

S

Sandip is presenting

```
component.html > app.component.html TS add-plan.component.ts add-plan.component.html x TS plan-details.component.html ...
onent.html > div.col-md-6.col-md-offset-3 > div.panel.panel-primary > div.panel-body > form > div > span#successMessage.text.text-success
32   fromControlName="planValue" id="planValue">
33   ie").valid && addPlanForm.get('planValue').touched" id="planValueError">Please Enter Plan Value
34
35
36
37   it' fromControlName="data" id="data">
38   .valid && addPlanForm.get('data').touched" id="dataError">Please Enter GB/day</span>
39
40
41   all</label>
42   * fromControlName="unlimitedCalls" name="unlimitedCalls" id="true" value="true">True
43   * fromControlName="unlimitedCalls" name="unlimitedCalls" id="false" value="false">False
44
45
46
47   ol' fromControlName="addOns" id="addOns" cols="30" rows="10"></textarea>
48   .valid && addPlanForm.get('addOns').touched" id="addOnsError">Please Enter Add Ons</span>
49
50
51
52   * fromControlName="planType" name="planType" id="post" value="postPaid">Postpaid
53   * fromControlName="planType" name="planType" id="pre" value="prePaid">Prepaid
54
55
56   ury" [disabled]="!addPlanForm.valid">Add Plan</button>
57
58
59   t text-success" id="successMessage"{{(successMessage)}}</span>
60   text-danger" id="errorMessage"{{(errorMessage)}}</span>
61
62
63
```

S

Sandip

You

master 9.2.0

Ln 59, Col 94 Spaces: 4 UTF-8 LF HTML

Mozilla Firefox

add-plan.comp

10:3

Execute

View Result

et.google.com/mrs-ktms-vph

enting

Result

Last modified on 21/8/2022 at 16:7:12

Module	Success	Failed	Total
Testing PlanService	8	0	8
Testing Routing	3	0	3
Testing AppComponent	3	0	3
Testing UnitPipe	1	1	2
Testing ViewPlanComponent	1	1	2
Testing AddPlanComponent	30	7	37
Total	46	9	55

Close

ktms-vph

to search



S Sandip is presenting

```
component.ts  app.component.html  add-plan.component.ts  add-plan.component.html  plan-details.component
Frontend > src > app > add-plan > add-plan.component.ts > AddPlanComponent > addPlan
20  ngOnInit() {
21    //Add specified validators
22    this.addPlanForm=this.fb.group({
23      planValue:['',Validators.required],
24      data:['',[Validators.required,Validators.max(200)]],
25      unlimitedCalls:['',Validators.required],
26      addOns:['',[Validators.required,Validators.pattern(/^[a-zA-Z!@#$%^&*$/)]],
27      planType:['',Validators.required]
28    })
29  }
30
31  /*
32  It should invoke addPlan() method of PlanService by passing addPlanForm object as a parameter.
33  The success callback should populate the successMessage with the message in response
34  The error callback should populate the errorMessage with the message in response
35  */
36  addPlan() {
37    // Code here
38    this.errorMessage=null;
39    this.successMessage=null;
40    this.planService.addPlan(this.addPlanForm.value).subscribe(
41      res=>{
42        this.successMessage=res.message
43      },
44      err=>{
45        this.errorMessage=err.error.message
46      }
47    )
48  }
49
50 }
```

Sandip

You

S

Sandip is presenting

```
ts plan-details.component.html add-plan.component.html plan-details.component.ts lib.es5.d.ts ts lib ts ...
Frontend > src > app > plan-details > plan-details.component.ts > PlanDetailsComponent > deletePlan
3. The success callback should populate the planDetails with response
22
23
24
25
26 set planId(pId: number) {
27     // Code here
28     this.planService.getPlanDetails(this.pId).subscribe(
29         res=>{
30             this.planDetails=res;
31         }
32     )
33 }
34
35
36
37
38 /*
39 1. It should invoke deletePlan() of PlanService
40 by passing the value of pId which in turn returns an observable
41 2. The success callback should populate the displayMessage with message in response
42 */
43
44 deletePlan() {
45     this.planService.deletePlan(this.pId).subscribe(
46         res=>{
47             this.displayMessage=res.message
48         }
49     )
50     // Code here
51 }
52
```

