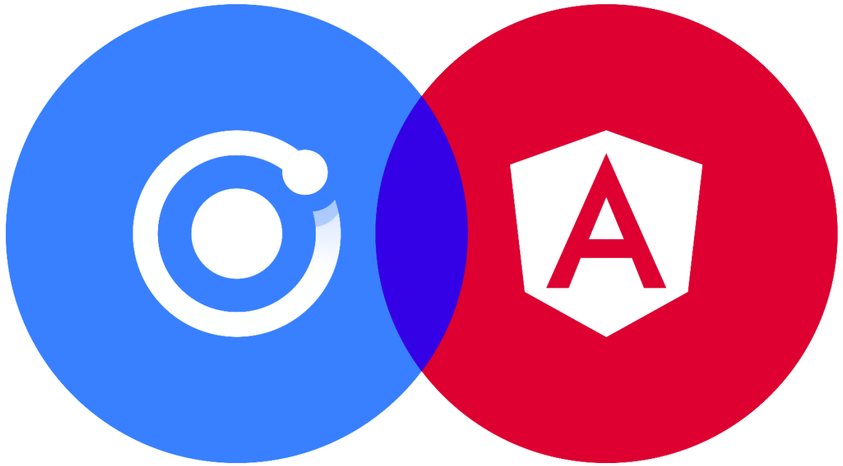
- Cấu trúc ứng dụng IONIC với NgModule
- Thành phần chính: Page, Component, Service

OPENGIS

1. Cấu trúc ứng dụng Ionic với NgModules

OPENGIS

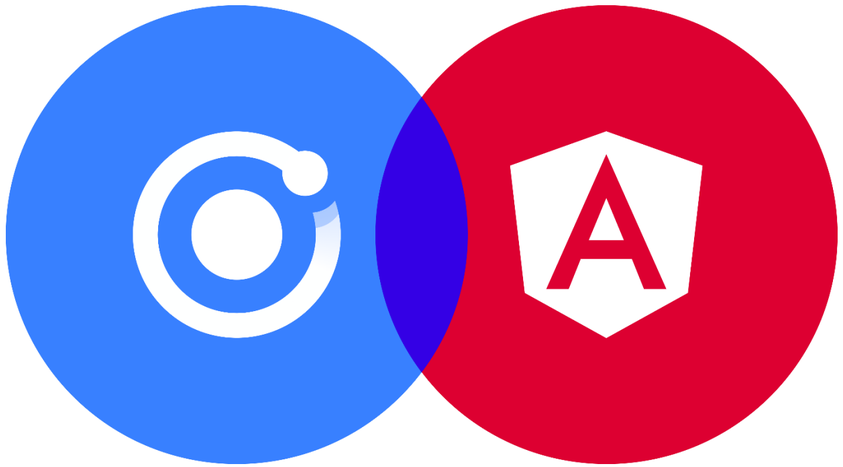
1.1. Giới thiệu về NgModules



- NgModules là một tính năng cốt lõi trong Angular, được Ionic sử dụng để tổ chức mã nguồn của ứng dụng.
- Mỗi NgModule là một khối độc lập, chứa các thành phần (components), dịch vụ (services), và các module khác cần thiết cho một phần cụ thể của ứng dụng.

OPENGIS

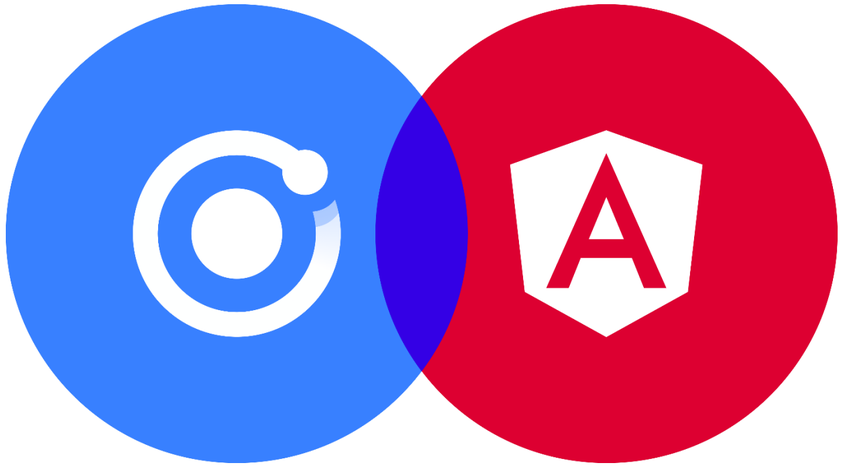
1.1. Giới thiệu về NgModules



- Giúp quản lý cấu trúc ứng dụng một cách có tổ chức, đặc biệt khi ứng dụng lớn dần.
- Hỗ trợ lazy loading để tối ưu hóa hiệu suất, chỉ tải các module khi cần thiết.

