Services and DI

Mohamed Mukthar Ahmed

Services & DI

Mohamed Mukthar Ahmed

What is a Service?

- Service is a broad category encompassing any value, function, or feature that an application needs.
- A **service** is typically a **class** with a narrow, well-defined purpose.
- It should do something specific and do it well.
- **Angular** distinguishes **components** from **services** to **increase modularity and reusability**.
- Ideally, a component's job is to enable only the UX (user experience).

Mohamed Mukthar Ahmed

129

Use Cases

- A component should use **services** for tasks that don't involve the **view** or application logic.
- Services are good for tasks such as
  - Fetching data from the server,
  - Validating user input, or
  - Logging directly to the console.
- By defining such processing tasks in an **injectable** service class, you make those tasks available to any component.
- In **Angular**, **dependency injection** makes those services available to components.

Mohamed Mukthar Ahmed

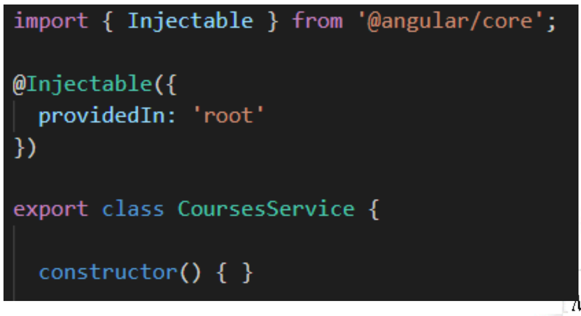
130

Create a Service

- To generate a service, use the `ng` command as follows:

```
ng  generate  service  <service_name>
```

- It will generate the service code skeleton as shown below:

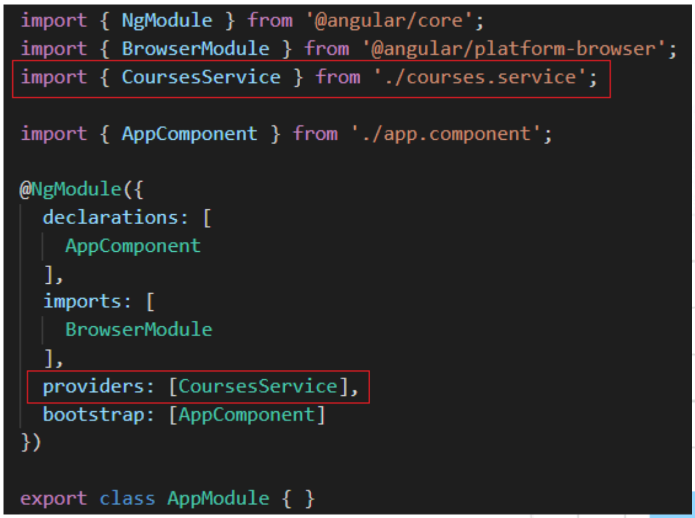


```
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})

export class CoursesService {

  constructor() { }
```

Mohamed Mukthar Ahmed

Providing a Service

- In the **app module**, include the service in the **`providers:`**



```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { CoursesService } from './courses.service';

import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule
  ],
  providers: [CoursesService],
  bootstrap: [AppComponent]
})

export class AppModule { }
```

Mohamed Mukthar Ahmed

Dependency Injection

- A **Dependency injection** (DI) is the part of the Angular framework that provides components with **access** to **services** and other resources.
- **Angular** provides the ability for you to inject a service into a component to give that component access to the service.
- In **Angular**, **dependency injection** makes those services available to components.
- Dependency injection lets you **declare the dependencies** of your TS classes without taking care of their instantiation.
- Instead, **Angular** handles the instantiation for you.

Mohamed Mukthar Ahmed

133

Dependency Injection

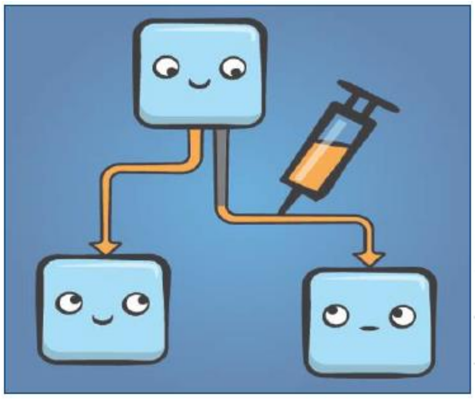
- **Dependency Injection** (DI) **design pattern** lets you write more testable and flexible code.
- ***Even though understanding DI is not critical to the usage of Angular***.
- It is strongly recommended as a best practice and many aspects of **Angular** take advantage of it to some degree.

