

# GABRIELE ROMANATO

## Angular: creare e usare i servizi

Angular è un framework front-end potente e flessibile che facilita lo sviluppo di applicazioni web complesse e dinamiche. Quando si tratta di gestire dati e logica di business in un'applicazione Angular, i service giocano un ruolo fondamentale. I service sono classi singole che offrono un modo efficiente per organizzare e condividere codice, consentendo la separazione delle responsabilità e la riutilizzabilità del codice.

## Cos'è un Service in Angular?

In Angular, un service è una classe TypeScript annotata con il decoratore `@Injectable`. Questa annotazione indica a Angular che la classe può essere iniettata in altri componenti, direttive o service all'interno dell'applicazione. I service sono progettati per gestire logiche di business, condividere dati tra diversi componenti e fornire una soluzione centralizzata per l'accesso ai dati o per l'implementazione di funzionalità condivise.

## Creare un Service in Angular

Per creare un service in Angular, è possibile utilizzare il comando Angular CLI:

```
ng generate service serviceName
```

Questo comando crea automaticamente un file `service-name.service.ts` con una classe di base del service. All'interno di questa classe, è possibile definire metodi, proprietà e logiche di business necessarie.

# Dependency Injection in Angular

Per utilizzare un service in un componente o in un altro service, è necessario iniettarlo. La dependency injection in Angular è un concetto cruciale che consente di ottenere un'istanza di un service all'interno di un componente o di un altro service.

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

@Injectable({
  providedIn: 'root',
})
export class NameServiceService {
  // Proprietà o metodi del servizio
}
```

Nel caso in cui si voglia iniettare il service in un componente, è sufficiente dichiararlo nel costruttore del componente:

```
import { Component } from '@angular/core';
import { NameServiceService } from './name-service.service';

@Component({
  selector: 'app-my-component',
  template: '<p>My component</p>',
  providers: [NameServiceService],
})
export class MyComponent {
  constructor(private nameService: NameServiceService) {}
}
```

## Condividere Dati tra Componenti con i Service

Uno degli utilizzi principali dei service in Angular è la condivisione di dati tra componenti. Quando due o più componenti devono accedere o modificare gli stessi dati, un service può fungere da intermediario.

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

@Injectable({
  providedIn: 'root',
})
export class SharedDataService {
  data: string[] = [];

  addData(value: string) {
    this.data.push(value);
  }

  getData(): string[] {
    return this.data;
  }
}
```

In questo esempio, il service `SharedDataService` offre metodi per aggiungere e ottenere dati. I componenti interessati possono iniettare questo service e utilizzarlo per condividere informazioni.

## Gestire la business logic con i service

I service in Angular sono anche ideali per gestire logiche di business complesse. Questo può includere chiamate API, elaborazioni dati o qualsiasi altra operazione che coinvolga la manipolazione dei dati.

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';
import { PaginatedData } from '../dto/paginated-data.dto';
```

```
import { environment } from '../environments/environment';
import { DataModel } from '../models/data.model';

@Injectable({
  providedIn: 'root',
})
export class DataService {
  private baseUrl = environment.API_URL + '/data';

  constructor(private http: HttpClient) {}

  fetchData(page: number = 1):
  Observable<PaginatedData<DataModel>> {
    return this.http.get<PaginatedData<DataModel>>
    (`${this.baseUrl}?page=${page}`);
  }
}
```

In questo esempio, il service `DataService` utilizza Angular's `HttpClient` per effettuare una chiamata API e restituisce un'istanza di `Observable` contenente i dati ottenuti. Questo service può essere iniettato in diversi componenti che necessitano di tali dati.

## Conclusioni

I service in Angular svolgono un ruolo cruciale nella gestione dei dati e nella separazione delle responsabilità all'interno di un'applicazione. Consentono una progettazione modulare, migliorando la manutenibilità e la scalabilità del codice. Sfruttare appieno i service in Angular contribuisce a creare applicazioni robuste e flessibili, in grado di adattarsi alle esigenze complesse del mondo moderno dello sviluppo web.