Ln 47, Col 10 Spaces: 4 UTF-8 LF TypeScript 3.8.3

plan-details.co... 10:35

Execute View Result

S

Sandip

You


```
ts plan-details.component.html add-plan.component.html plan-details.component.ts lib.es5.d.ts
Frontend > src > app > plan-details > TS plan-details.component.ts > PlanDetailsComponent > deletePlan
0 templateUrl: './plan-details.component.html'
1 styleUrls: ['./plan-details.component.css']
2
3 export class PlanDetailsComponent {
4
5     constructor(private planService: PlanService) { }
6
7     planDetails: any;
8     displayMessage: string;
9     pid: number;
10    bool: any;
11
12    /*
13     1. Add the appropriate decorator to below setter method so that it can receive
14     the data sent by ViewPlan Component
15     2. It should invoke getPlanDetails() of PlanService, which in turn returns an observal
16     3. The success callback should populate the planDetails with response
17    */
18
19    set planId(pid: number) {
20        // Code here
21        this.planService.getPlanDetails(this.pid).subscribe(
22            res=>{
23                this.planDetails=res;
24            }
25        )
26    }
27
28    /*
29     1. It should invoke deletePlan() of PlanService
30    */
31}
```

Ln 47, Col 16 Spaces: 4 UTF-8 LF TypeScript 3.0

plan-details.co

Executing ... View Result



Sandip



You

EXPLORER

src > app > app.component.html > nav.navbar.navbar-inverse > div.container-fluid > ul.nav.navbar-nav

1. Add the code to link to the appropriate routes
2. Add the tag required to load the view of routed componen

```
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">Pictor Telecom</a>
    </div>
    <ul class="nav navbar-nav">
      <li>
        <a [routerLink]="['/addPlan']">Add Plan</a>
      </li>
      <li>
        <a [routerLink]="['/viewPlan']">View Plan</a>
      </li>
    </ul>
  </div>
</nav>
<div>
  <router-outlet></router-outlet>
</div>
```

app.component.html

app.component.ts

app.module.ts

assets

environments

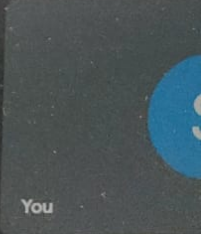
browserslist

OUTLINE

TIMELINE

Ln 16, Col 33 Spaces 2 UTF-8 LF HTML

Execute View Result



S

Sandip is presenting

```
component.html > app.component.html > add-plan.component.ts > add-plan.component.html > plan-details.component.html > ...
component.html > div.col-md-6.col-md-offset-3 > div.panel.panel-primary > div.panel-body > form > div > span.Rs.successMessage.test.success
23
24
25
26
27
28
29
30
31
32 <input type="text" name="planValue" id="planValue">
33 & addPlanForm.get('planValue').touched" id="planValueError">Please Enter Plan Value</span>
34
35
36
37 <input type="text" name="data" id="data">
38 PlanForm.get('data').touched" id="dataError">Please Enter GB/day</span>
39
40
41 >
42 <input type="checkbox" name="unlimitedCalls" id="unlimitedCalls" value="true">True
43 <input type="checkbox" name="unlimitedCalls" id="unlimitedCalls" value="false">False
44
45
46
47 <div id="addOns" style="border: 1px solid #ccc; padding: 5px; min-height: 40px;>
48 addPlanForm.get('addOns').touched" id="addOnsError">Please Enter Add Ons</div>
49
50
51
52 <input type="radio" name="planType" id="post" value="postPaid">Postpaid
53 <input type="radio" name="planType" id="pre" value="prePaid">Prepaid
54
```



master



Mozilla Firefox

add-plan.comp...

