

SERVICES

Dependencies are services or objects that a class needs to perform its function. Dependency injection, or DI, is a design pattern in which a class requests dependencies from external sources rather than creating them.

Angular's DI framework provides dependencies to a class upon instantiation. Use Angular DI to increase flexibility and modularity in your applications

Creating an injectable service

➤ `ng g s shared/common`

This command will create a common service in shared folder

Example:

Step 1: `ng g m mod/module1`

This will create one module and need to register this module in parent app module.

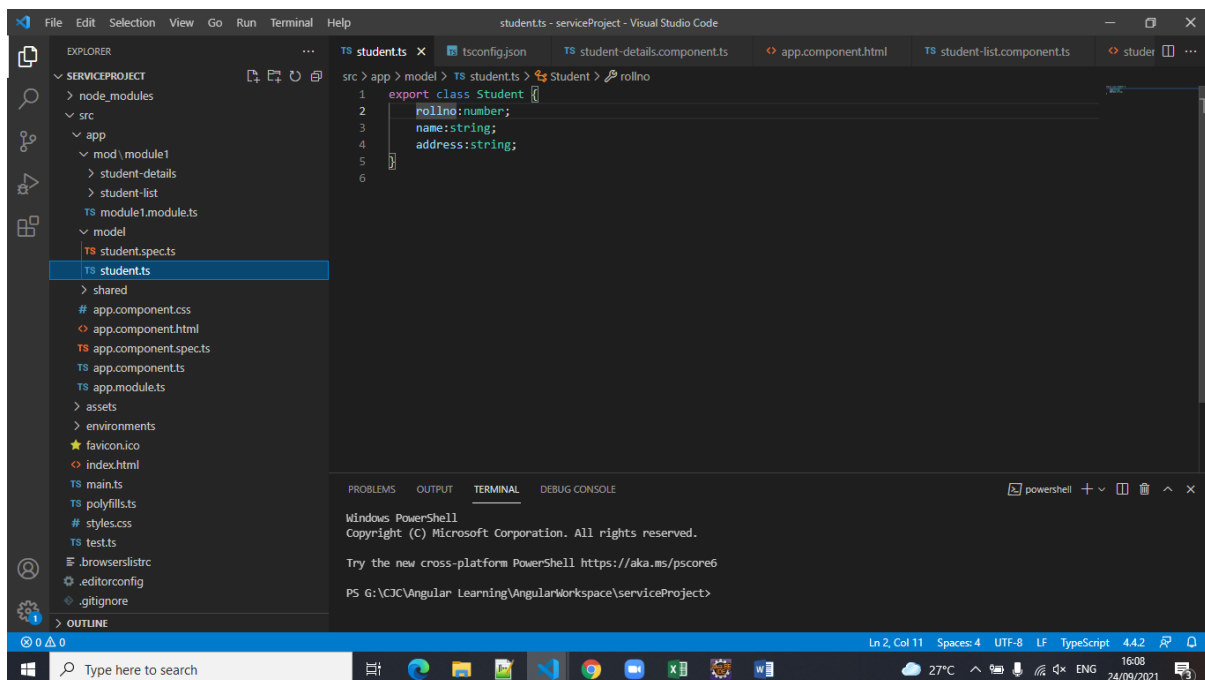
Step 2: `ng g c mod/module1/student-list`

Register this component in declarations and export section of parent module.

Step 3: `ng g c mod/module1/student-details`

(Same as above)

Step 4: `ng g class Student`



Step 5 : ng g s shared/common

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The file explorer shows a project structure with a 'shared' folder under 'src/app'. The 'common.service.ts' file is selected and its content is displayed in the editor. The code defines an injectable service with a constructor and a static array of student data.

```
src > app > shared > TS common.service.ts > CommonsService > stuData
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class CommonsService {
8
9   constructor() { }
10
11   stuData:Student[]=[
12     {
13       rollno:1,
14       name:'abc',
15       address: 'pune'
16     },
17     {
18       rollno:2,
19       name:'pqr',
20       address: 'pune'
21     },
22     {
23       rollno:3,
24       name:'xyz',
25       address: 'pune'
26     }
27   ]
28 }
29
```

The terminal at the bottom shows the command prompt and the status bar indicates the file is at line 17, column 6.

Step 6:

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The file explorer shows a project structure with a 'student-list' folder under 'src/app/mod'. The 'student-list.component.ts' file is selected and its content is displayed in the editor. The code defines a component that imports the 'CommonService' and uses it to initialize the 'stuData' property.

```
src > app > mod > student-list > TS student-list.component.ts > StudentListComponent > stuData
1 import { Component, OnInit } from '@angular/core';
2 import { Student } from 'src/app/mod/student';
3 import { CommonsService } from 'src/app/shared/common.service';
4
5 @Component({
6   selector: 'app-student-list',
7   templateUrl: './student-list.component.html',
8   styleUrls: ['./student-list.component.css']
9 })
10 export class StudentListComponent implements OnInit {
11
12   constructor(private commonsService:CommonService) { }
13
14   ngOnInit(): void {
15   }
16   stuData1:Student[]=this.commonsService.stuData;
17 }
18
```

The terminal at the bottom shows the command prompt and the status bar indicates the file is at line 16, column 11.

Step 7:

