A service is used when a common functionality needs to be provided to various modules. For example, we could have a database functionality that could be reused among various modules. And hence you could create a service that could have the database functionality.

**Step 1** − Create a separate class which has the injectable decorator. The injectable decorator allows the functionality of this class to be injected and used in any Angular JS module.

@Injectable()

export class classname {

}

**Step 2** − Next in your appComponent module or the module in which you want to use the service, you need to define it as a provider in the @Component decorator.

@Component ({

providers : [classname]

})

app.service.ts

**Step 1** − Place the following code in the file created above.

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

@Injectable()

export class appService {

getApp(): string {

return "Hello world"; } }

The Injectable decorator is imported from the angular/core module. We are creating a class called appService that is decorated with the Injectable decorator.We are creating a simple function called getApp, which returns a simple string called Hello world.

**Step 3** − app.component.ts.

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

import { appService } from './app.service';

@Component ({

selector: 'demo-app',

template: '<div>{{value}}</div>',

providers: [appService] })

export class AppComponent {

value: string = "";

constructor(private xyz: appService) { }

ngOnInit(): void {

this.value = this.xyz.getApp(); }}

First, we import our appService module in the appComponent module.

Then, we register the service as a provider in this module.

In the constructor, we define a variable called \_appService of the type appService so that it can be called anywhere in the appComponent module.

As an example, in the ngOnInit lifecyclehook, we called the getApp function of the service and assign the output to the value property of the AppComponent class.