OPENGIS

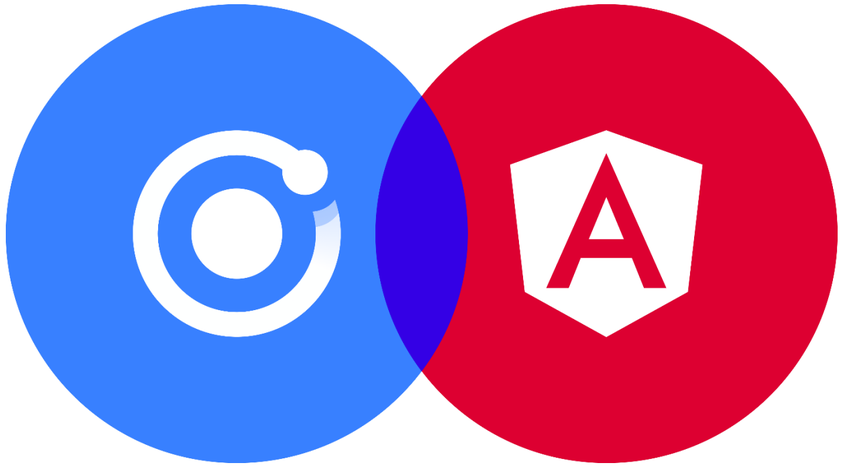
1.1. Giới thiệu về NgModules



- Trong một ứng dụng Ionic, NgModule chính (AppModule) đóng vai trò khởi động ứng dụng, trong khi các module khác (như HomeModule) quản lý các tính năng cụ thể.
- Chia nhỏ ứng dụng thành các phần độc lập, dễ bảo trì, tái sử dụng và mở rộng.

OPENGIS

1.2. Cấu trúc chi tiết của NgModule

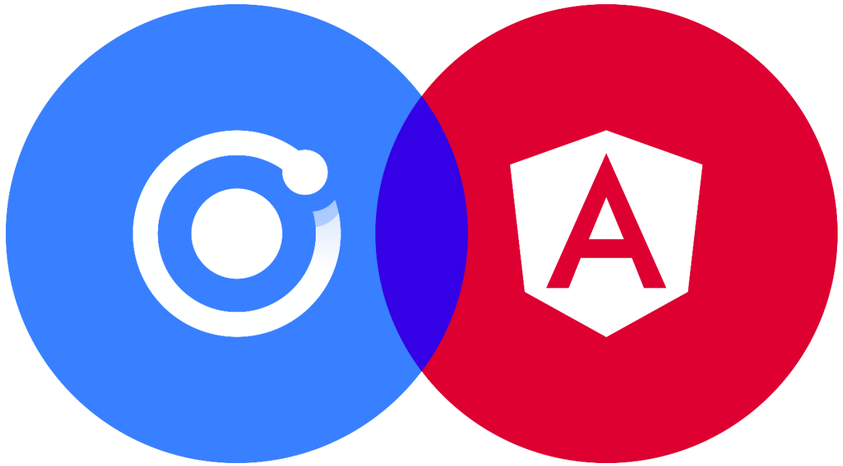


- Các thành phần chính của NgModule:
  - declarations,
  - imports,
  - providers,
  - bootstrap,
  - exports

OPENGIS

1.2. Cấu trúc chi tiết của NgModule

declarations

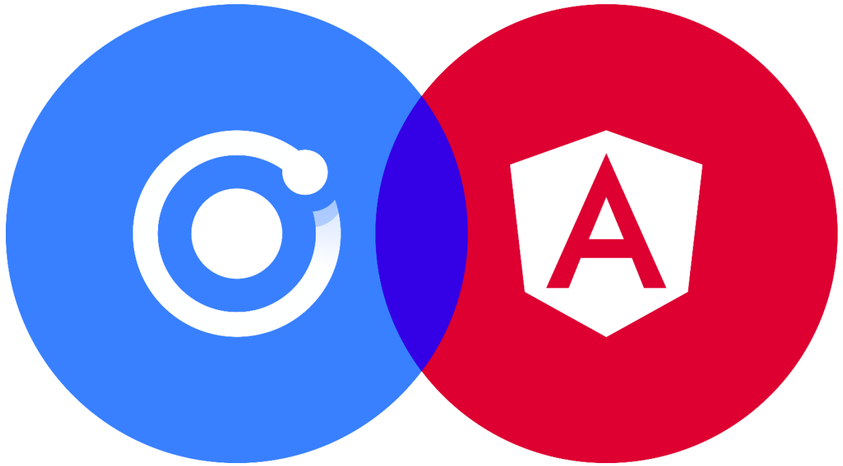


- Khai báo các Component, Directive, và Pipe thuộc module này.
- Ví dụ: Một Page như HomePage hoặc Component như ProductCardComponent phải được khai báo ở đây để sử dụng.
- Lưu ý: Mỗi thành phần chỉ được khai báo trong một NgModule duy nhất.

OPENGIS

1.2. Cấu trúc chi tiết của NgModule

imports

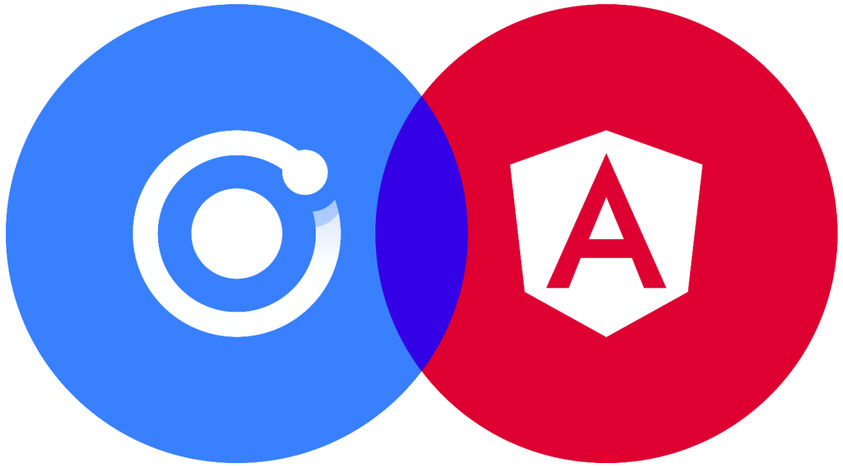


- Nhập các module khác cần thiết để module hiện tại hoạt động.
- Ví dụ: IonicModule, FormsModule, hoặc HttpClientModule.
- Trong Ionic, IonicModule.forRoot() thường được nhập vào AppModule để kích hoạt các tính năng Ionic.

