

## Práctica 4.2: Servicio Mock Observable

Rama git: p\_observable

Rama resuelta: p\_observable\_resuelta

En la Carpeta “Solucion Practicas” hay una solución paso a paso de cada práctica

Usar el servicio mock ya creado: salas.mock.service.ts y hacer que el método `getTiposSalas()`

devuelva un Observable

### Solución

En `sala.mock.service.ts`:

1. Importamos Observable de rxjs

```
import {Observable} from 'rxjs';
```

2. En el método `geteTiposSalas()` devolvemos un **Observable.of** con los tipos y añadimos **.delay(2000)** para retrasar la respuesta 2 segundos:

```
return Observable.of(this.tipos).delay(2000);
```

```
TS salas.mock.service.ts x
1  import { Injectable } from '@angular/core';
2  import { TipoSala } from '../../../api-rest/';
3  // Práctica Observable:
4  // Alguien debería importar Observable de rxjs aquí:
5  import {Observable} from 'rxjs';
6
7  @Injectable()
8
9  export class SalasMockService {
10     private tipos: TipoSala[] = [
11         {"id":1, "tipo":'Auditorio'},
12         {"id":2,"tipo":'Sala de reuniones'},
13         {"id":3,"tipo":'Despacho'}
14     ];
15
16     getTiposSalas(){
17         // Utilizamos Observable.of porque ya tenemos los datos y su
18         // Práctica Observable:
19         // Alguien debería devolver un Observable.of con los tipos:
20         return Observable.of(this.tipos).delay(2000);
21     }
22 }
```

En **nueva-sala.component.ts**

3. Importamos Observable de rxjs

```
import {Observable} from 'rxjs';
```

4. Nos suscribimos al servicio y asignamos la respuesta a this.tipoSalas

```
this.salasMockService.getTiposSalas().subscribe(
    response => { this.tiposSalas = response; } );
```

TS nueva-sala.component.ts ✕

```
7 // Práctica Observable:
8 // Alguien debería importar Observable de rxjs aquí:
9 import {Observable} from 'rxjs/Observable';
10 import {Router, ActivatedRoute} from "@angular/router";
11 // Librerías para formularios reactivos:
12 import { FormGroup, FormControl, FormBuilder, Validators
13 import { MatSnackBar } from '@angular/material';
14
15 @Component({
16   selector: 'app-nueva-sala',
17   templateUrl: './nueva-sala.component.html',
18   styleUrls: ['./nueva-sala.component.css']
19 })
20 export class NuevaSalaComponent implements OnInit {
21   sala = {} as Sala;
22   tiposSalas: TipoSala[] = [];
23   idEdificio: number;
24   private salaForm: FormGroup;
25   errorMessage: string;
26
27   constructor(private salasMockService: SalasMockService,
28     // Práctica Observable:
29     // Alguien debería suscribirse al servicio y asignar
30     this.salasMockService.getTiposSalas().subscribe(
31       response => {
32         this.tiposSalas = response;
33       }
34     );
35   }
```