Angular 8 : Online Class

21-August-2019

By: Sahosoft Solutions

Presented by : Chandan Kumar

Angular 8 : Online Class

providedIn in Service

By: Sahosoft Solutions

Presented by : Chandan Kumar

providedIn

Services generated by the Angular CLI will automatically be created with @Injectable along with a new property called providedIn.

providedIn instructs the application where to provide the service. This allows you to only provide a service in a particular module. This also enables tree shaking so the service won't be included in the application if the module is not being used.

if you use providedIn, the injectable is registered as a provider of the Module without adding it to the providers of the module.

The service itself is a class that the CLI generated and that's decorated with @Injectable. By default, this decorator is configured with a providedIn property, which creates a provider for the service.

In this case, **providedIn: 'root'** specifies that the service should be provided in the root injector.

providedIn

There is now a new, recommended,

Way to register a provider, directly inside the @Injectable() decorator, using the new providedIn attribute.
It accepts **'root'** as a value or any module of your application. When you use 'root', your injectable will be registered as a singleton in the application, and you don't need to add it to the providers of the root module.
Similarly suppose, if you use **providedIn: UsersModule**, the injectable is registered as a provider of the **UsersModule** without adding it to the providers of the module.

src/app/user.service.ts

```
import { Injectable } from '@angular/core';
import { UserModule } from './user.module';
@Injectable({
  providedIn: UserModule,
})
export class UserService {
}
```

providedIn

The example above shows the preferred way to provide a service in a module. This method is preferred because it enables tree-shaking of the service if nothing injects it.

If it's not possible to specify in the service which module should provide it, you can also declare a provider for the service within the module:

src/app/user.module.ts

```
import { NgModule } from '@angular/core';
import { UserService } from './user.service';
@NgModule({
  providers: [UserService],
})
export class UserModule {
}
```

Angular 8 : Online Class

Tree Shakeable Providers

By: Sahosoft Solutions

Presented by : Chandan Kumar

Tree Shakeable Providers

Angular version 6 introduced a new feature, Tree Shakeable Providers.

Tree Shakeable Providers are a way to define services and other things to be used by Angular's dependency injection system in a way that can improve the performance of an Angular application.

Tree shaking

Tree shaking is a step in a build process that removes unused code from a code base. Removing unused code can be thought as "tree shaking".

By using tree shaking, we can make sure our application only includes the code that is needed for our application to run.

For example,

Suppose we have a utility library that has functions a (), b () and c ().
In our application we import and use function a () and c () but do not use b (). We would expect that the code for b () to not be bundled and deployed to our users. Tree shaking is the mechanism to remove function b () from our deployed production code that we send to our user's browsers.

Tree Shakeable Providers

Angular Tree Shaking Providers

With Tree Shaking Providers (TSP) we can use a different mechanism to register our services. Using this new TSP mechanism will provide the benefits of both tree shaking performance and dependency injection.

We have a demo application with specific code to demonstrate the different performance characteristics of how we register these services. Let's take a look at what the new TSP syntax looks like.

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

@Injectable({
  providedIn: 'root'
})
export class SharedService {
  constructor() { }
}
```