OPENGIS

1.2. Cấu trúc chi tiết của NgModule

providers

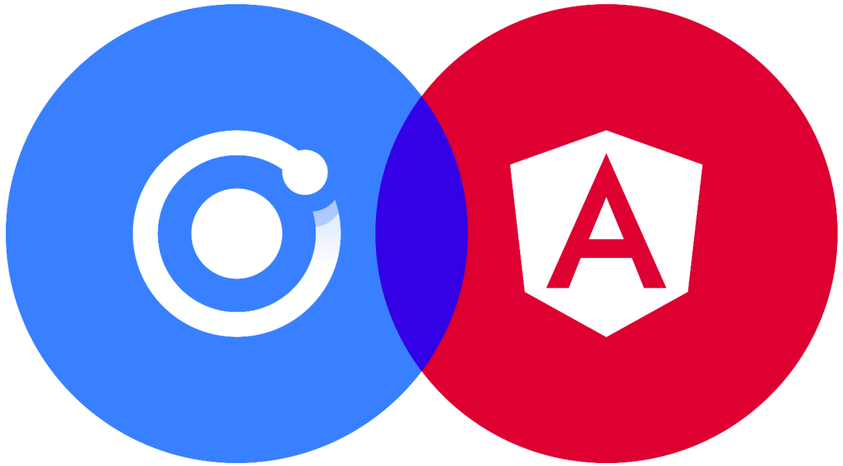


- Định nghĩa các Service được cung cấp trong module.
- Nếu khai báo ở đây, Service chỉ khả dụng trong scope của module này.
- Nếu muốn Service khả dụng toàn cục, thường khai báo trong providedIn: 'root' của Service thay vì trong providers.

OPENGIS

1.2. Cấu trúc chi tiết của NgModule

bootstrap

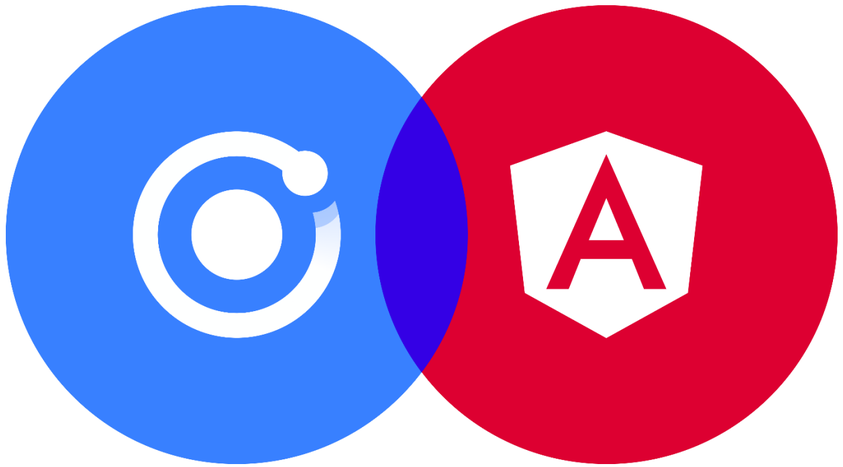


- Chỉ định component khởi động ứng dụng (chỉ dùng trong module chính - AppModule).

- Ví dụ: AppComponent là thành phần khởi động của ứng dụng Ionic.

OPENGIS

1.2. Cấu trúc chi tiết của NgModule



exports

- Chỉ định các thành phần hoặc module nào được chia sẻ ra ngoài để module khác sử dụng.

- Ví dụ: Một SharedModule có thể export một HeaderComponent để dùng ở nhiều nơi.

OPENGIS

1.2. Cấu trúc chi tiết của NgModule

- **AppModule (Module chính):**

typescript

```typescript
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { IonicModule } from '@ionic/angular';
import { AppComponent } from './app.component';
import { AppRoutingModule } from './app-routing.module';

@NgModule({
  declarations: [AppComponent],
  imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModule],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

OPENGIS

1.2. Cấu trúc chi tiết của NgModule

- **Feature Module (Module tính năng):**

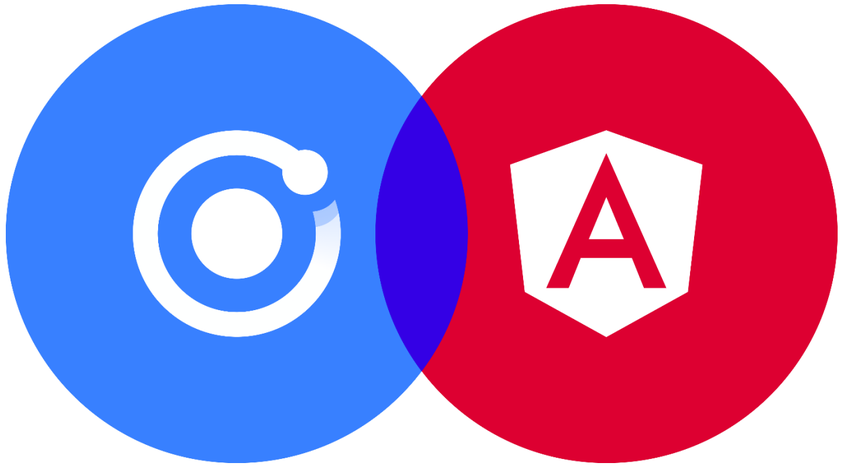
typescript

```typescript
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { IonicModule } from '@ionic/angular';
import { ProductPage } from './product.page';