Ln 55, Col 94 Spaces: 4 UTF-8 LF HTML

Execute

View Result

10:31

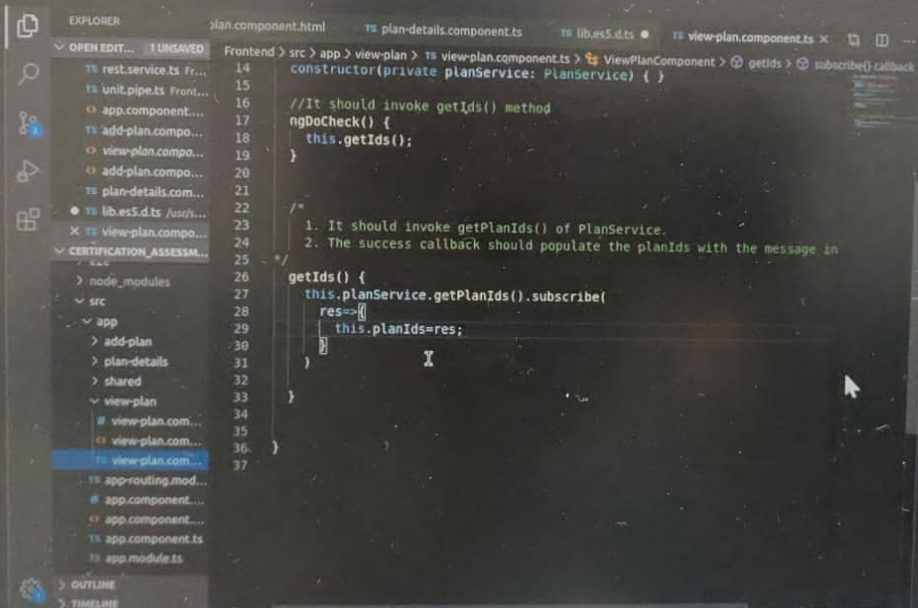
S

Sandip

You

S

Sandip is presenting



The screenshot shows a VS Code editor with the following components:

- EXPLORER:** A file tree on the left showing the project structure. The 'view-plan' directory is expanded, showing files like 'view-plan.component.ts' and 'view-plan.component.html'. The 'view-plan.component.ts' file is selected and highlighted in blue.
- EDITOR:** The main workspace showing the code for 'view-plan.component.ts'. The code is as follows:

```
14 constructor(private planService: PlanService) { }
15
16 //It should invoke getIds() method
17 ngDoCheck() {
18   this.getIds();
19 }
20
21 /*
22 1. It should invoke getPlanIds() of PlanService.
23 2. The success callback should populate the planIds with the message in
24 */
25
26 getId() {
27   this.planService.getPlanIds().subscribe(
28     res=>{
29       this.planIds=res;
30     }
31   )
32 }
33
34
35
36
37
```
- STATUS BAR:** At the bottom, it shows 'Ln 29, Col 26', 'Spaces: 2', 'UTF-8', 'LF', 'TypeScript', and '3.8.3'.

S

Sandip

You


```
component.ts  app.component.html  add-plan.component.ts  add-plan.component.html  plan-details.component

```

S

Sandip

You

Ln 59, Col 94 Spaces: 4 UTF-8 LF HTML

Execute

View Result

10:3

Sandip is presenting

14

[View Result](#)

S

Sandip

S

You

```
ts plan-details.component.html add-plan.component.html plan-details.component.ts lib.es5.d.ts
Frontend > src > app > plan-details > ts plan-details.component.ts > PlanDetailsComponent > deletePlan
22 3. The success callback should populate the planDetails with response
23
24 */
25
26 set planId(pId: number) {
27 // Code here
28 this.planService.getPlanDetails(this.pId).subscribe(
29 res=>{
30 this.planDetails=res;
31 }
32 )
33 }
34
35
36
37 /*
38 1. It should invoke deletePlan() of PlanService
39 by passing the value of pId which in turn returns an observable
40 2. The success callback should populate the displayMessage with message in response
41 */
42
43 deletePlan() {
44 this.planService.deletePlan(this.pId).subscribe()
45 res=>{
46 this.displayMessage=res.message
47 }
48 // Code here
49 }
50
51
52
```

Ln 47, Col 10 Spaces: 4 UTF-8 LF TypeScript 3.8.3

Mozilla Firefox plan-details.co... 10:30 Execute View Result

S

Sandip

You

S

Sandip is presenting

```
component.ts  app.component.html  TS add-plan.component.ts  add-plan.component.html x  plan-details.component.html  ...
div.col-md-6.col-md-offset-3 > div.panel.panel-primary > div.panel-body > form > div > span#successMessage.text.text-success
45 <div class="from-group">
46   <label for="addOns">Add Ons</label>
47   <textarea type="text" class="from-control" fromControlName="addOns" id="addOns">
48   <span *ngIf="!addPlanForm.get('addOns').valid && addPlanForm.get('addOns')
49   </div>
50   <div class="from-group">
51   <label for="planType">Type</label>
52   <input type="radio" class="from-control" fromControlName="planType" name="
53   <input type="radio" class="from-control" fromControlName="planType" name="
54   </div>
55   <div>
56   <button type="submit" class="btn btn-primary" [disabled]="!addPlanForm.va
57   </div>
58   <div>
59   <span *ngIf="successMessage" class="text text-success" id="successMessage"
60   <span *ngIf="errorMessage" class="text text-danger" id="errorMessage">{{
61   </div>
62   </form>
63
64
65
66   </div>
67 </div>
68 </div>
```

