SPRING BOOT + ANGULAR +STRIPE

# PAGOS CON STRIPE EN SPRING BOOT Y ANGULAR 8. CAPÍTULO 6: ARTÍCULOS

En este post crearemos las vistas de los artículos (la lista y el detalle):  
**Artículos (servicio):**

Primero vamos al servicio el servicio de artículos (articulo.service.ts). Crearemos solamente dos métodos, uno para obtener la lista y otro para ver un artículo en concreto (detalle).

|  |  |
| --- | --- |
|  | import { Injectable } from '@angular/core'; |
|  | import { HttpClient, HttpHeaders } from '@angular/common/http'; |
|  | import { Observable } from 'rxjs'; |
|  | import { Articulo } from '../model/articulo'; |
|  |  |
|  | const cabecera = {headers: new HttpHeaders({'Content-Type': 'application/json'})}; |
|  |  |
|  | @Injectable({ |
|  | providedIn: 'root' |
|  | }) |
|  | export class ArticuloService { |
|  |  |
|  | articuloURL = 'http://localhost:8080/articulo/'; |
|  |  |
|  | constructor(private httpClient: HttpClient) { } |
|  |  |
|  | public lista(): Observable<Articulo[]> { |
|  | return this.httpClient.get<Articulo[]>(this.articuloURL + 'lista', cabecera); |
|  | } |
|  |  |
|  | public detalle(id: number): Observable<Articulo> { |
|  | return this.httpClient.get<Articulo>(this.articuloURL + `detalle/${id}`, cabecera); |
|  | } |
|  | } |

[**view raw**](https://gist.github.com/cavanosa/2749ae09b84e0356a12b5167df5af636/raw/488fd086f55a34f0a8c7b577742838763a402d19/articulo.service.ts)[**articulo.service.ts**](https://gist.github.com/cavanosa/2749ae09b84e0356a12b5167df5af636#file-articulo-service-ts) hosted with  by [**GitHub**](https://github.com/)

Y ojo con las comillas del método detalle, si le pasamos algún parámetro -el id en nuestro caso:

OK: this.articuloURL + `detalle/${id}`, cabecera

MAL: this.articuloURL + 'detalle/${id}', cabecera

**Artículos (lista):**

Modificamos el archivo lista-articulo.component.ts:

|  |  |
| --- | --- |
|  | import { Component, OnInit } from '@angular/core'; |
|  | import { Articulo } from '../model/articulo'; |
|  | import { ArticuloService } from './articulo.service'; |
|  |  |
|  | @Component({ |
|  | selector: 'app-lista-articulo', |
|  | templateUrl: './lista-articulo.component.html', |
|  | styleUrls: ['./lista-articulo.component.css'] |
|  | }) |
|  | export class ListaArticuloComponent implements OnInit { |
|  |  |
|  | articulos: Articulo[] = []; |
|  |  |
|  | constructor(private articuloService: ArticuloService) { } |
|  |  |
|  | ngOnInit() { |
|  | this.cargarLista(); |
|  | } |
|  |  |
|  | cargarLista(): void { |
|  | this.articuloService.lista().subscribe( |
|  | data => { |
|  | this.articulos = data; |
|  | }, |
|  | err => { |
|  | console.log(err); |
|  | } |
|  | ); |
|  | } |
|  |  |
|  | } |

[**view raw**](https://gist.github.com/cavanosa/db5576a5d52fde73f04216765e1ad929/raw/42dc4eaecd4f5a7db6b102270eed436bc8b56a8b/lista-articulo.component.ts)[**lista-articulo.component.ts**](https://gist.github.com/cavanosa/db5576a5d52fde73f04216765e1ad929#file-lista-articulo-component-ts) hosted with  by [**GitHub**](https://github.com/)

Sólo hay un método para cargar la lista, que se va a guardar en el array Articulos. Como el servicio devuelve un observable pues no nos queda otra que consumir el servicio por medio del método subscribe.

