Práctica 4.1: Servicio Mock TipoSalas

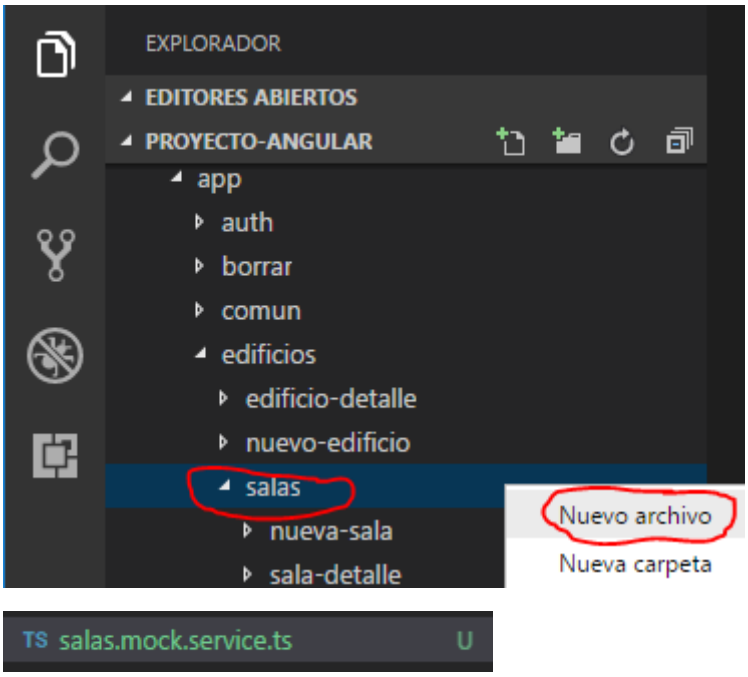
Rama git: p_mock **Rama resuelta: p_mock_resuelta**

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

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

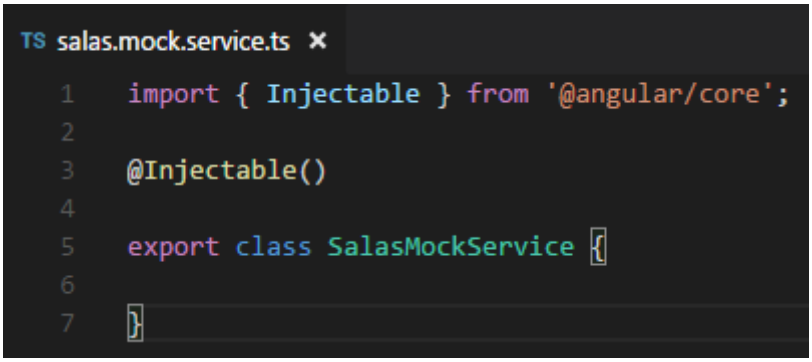
Solución:

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{ }
```

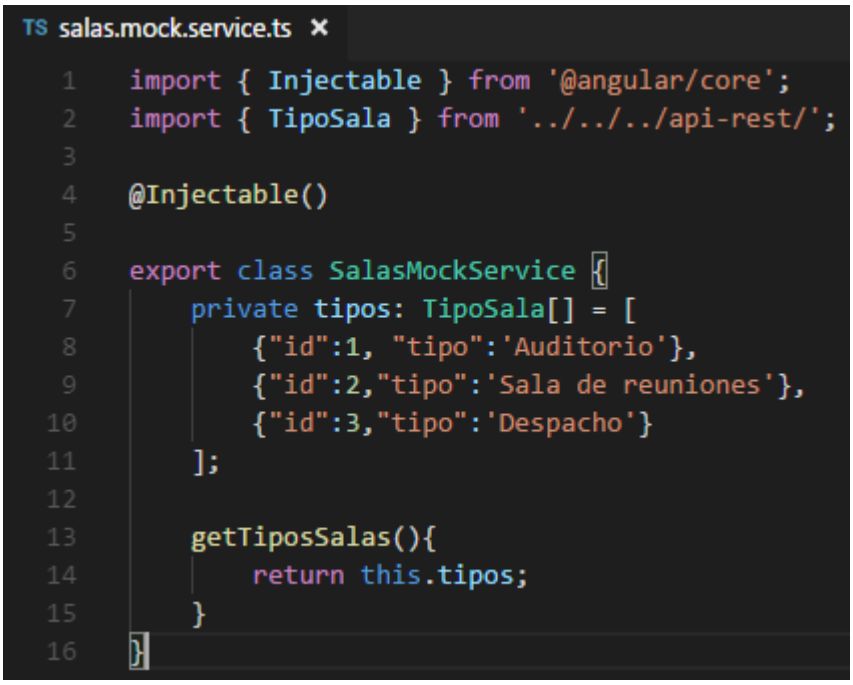


3. Ya está hecho: Hemos importamos el modelo "TipoSala" de api-rest:

```
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'}
];
```



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

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

[...]

providers: [
    SalasMockService,
```

```
providers: [
  SalasMockService,
```

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

En el **componente** que va a consumir el servicio: **nueva-sala.componente.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/';
```

```
// Práctica mock:
// Alguien debería importar la interfaz TipoSala de api-rest models
import { TipoSala } from '../../../../api-rest/';
import { DefaultService } from '../../../../api-rest';
// Práctica mock:
// Alguien debería importar el servicio Mock
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();
```

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

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>
```

nueva-sala.component.html

```
<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>
```