S

Sandip

You

master

Mozilla Firefox

add-plan.comp

Ln 59, Col 94, Spaces: 4, UTF-8, LF, HTML

Execute

View Result

10:30


```
PlanService > deletePlan
```

```

23 // Code here
24 return this.httpClient.post(this.url,data);
25 }
26
27 /*
28 Consumes the web service exposed at the URL -> http://localhost:3020/planDetails
29 After sending the request, the response must be an observable of array*
30 Return the response back to the ViewPlanComponent
31
32 */
33 getPlanIds() :Observable<any[]>{
34 // Code here
35 return this.httpClient.get<any[]>(this.url);
36 }
37
38 /*
39 Consumes the web service exposed at the URL -> http://localhost:3020/planDetails/:planId
40 After sending the request, the response must be an Observable of type 'Plan'
41 Return the response back to the PlanDetailsComponent
42
43 */
44 getPlanDetails(planId) :Observable<Plan>{
45 // Code here
46 return this.httpClient.get<Plan>(`${this.url}/${planId}`);
47 }
48
49
50
51 /*
52 Consumes the web service exposed at the URL -> http://localhost:3020/planDetails/:planId
53 After sending the request, the response must be an Observable
54 Return the response back to the PlanDetailsComponent

```

[View Result](#)

S

Sandip is presenting

```
component.html > app.component.html TS add-plan.component.ts add-plan.component.html x plan-details.component.html  
onent.html > div.col-md-6.col-md-offset-3 > div.panel.panel-primary > div.panel-body > form > div > span#successMessage.text.text-success  
32 fromControlName="planValue" id="planValue">  
33 ).valid && addPlanForm.get('planValue').touched" id="planValueError">Please Enter Plan Value  
34  
35  
36  
37 fromControlName="data" id="data">  
38 id && addPlanForm.get('data').touched" id="dataError">Please Enter GB/day</span>  
39  
40  
41 </label>  
42 fromControlName="unLimitedCalls" name="unLimitedCalls" id="true" value="true">True  
43 fromControlName="unLimitedCalls" name="unLimitedCalls" id="false" value="false">False  
44  
45  
46  
47 <input fromControlName="addOns" id="addOns" cols="30" rows="10"></Textarea>  
48 valid && addPlanForm.get('addOns').touched" id="addOnsError">Please Enter Add Ons</span>  
49  
50  
51  
52 fromControlName="planType" name="planType" id="post" value="postPaid">Postpaid  
53 fromControlName="planType" name="planType" id="pre" value="prePaid">Prepaid  
54  
55  
56 <button [disabled]="!addPlanForm.valid">Add Plan</button>  
57  
58  
59 <span text="success" id="successMessage">{{(successMessage)}}</span>  
60 <span ext="danger" id="errorMessage">{{(errorMessage)}}</span>  
61  
62  
63
```