Mohamed Mukthar Ahmed

134

How DI Works?

- The **@Injectable()** decorator defines a class as a **service** in Angular and allows Angular to inject it into a component as a **dependency**.



Mohamed Mukthar Ahmed

How DI Works?

- The **injector** is the main mechanism.
- Angular creates an application-wide injector for you during the bootstrap process
- A **provider** is an object that tells an injector how to obtain or create a dependency
- For any dependency that you need in your app, you must **register a provider with the application**
- When Angular creates a new instance of a component
  - it determines which services or other dependencies that component needs by looking at the constructor parameter.
  - **constructor(private theService: ServiceName) { }**

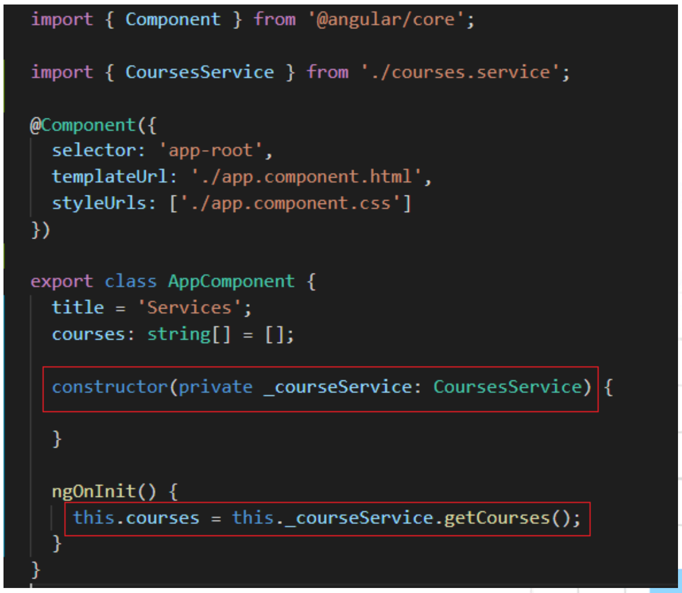
Mohamed Mukthar Ahmed

How DI Works?

- When **Angular** creates a new instance of a component
  - it determines which services or other dependencies that component needs by looking at the constructor parameter.
  - **constructor(private theService: ServiceName) { }**
- When all requested services have been resolved and returned, **Angular** can call the component's constructor with those **services** as arguments.

Mohamed Mukthar Ahmed

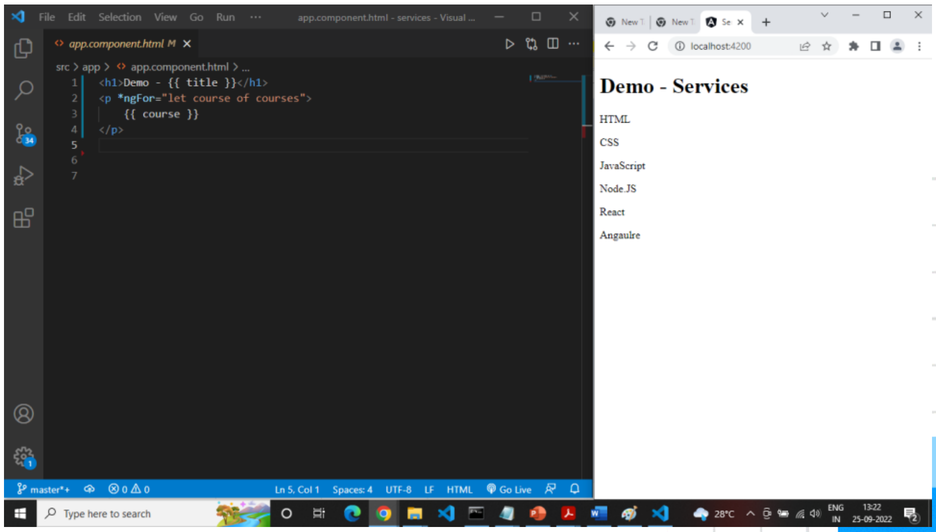
Using a Service



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

import { CoursesService } from './courses.service';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})

export class AppComponent {
  title = 'Services';
  courses: string[] = [];

  constructor(private _courseService: CoursesService) {

  }

  ngOnInit() {
    this.courses = this._courseService.getCourses();
  }
}
```

Mohamed Mukthar Ahmed

Template



```
<h1>Demo - {{ title }}</h1>
<p *ngFor="let course of courses">
    {{ course }}
</p>
```

Demo - Services

HTML

CSS

JavaScript

Node.JS

React

Angaulre

Mohamed Mukthar Ahmed

139

Mohamed Mukthar Ahmed

140