@NgModule({
  declarations: [ProductPage],
  imports: [CommonModule, IonicModule]
})
export class ProductPageModule {}
```

OPENGIS

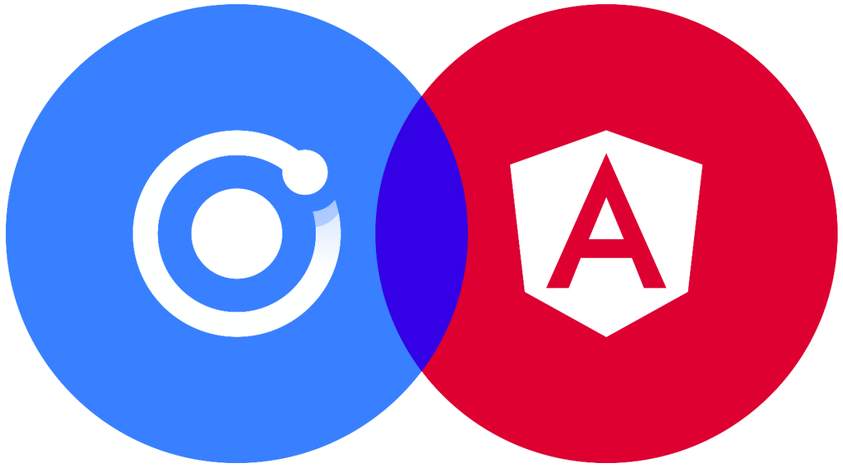
1.3. Vai trò của NgModules trong Ionic



- Ionic sử dụng NgModules để tổ chức các Page (mỗi Page thường có một module riêng).
- Ví dụ: Khi tạo một Page bằng lệnh *ionic generate page home*, Ionic tự động tạo một HomePageModule.

OPENGIS

1.3. Vai trò của NgModules trong Ionic



- NgModules cho phép Ionic áp dụng lazy loading, chỉ tải mã nguồn của Page khi người dùng truy cập vào đó.
- Điều này đặc biệt hữu ích trong các ứng dụng lớn với nhiều màn hình, giảm thời gian tải ban đầu.

OPENGIS

1.3. Vai trò của NgModules trong Ionic lazy loading

- Trong `app-routing.module.ts`:

```typescript
const routes: Routes = [
  { path: 'home', loadChildren: () => import('./home/home.module').then(m => m.HomePageModule) }
];
```

1.4. Cách tổ chức ứng dụng Ionic với NgModules

- **Cấu trúc thư mục tiêu chuẩn:**
  - `src/app/` : Thư mục gốc chứa mã nguồn ứng dụng.
    - `app.component.ts` : Component chính.
    - `app.module.ts` : NgModule chính.
    - `app-routing.module.ts` : Cấu hình điều hướng.
    - `pages/` : Thư mục chứa các Page (mỗi Page có module riêng).
    - `shared/` : Thư mục chứa các module dùng chung (Shared Modules).

https://github.com/OpenGIS-VN/Mobile-GIS-Programing-Course/tree/master/lythuyet/buoi4/test1/src/app

OPENGIS

1.4. Cách tổ chức ứng dụng Ionic với NgModules

OPENGIS

2. Thành phần chính: Page, Component, Service

OPENGIS

2. Tạo ứng dụng Ionic

Dùng lệnh CLI:

Tạo ứng dụng

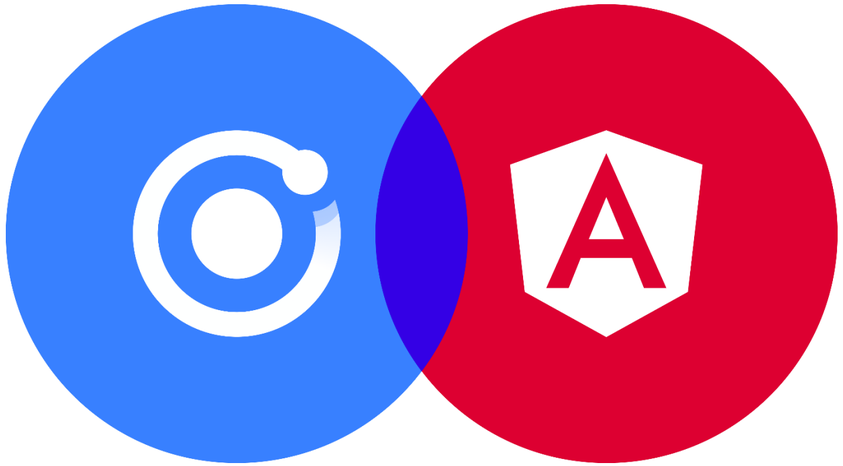
ionic start myFirstApp

 # Chạy ứng dụng trên web

ionic serve

OPENGIS

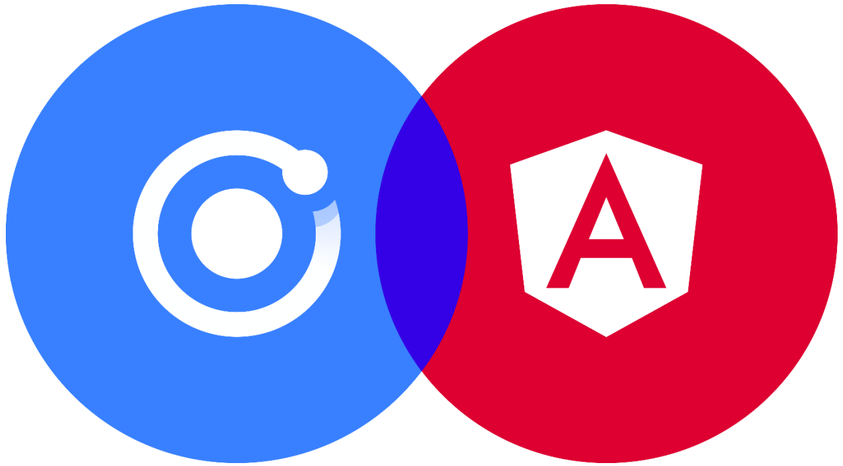
2.1. Vai trò các thành phần trong Ionic



- Trong Ionic (dựa trên Angular), ứng dụng được xây dựng từ ba thành phần chính: **Page, Component, và Service**.
-
- Mỗi thành phần có vai trò riêng, phối hợp với nhau để tạo nên một ứng dụng hoàn chỉnh.

OPENGIS

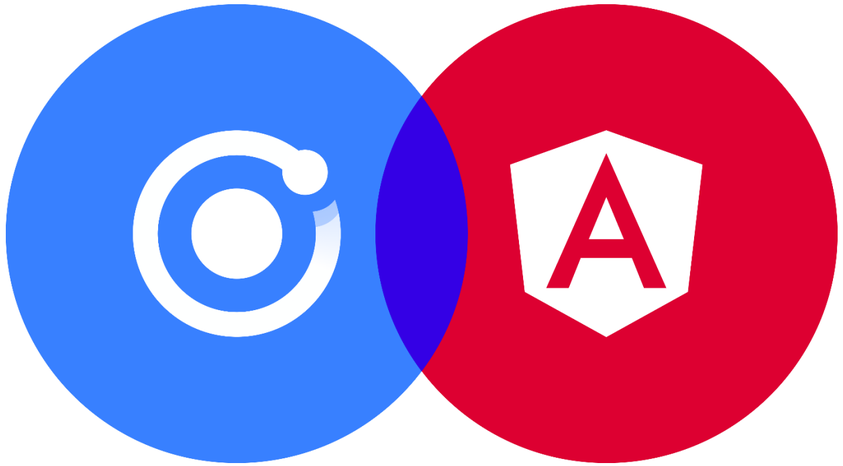
2.1. Vai trò các thành phần trong Ionic



- **Page**: Đại diện cho một màn hình hoàn chỉnh trong ứng dụng (ví dụ: trang chủ, trang chi tiết).
- **Component**: Các khối giao diện nhỏ hơn, có thể tái sử dụng trong nhiều Page (ví dụ: nút, thẻ thông tin).
- **Service**: Xử lý logic nghiệp vụ và dữ liệu, cung cấp cho Page hoặc Component (ví dụ: gọi API).

OPENGIS

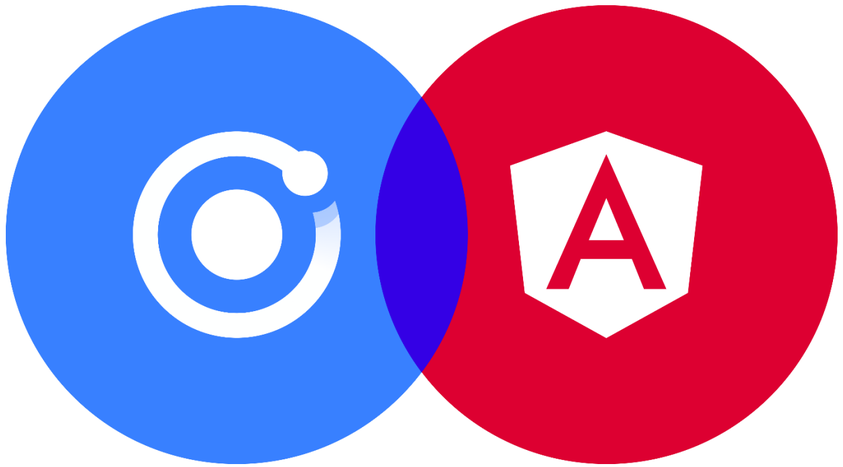
2.1. Vai trò các thành phần trong Ionic



- Một ứng dụng GIS mobile:
  - **Page**: Trang hiển thị bản đồ.
  - **Component**: Thanh điều hướng hoặc thẻ thông tin vị trí.
  - **Service**: Lấy dữ liệu tọa độ từ server.

OPENGIS

2.2. Page trong Ionic



- Page là một loại Component đặc biệt trong Ionic, đại diện cho một màn hình độc lập trong ứng dụng.
- Mỗi Page thường có:
  - Template (HTML): Giao diện của trang.
  - Class (TypeScript): Logic xử lý.
  - Styles (SCSS): Kiểu dáng riêng.

OPENGIS

2.2. Tạo Page trong Ionic

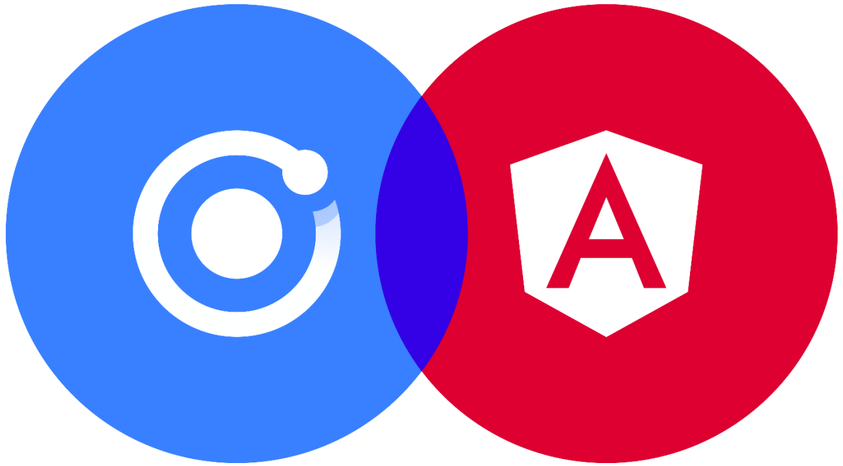
Dùng lệnh CLI:

```
ionic generate page home
```

Kết quả: Tạo thư mục `home` chứa `home.page.ts`, `home.page.html`, `home.page.scss`, và `home.module.ts`.

OPENGIS

2.2. Ionic Lifecycle Events

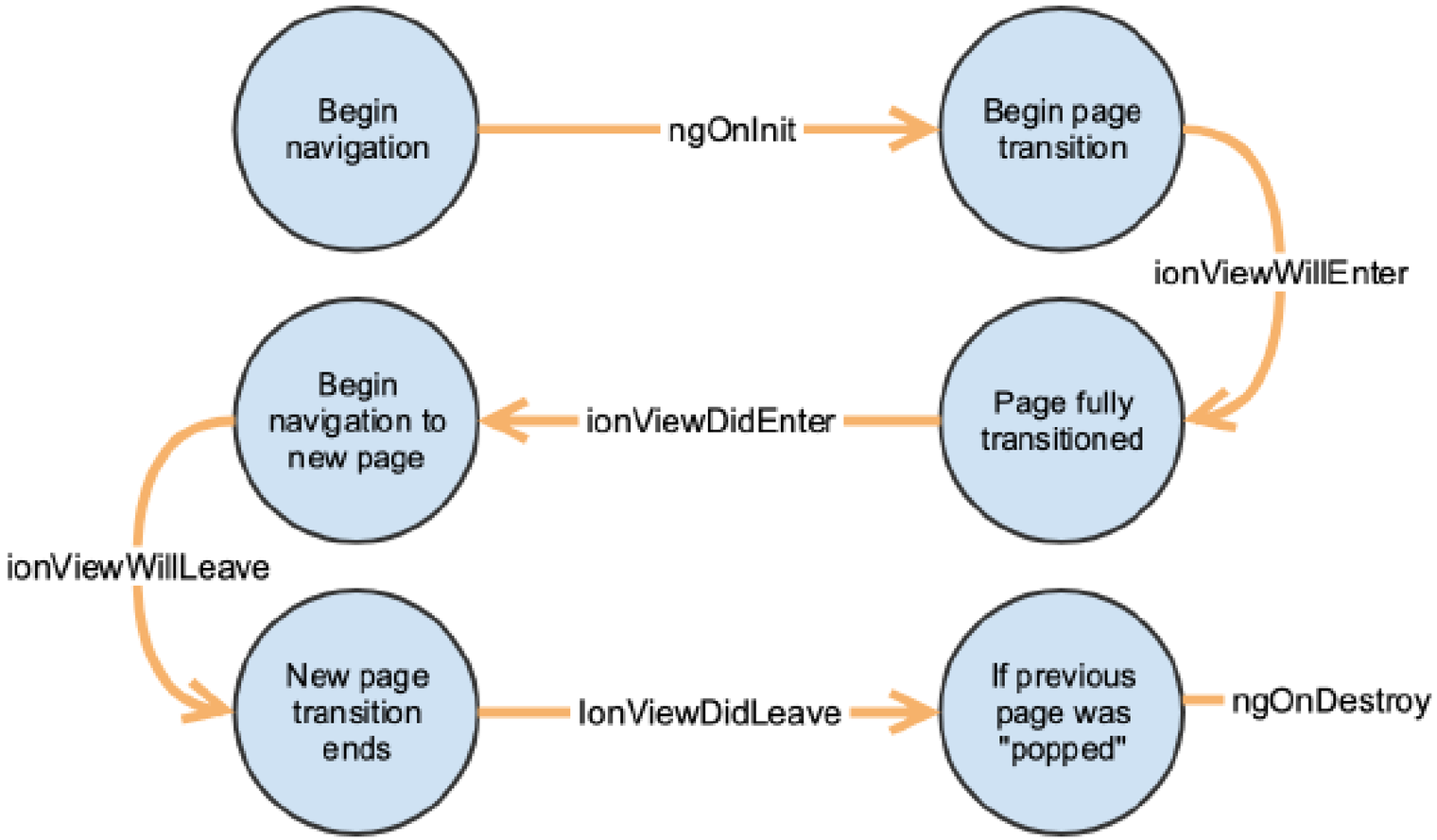


Page trong Ionic có các sự kiện vòng đời riêng:

- **ionViewWillEnter**: Chạy trước khi trang hiển thị.
- **ionViewDidEnter**: Chạy sau khi trang hiển thị hoàn toàn.
- **ionViewWillLeave**: Chạy trước khi rời trang.
- **ionViewDidLeave**: Chạy sau khi rời trang hoàn toàn

OPENGIS

2.2. Ionic Lifecycle Events



https://ionicframework.com/

OPENGIS

2.2. Ionic Lifecycle Events

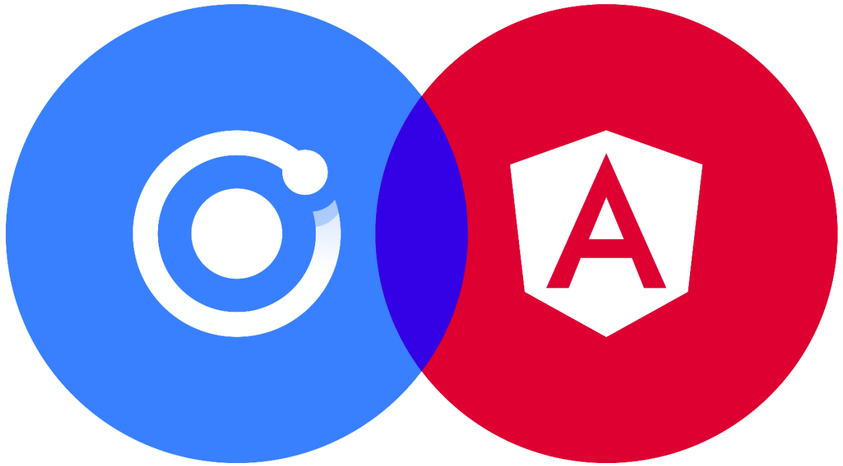
typescript