Y la vista (lista-articulo.component.html):

|  |  |
| --- | --- |
|  | <table class="table table-dark"> |
|  | <thead> |
|  | <tr> |
|  | <th scope="col">Nombre</th> |
|  | <th scope="col">Descripción</th> |
|  | <th scope="col">Precio</th> |
|  | <th scope="col">Comprar</th> |
|  | </tr> |
|  | </thead> |
|  | <tbody> |
|  | <tr \*ngFor="let articulo of articulos"> |
|  | <td>{{articulo.nombre}}</td> |
|  | <td>{{articulo.descripcion}}</td> |
|  | <td>{{articulo.precio / 100}}€</td> |
|  | <td><button class="btn btn-outline-info" routerLink="/detalle/{{articulo.id}}">Comprar</button></td> |
|  | </tr> |
|  | </tbody> |
|  | </table> |

[**view raw**](https://gist.github.com/cavanosa/5fbcd3582da26f4145fc1a924e8f19f5/raw/33da79d6616535fbeec608615b445329c7c9c0a1/lista-articulo.component.html)[**lista-articulo.component.html**](https://gist.github.com/cavanosa/5fbcd3582da26f4145fc1a924e8f19f5#file-lista-articulo-component-html) hosted with  by [**GitHub**](https://github.com/)

**Artículos (detalle):**

Vamos al archivo detalle-articulo.component.ts:

|  |  |
| --- | --- |
|  | import { Component, OnInit } from '@angular/core'; |
|  | import { Articulo } from '../model/articulo'; |
|  | import { ArticuloService } from './articulo.service'; |
|  | import { ActivatedRoute } from '@angular/router'; |
|  |  |
|  | @Component({ |
|  | selector: 'app-detalle-articulo', |
|  | templateUrl: './detalle-articulo.component.html', |
|  | styleUrls: ['./detalle-articulo.component.css'] |
|  | }) |
|  | export class DetalleArticuloComponent implements OnInit { |
|  |  |
|  | articulo: Articulo; |
|  |  |
|  | constructor(private articuloService: ArticuloService, private activatedRoute: ActivatedRoute) { } |
|  |  |
|  | ngOnInit() { |
|  | this.cargarArticulo(); |
|  | } |
|  |  |
|  | cargarArticulo(): void { |
|  | const id = this.activatedRoute.snapshot.params.id; |
|  | this.articuloService.detalle(id).subscribe( |
|  | data => { |
|  | this.articulo = data; |
|  | }, |
|  | err => { |
|  | console.log(err); |
|  | } |
|  | ); |
|  | } |
|  |  |
|  | } |

[**view raw**](https://gist.github.com/cavanosa/79ac099e8de861d6ac598bc3a6189dd8/raw/be7f3e9794cf07bd6fefa35cf96c768ad032580f/detalle-articulo.component.ts)[**detalle-articulo.component.ts**](https://gist.github.com/cavanosa/79ac099e8de861d6ac598bc3a6189dd8#file-detalle-articulo-component-ts) hosted with  by [**GitHub**](https://github.com/)

Cargamos el artículo a partir del id que va en la url por medio de un objeto ActivatedRoute.

En la vista mostraremos un card en el que se incluirá el formulario de pago:

|  |  |
| --- | --- |
|  | <div class="container mt-3"> |
|  | <div class="card p-3" style="width: 28rem;" \*ngIf="articulo"> |
|  | <img src="{{articulo.imagenURL}}" class="card-img-top mx-auto" alt="..." style="max-height: 200px; width: auto"> |
|  | <div class="card-body mx-auto"> |
|  | <h5 class="card-title">{{articulo.nombre}}</h5> |
|  | <p class="card-text">{{articulo.descripcion}}</p> |
|  | <p><b>Precio: </b>{{articulo.precio / 100}}€</p> |
|  | </div> |
|  | <div class="card-body"> |
|  | <app-payment></app-payment> |
|  | </div> |
|  | </div> |
|  | </div> |

[**view raw**](https://gist.github.com/cavanosa/01cb34b87afd3cc00cfc1458b1c66347/raw/e49780e80f260a340bd2bdca740495a6c5e8b74e/detalle-articulo.component.html)[**detalle-articulo.component.html**](https://gist.github.com/cavanosa/01cb34b87afd3cc00cfc1458b1c66347#file-detalle-articulo-component-html) hosted with  by [**GitHub**](https://github.com/)

De momento el componente payment no está implementado así que mostrará el mensaje payment works!!



El pago lo implementamos ya en el siguiente capítulo.