S

Sandip

You

Ln 33, Col 84 Spaces 4 UTF-8 LF HTML

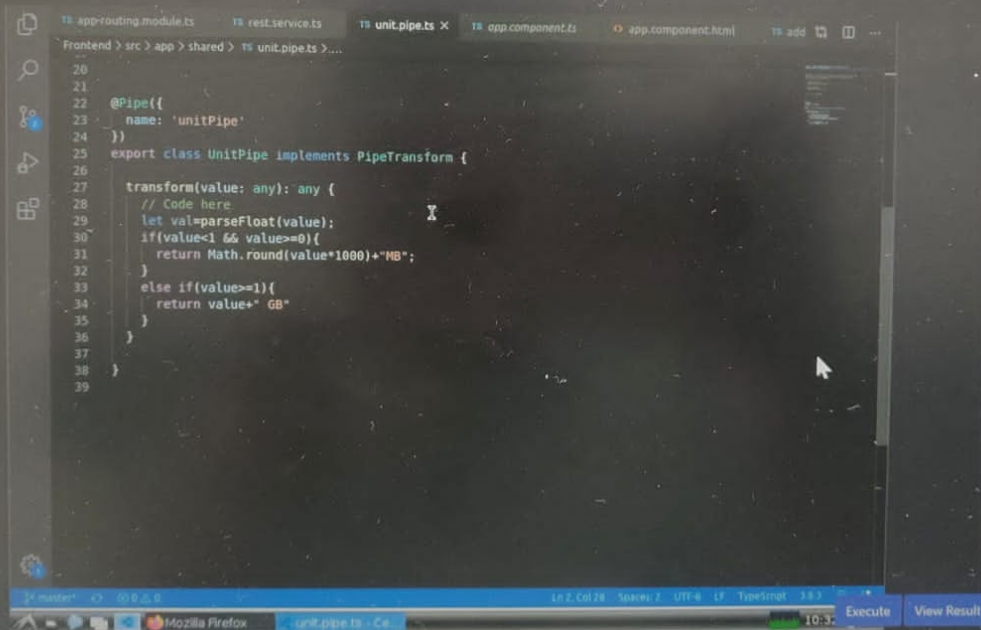
Execute

View Result

10:3

S

Sandip is presenting



The screenshot shows a code editor with several tabs: `app-routing.module.ts`, `rest.service.ts`, `unit.pipe.ts` (active), `app.component.ts`, `app.component.html`, and `add`. The active file `unit.pipe.ts` contains the following TypeScript code:

```
Frontend > src > app > shared > TS unit.pipets >...  
20  
21  
22 @Pipe({  
23   name: 'unitPipe'  
24 })  
25 export class UnitPipe implements PipeTransform {  
26  
27   transform(value: any): any {  
28     // Code here  
29     let val=parseFloat(value);  
30     if(value<1 && value>=0){  
31       return Math.round(value*1000)+"MB";  
32     }  
33     else if(value>=1){  
34       return value+" GB"  
35     }  
36   }  
37 }  
38  
39
```

The status bar at the bottom indicates the file is `unit.pipe.ts` in the `src` directory, using `UTF-8` encoding and `LF` line endings. The editor is running on `VS Code` version `1.53.2`. The bottom right corner shows a green `Execute` button and a `View Result` button.



Sandip



You

S

Sandip is presenting

Result

Last modified on 21/8/2022 at 16:4:50

Module	Success	Failed	Total
Testing PlanService	8	0	8
Testing Routing	3	0	3
Testing AppComponent	3	0	3
Testing UnitPipe	1	1	2
Testing ViewPlanComponent	1	1	2
Testing AddPlanComponent	30	7	37
Total	46	9	55

Close

S

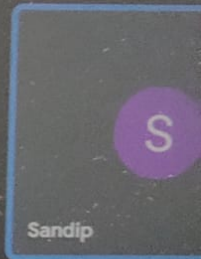
Sandip

You


```
TS app-routing.module.ts TS rest.service.ts X TS unit.pipe.ts TS app.component.ts app.component.html TS add TS ...
Frontend > src > app > shared > TS rest.service.ts > PlanService > deletePlan
41  Return the response back to the PlanDetailsComponent
42  */
43
44  getPlanDetails(planId) :Observable<Plan>{
45      // Code here
46      return this.httpClient.get<Plan>(`${this.url}/${planId}`);
47  }
48
49
50
51  /*
52  Consumes the web service exposed at the URL -> http://localhost:3020/planDetails/:planId
53  After sending the request, the response must be an Observable
54  Return the response back to the PlanDetailsComponent
55  */
56
57  deletePlan(planId) {
58      // Code here
59      return this.httpClient.delete(`${this.url}/${planId}`);
60  }
61
62  }
63
```

