

Práctica 4.1: Servicio Mock TipoSalas

Rama git: p_mock

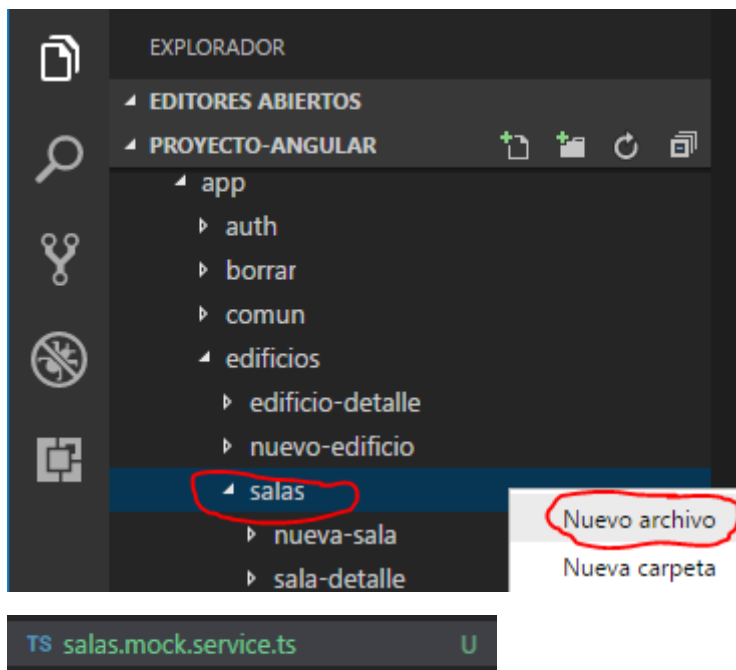
Rama resuelta: p_mock_resuelta

Usar un servicio mock ya creado: **salas.mock.service.ts** que proporciona los tipos de sala (id y descripción)

Injectar el servicio en **nueva-sala.component.ts** para que lo utilice el desplegable de tipo sala

Solución:

1. Ya está hecho: Hemos creado un nuevo servicio, nos interesa crearlo en la carpeta “/edificios/salas” porque es un servicio para salas:



2. Ya está hecho: Creamos lo básico del servicio:

```
import { Injectable } from '@angular/core';
@Injectable()
export class SalasMockService{ }
```

```

TS salas.mock.service.ts x
1  import { Injectable } from '@angular/core';
2
3  @Injectable()
4
5  export class SalasMockService {}
6
7  }

```

3. Ya está hecho: Hemos importamos el modelo "TipoSala" de api-rest:

```
import { TipoSala } from '../../api-rest/';
```

4. Ya está hecho: Hemos definido una propiedad privada "tipos" con la interfaz TipoSala y le hemos asignado el array que queremos devolver en formato json:

```

private tipos: TipoSala[] = [
  {"id":1, "tipo":'Auditorio'},
  {"id":2,"tipo":'Sala de reuniones'},
  {"id":3,"tipo":'Despacho'}
];

```

```

TS salas.mock.service.ts x
1  import { Injectable } from '@angular/core';
2  import { TipoSala } from '../../api-rest/';
3
4  @Injectable()
5
6  export class SalasMockService {
7      private tipos: TipoSala[] = [
8          {"id":1, "tipo":'Auditorio'},
9          {"id":2,"tipo":'Sala de reuniones'},
10         {"id":3,"tipo":'Despacho'}
11     ];
12
13     getTiposSalas(){
14         return this.tipos;
15     }
16 }

```

5. En **app.module.ts** importamos el servicio y lo definimos como "provider":

```
import { SalasMockService } from '../edificios/salas/salas.mock.service';
```

```
[...]
```

```

providers: [
  SalasMockService,

```

```
10 | import { SalasMockService } from '../edificios/salas/salas.mock.service';
```

```
72 | providers: [  
73 |   SalasMockService,
```

En el **componente** que va a consumir el servicio: **nueva-sala.component.ts**

6. Importamos el **servicio** y la **interfaz** TipoSala de api-rest models

```
import { SalasMockService } from  
'../../../../../edificios/salas/salas.mock.service';  
  
import { TipoSala } from '../../../api-rest/';
```

```
4 | // Práctica mock:  
5 | // Alguien debería importar la interfaz TipoSala de api-rest models  
6 | import { TipoSala } from '../../../api-rest/';  
7 | import { DefaultService } from '../../../api-rest/';  
8 | // Práctica mock:  
9 | // Alguien debería importar el servicio Mock  
10 | import { SalasMockService } from '../../../../../edificios/salas/salas.mock.service';
```

7. Lo instanciamos en el constructor y hacemos una llamada al servicio guardándolo en una propiedad de TipoSala:

```
constructor(private salasMockService: SalasMockService, [...] {  
    this.tiposSalas = this.salasMockService.getTiposSalas();
```

```
24 | constructor(private salasMockService: SalasMockService, private  
25 |   this.tiposSalas = this.salasMockService.getTiposSalas();  
26 | }
```

En el componente **nueva-sala.component.html**

8. En el **<mat-select>** **<mat-option>** cargamos los valores de **tiposSalas** con un ***ngFor**

```
<mat-option *ngFor="let tipo of tiposSalas" [value]="tipo.id">  
  {{ tipo.tipo }}  
</mat-option>
```

```
<div formGroupName="tipoSala" fxFlex="15%">  
  <mat-form-field>  
    <mat-select formControlName="id" placeholder="Tipo de Sala" required>  
      <mat-option *ngFor="let tipo of tiposSalas" [value]="tipo.id">  
        {{ tipo.tipo }}  
      </mat-option>  
    </mat-select>  
  </mat-form-field>  
</div>
```