



## Bài 7: Angular Services và Dependency Injection

### 1. Giới thiệu về Services trong Angular

Trong Angular, **Services** là các class dùng để chứa logic xử lý chung mà có thể được sử dụng lại ở nhiều nơi trong ứng dụng. Các dịch vụ này giúp chia tách logic khỏi component, giúp code dễ bảo trì hơn.

### 2. Tại sao cần dùng Services?

- **Tái sử dụng logic:** Các chức năng như gọi API, xử lý dữ liệu, xác thực người dùng nên được đặt trong Services thay vì viết lại ở nhiều component.
- **Giảm sự phụ thuộc của Component:** Giúp component chỉ tập trung vào việc hiển thị dữ liệu.
- **Dễ dàng kiểm thử (Unit Test):** Dịch vụ tách biệt giúp dễ kiểm thử hơn.

### 3. Tạo một Service trong Angular

Bạn có thể tạo một service bằng Angular CLI:

```
bash
```

```
ng generate service my-service
```

hoặc viết tắt:

```
bash
```

```
ng g s my-service
```

Lệnh này sẽ tạo ra hai file:

- `my-service.service.ts`
- `my-service.service.spec.ts` (dùng để test)

Ví dụ một service đơn giản:

typescript

```
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root' // Đăng ký service ở cấp root để có thể dùng ở mọi nơi
  trong ứng dụng
})
export class MyService {
  constructor() { }

  getMessage(): string {
    return "Hello from MyService!";
  }
}
```

## 4. Dependency Injection trong Angular

---

**Dependency Injection (DI)** là một thiết kế trong Angular giúp tự động cung cấp các service cho component mà không cần tạo thủ công.

Ví dụ: Inject một service vào component:

typescript

```
import { Component } from '@angular/core';
import { MyService } from './my-service.service';

@Component({
  selector: 'app-my-component',
  template: `<h1>{{ message }}</h1>`,
})
export class MyComponent {
  message: string = '';

  constructor(private myService: MyService) {
    this.message = this.myService.getMessage();
  }
}
```

Ở đây, **MyService** được inject vào component thông qua constructor.

## 5. Cấu hình Provider cho Services

---

Nếu không muốn dùng `providedIn: 'root'`, bạn có thể cung cấp service chỉ cho một module hoặc component bằng cách khai báo trong `providers`:

typescript

```
@NgModule({
  providers: [MyService]
})
export class AppModule { }
```

Hoặc chỉ dùng trong một component:

typescript

```
@Component({
  selector: 'app-child',
  template: `<h1>{{ message }}</h1>`,
  providers: [MyService] // Chỉ dùng trong component này
})
export class ChildComponent {
  message: string = '';

  constructor(private myService: MyService) {
    this.message = this.myService.getMessage();
  }
}
```

## 6. Kết luận

---

- **Service** giúp tách biệt logic khỏi component.
- **Dependency Injection** giúp tự động cung cấp service mà không cần khởi tạo thủ công.
- Có thể đăng ký **Service** ở cấp root, module, hoặc component.

Bạn hãy thử tạo một service riêng để thực hành nhé! Nếu có câu hỏi hoặc cần bài tập, cứ nói mình nhé. 🚀