Ln 59, Col 60 Spaces: 2 UTF-8 LF TypeScript 3.8.3

Execute View Result



Sandip



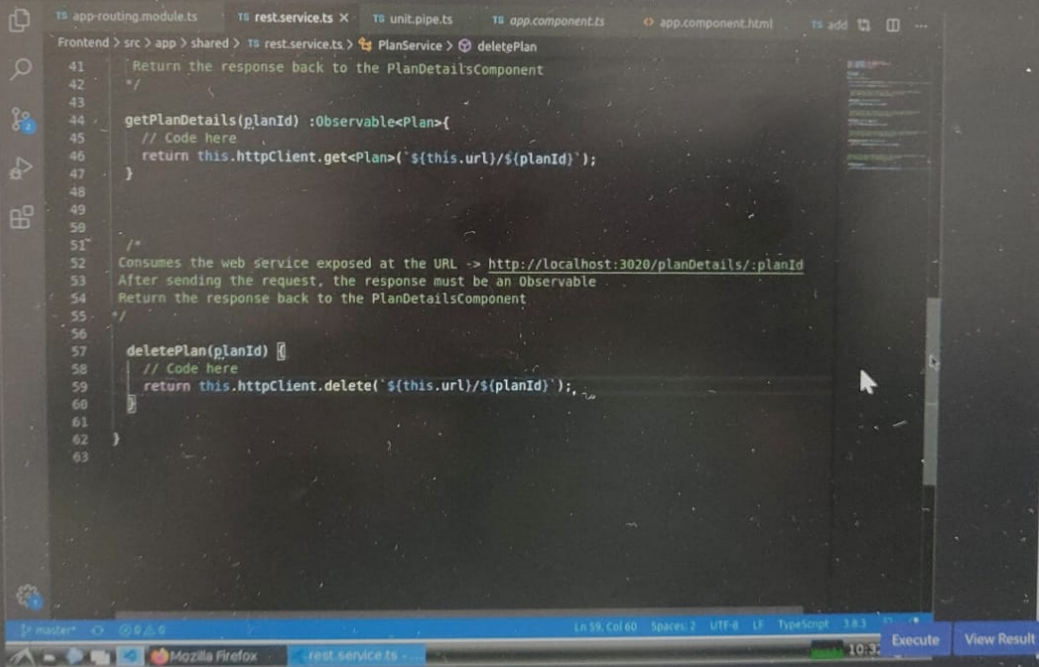
You

S

Sandip is presenting

```
TS app-routing.module.ts TS rest.service.ts X TS unit.pipe.ts TS app.component.ts app.component.html TS add TS ...
Frontend > src > app > shared > TS rest.service.ts > PlanService > deletePlan
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { Observable } from 'rxjs';
4 import { Plan } from './plan';
5
6
7 @Injectable({
8   providedIn: 'root'
9 })
10 export class PlanService {
11
12   url = "http://localhost:3020/planDetails" /*Provide the URL of the web service to consume*/
13
14   constructor(private httpClient: HttpClient) { }
15
16   /*
17    * Consumes the web service exposed at the URL -> http://localhost:3020/planDetails
18    * After sending the request, the response must be an Observable
19    * Return the response back to the AddPlanComponent
20    */
21
22   addPlan(data: any): Observable<any> {
23     // Code here
24     return this.httpClient.post(this.url, data);
25   }
26
27   /*
28    * Consumes the web service exposed at the URL -> http://localhost:3020/planDetails
29    * After sending the request, the response must be an observable of array"
30    * Return the response back to the ViewPlanComponent
```

Sandip



```
TS app-routing.module.ts TS rest.service.ts X TS unit.pipe.ts TS app.component.ts app.component.html TS add TS ...
Frontend > src > app > shared > TS rest.service.ts > PlanService > deletePlan
41 Return the response back to the PlanDetailsComponent
42 */
43
44 getPlanDetails(planId) :Observable<Plan>{
45 // Code here
46 return this.httpClient.get<Plan>(`${this.url}/${planId}`);
47 }
48
49
50
51 /*
52 Consumes the web service exposed at the URL -> http://localhost:3020/planDetails/:planId
53 After sending the request, the response must be an Observable
54 Return the response back to the PlanDetailsComponent
55 */
56
57 deletePlan(planId) {
58 // Code here
59 return this.httpClient.delete(`${this.url}/${planId}`);
60 }
61
62
63
```

Ln 39, Col 60 Spaces: 2 UTF-8 LF TypeScript 3.8.3 Execute View Result

S

Sandip

You



```
TS app-routing.module.ts x TS rest.service.ts TS unit.pipe.ts TS app.component.ts app.component.html TS add ...
Frontend > src > app > TS app-routing.module.ts > ...
6
7  /*
8   Add the route paths to navigate to appropriate
9   1. /addPlan -> AddPlanComponent
10  2. /viewPlan -> ViewPlanComponent
11  3. In case an Invalid URL is provided, the user must be redirected to the addPlan page.
12  */
13
14
15  export const routes: Routes = [
16    {path: 'addPlan', component: AddPlanComponent},
17    {path: 'viewPlan', component: ViewPlanComponent},
18    {path: '**', redirectTo: '/addPlan', pathMatch: 'full'}
19  ];
20
21  @NgModule({
22    imports: [RouterModule.forRoot(routes)],
23    exports: [RouterModule]
24  })
25  export class AppRoutingModule { }
26
```