```typescript
export class HomePage {
  ionViewWillEnter() {
    console.log('Trang sắp hiển thị, tải dữ liệu...');
  }
  ionViewDidEnter() {
    console.log('Trang đã hiển thị hoàn toàn.');
  }
}
```

https://ionicframework.com/

OPENGIS

2.2. Ionic Lifecycle Events

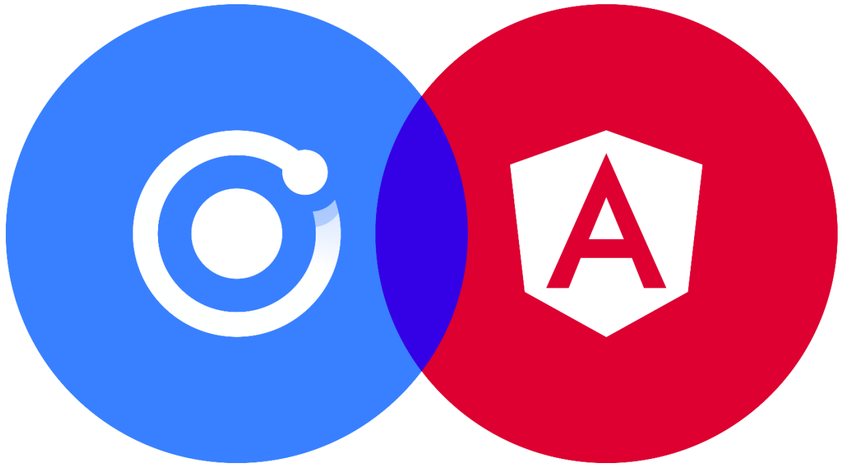


Trang MapPage:

- **ionViewDidEnter**: Khởi tạo bản đồ Leaflet.
- **ionViewWillLeave**: Lưu trạng thái bản đồ (vị trí zoom, tọa độ trung tâm).

OPENGIS

2.3. Component trong Ionic



- Component là các khối giao diện nhỏ hơn, có thể tái sử dụng trong nhiều Page hoặc Component khác.
- Mỗi Component gồm:
  - **Template (HTML)**: Giao diện hiển thị.
  - **Class (TypeScript)**: Logic điều khiển giao diện.
  - **Styles (SCSS)**: Kiểu dáng tùy chỉnh.

OPENGIS

2.2. Tạo Component trong Ionic

Dùng lệnh CLI:

```
ionic generate component product-card
```

Tạo thư mục `product-card` với các file tương ứng

OPENGIS

2.2. Tạo Component trong Ionic

- Component `ProductCardComponent`:

typescript

```typescript
import { Component, Input } from '@angular/core';

@Component({
  selector: 'app-product-card',
  template: `
    <ion-card>
      <ion-card-header>
        <ion-card-title>{{ productName }}</ion-card-title>
      </ion-card-header>
      <ion-card-content>{{ price }} VNĐ</ion-card-content>
    </ion-card>
  `,
  styles: ['ion-card { margin: 10px; }']
})
export class ProductCardComponent {
  @Input() productName: string;
  @Input() price: number;
}
```

OPENGIS

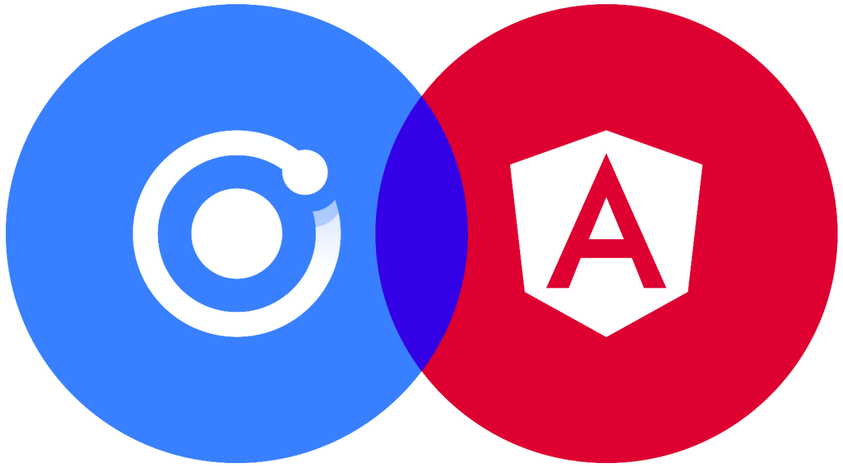
2.2. Tạo Component trong Ionic

- Sử dụng trong Page:

```html
<app-product-card [productName]="'Sách'" [price]="50000"></app-product-card>
```

OPENGIS

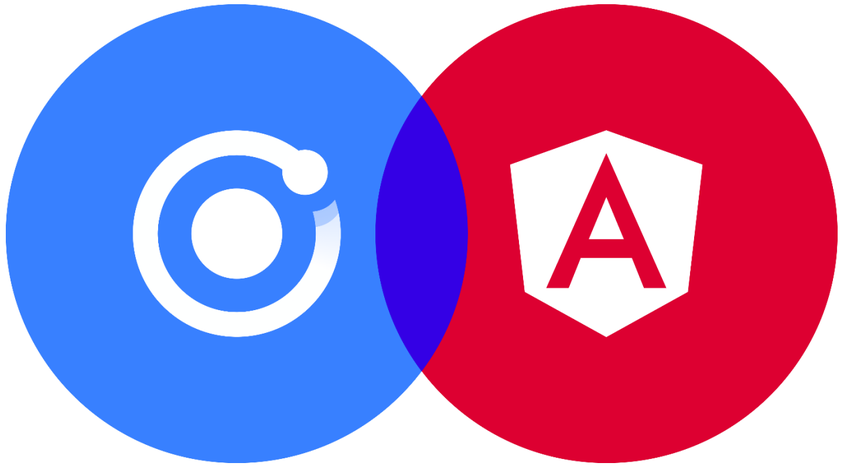
2.3. Component trong Ionic



- **Tái sử dụng Component:**
- Lợi ích: Giảm lặp mã, dễ bảo trì, đảm bảo giao diện nhất quán. Ví dụ: Một HeaderComponent có thể dùng ở mọi trang.

OPENGIS

2.4. Service trong Ionic



- Service là các lớp độc lập dùng để xử lý logic nghiệp vụ, quản lý dữ liệu hoặc tương tác với backend.
- Service được inject vào Page hoặc Component thông qua Dependency Injection (DI).

OPENGIS

2.4. Tạo Service trong Ionic

Dùng lệnh CLI:

```
ionic generate service services/product
```

Tạo file `product.service.ts` trong thư mục `services`

OPENGIS

2.4. Tạo Service trong Ionic

- Service `ProductService`:

```typescript
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Injectable({
  providedIn: 'root' // Service khả dụng toàn cục
})
export class ProductService {
  constructor(private http: HttpClient) {}

  getProducts() {
    return this.http.get('https://api.example.com/products');
  }
}
```

OPENGIS

2.4. Tạo Service trong Ionic

- Sử dụng trong Page:

typescript

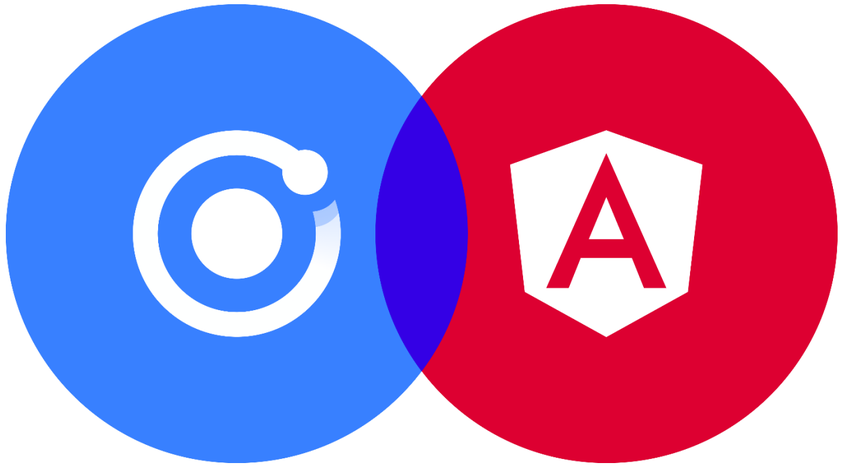
```typescript
import { Component } from '@angular/core';
import { ProductService } from '../services/product.service';

@Component({
  selector: 'app-home',
  template: '<ion-list><ion-item *ngFor="let product of products">{{ product.name }}</ion
})
export class HomePage {
  products: any[] = [];
  constructor(private productService: ProductService) {}

  ionViewDidEnter() {
    this.productService.getProducts().subscribe(data => {
      this.products = data as any[];
    });
  }
}
```

OPENGIS

2.4. Service trong Ionic

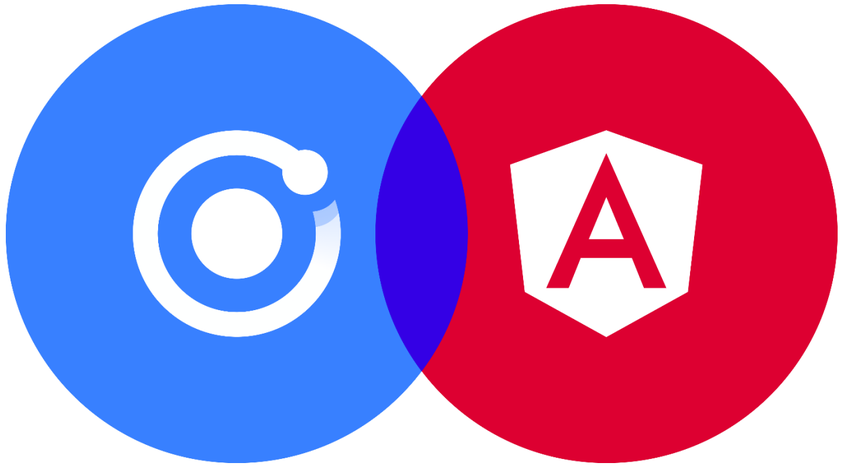


Phân loại Service:

- Root Service: Khả dụng toàn cục (dùng `providedIn: 'root'`).
- Module-level Service: Chỉ dùng trong một module cụ thể (khai báo trong `providers` của NgModule).

OPENGIS

2.4. Service trong Ionic



- `MapService`: Lấy dữ liệu bản đồ từ API (ví dụ: OpenStreetMap).
- `LocationService`: Quản lý vị trí GPS của người dùng.

OPENGIS

THANK YOU