Contents

* Dependency Injection
* Shared Data via Services
* Typescript Model, Enum



**Dependency Injection**

* **What is dependency injection?**
  1. **DI**
     + Dependency Injection is often more simply referred to as DI.
     + It keeps code flexible, testable, reusable.
     + Classes can inherit external logic without knowing how to create IT.
     + Service are not alone.
     + Directives, pipes, components and so on.
     + Every schematic in angular benefits from DI in some way or another.
  2. **Injectors**
     + The **providers: []** metadata accepts services that then register with the class injector.
     + The injector aligns this type a creates an instance of that service on the class behalf.
     + By default the value is set to **root.** This translates to the root injector of the application.
     + Basically, setting the field to **root** makes the service available anywhere.
  3. **providedIn:**
     + Injectable accepts **providedIn** in argument.
     + There are primary accept ENUM in argument.
     + ‘**root**’, ‘**any’**, ‘**platform**’.
       - **root** it means there will be only one single instance of the service throughout this application independent of the modules components lazy loading eager loading anything there will be only and only one instance if the value is **root**.
       - **any** said there are two possibilities the first possibility is that we are not using lazy loading in the application it means we are loading all the components or modules eagerly means we are having eager loading if the application is having eager loading the there is no difference in between **root** and **any**. Meaning there will be only one single instance of the service in all these components across all the modules you will notice only when your application with lazy loading. If application have lazy loading modules then each lazy loading module will have a separate instance of the service.
       - **platform** When we want to shared services between two application.
* **How to work with dependency injection?**
  1. **Create Service**
     + Service code like this,

**// service**

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

**@Injectable({**

**providedIn: /\* injector goes here \*/**

**})**

**export class TemplateService {**

**constructor() { }**

**}**

* + - Module injector instance

**// module**

**import { NgModule } from '@angular/core';**

**import { CommonModule } from '@angular/common';**

**@NgModule({**

**imports: [ CommonModule ], declarations: [], providers: [ /\* services go here \*/ ]**

**})**

**export class TemplateModule { }**



* **How to work with TypeScript Model and Enum?**

**export classs Product {**

**constructor (public firstName: number){}**

**}**

**export interface ResponseData {**

**Id: number;**

**Name: string;**

**}**

**export enum MyNumber {**

**FirstNumber: 1;**

**SecondNumber:2;**

**}**