



Doc - Frontend Instrumentos - Angular

Repositorios

SZDohmen/Instrumentos-Backend

Lab4 - Generic Backend for "Instrumentos TP". Contribute to SZDohmen/Instrumentos-Backend

<https://github.com/SZDohmen/Instrumentos-Backend>

SZDohmen/Instrumentos-Frontend-Angular

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh

<https://github.com/SZDohmen/Instrumentos-Frontend-Angular>

Creación del Proyecto

Instalación de Angular CLI de manera global en la pc:

```
npm install -g @angular/cli
```

Creamos nuestro proyecto llamado lab4-front-angular con:

```
ng new lab4-front-angular
```

Configuraciones

Archivo **tsconfig.json** → comentar, quitar o volver false el strict para evitar problemas con la entidad modelo que crearemos mas adelante

```
/*"strict": true*/
```

Vaciamos el archivo **app.component.html** y le agregamos:

```
<router-outlet></router-outlet>
```

Levantar la Api

Levantamos la aplicacion con:

```
ng serve -o
```

Bootstrap

Instalacion

```
npm install bootstrap
```

Agregarlo al **angular.json**

```
"styles": [  
  "node_modules/bootstrap/dist/css/bootstrap.min.css",
```

```
"src/styles.css"
],
```

Componentes y Carpetas

Carpeta **components**

Home

"Musical Hendrix es una tienda de instrumentos musicales con ya más de 15 años de experiencia. Tenemos el conocimiento y la capacidad como para informarte acerca de las mejores elecciones para tu compra musical."

Mapa

Insertar un mapa de Google con ubicación Av. Las Heras y Av. San Martin, Ciudad de Mendoza

Catalogo

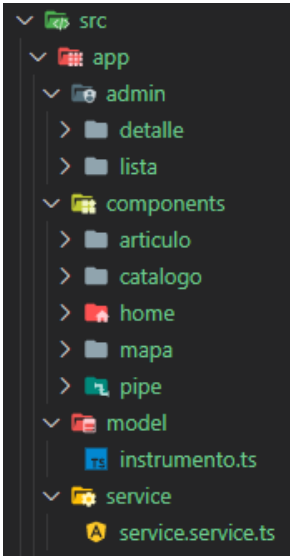
Grilla de cartas con el catalogo de productos/instrumentos

Articulo

Detalle del producto seleccionado

Pipe

Para la busqueda rapida del producto por su nombre en el Catalogo



Carpeta **admin**

lista

Tabla con los productos para su ABM

detalle

Detalle del producto para su ABM

Carpeta **model**

Entidad modelo del producto

Carpeta **service**

Servicio con conexion al servidor

Enrutado

Archivo **app-routing.module.ts** (solo la parte del routes).

```
const routes: Routes = [
  { path: '', component: HomeComponent },
  { path: 'donde-estamos', component: MapaComponent },
  { path: 'catalogo', component: CatalogoComponent },
  { path: 'catalogo/:id', component: ArticuloComponent },
  { path: 'admin/lista-productos', component: ListaComponent },
  { path: 'admin/detalle-producto/:id', component: DetalleComponent},
  { path:'*', pathMatch:'full', redirectTo: ''}
];
```

Modelo Entidad

Archivo **instrumentos.ts**

```
export class Instrumento {
  id:number;
  nombre:string;
  marca:string;
  imagen:string;
  precio:number;
  costoEnvio:String;
  cantVendida:number;
  descripcion:string;
}
```

Service

Conexion con el servidor con el archivo **service.service.ts** (solo la clase y su injectable).

```
@Injectable( {providedIn:'root'} )
export class Service {
  instrumento: Instrumento = new Instrumento();
  url='http://localhost:8080/lab4-back/instrumentos/';
  constructor(private http:HttpClient){}

  getAll():Observable<Instrumento[]> {
    return this.http.get<Instrumento[]>(this.url);
  }
  getOne(id:number):Observable<Instrumento> {
    return this.http.get<Instrumento>(this.url+id);
  }
  save(instrumento:Instrumento):Observable<Instrumento> {
    return this.http.post<Instrumento>(this.url, instrumento);
  }
  update(id:number, instrumento:Instrumento):Observable<Instrumento> {
    return this.http.put<Instrumento>(this.url+id, instrumento);
  }
  delete(id:number):Observable<Instrumento> {
    return this.http.delete<Instrumento>(this.url+id);
  }
}
```

Agregar el service al **app.module.ts** dentro de los providers e importar el **HttpClient**.

```
imports: [
  BrowserModule,
  AppRoutingModule,
  HttpClientModule,
],
providers: [Service],
bootstrap: [AppComponent]
```

Componente App

Archivo **app.component.html** contiene un navbar que redirecciona a los demas componentes principales de la aplicacion y la etiqueta del app-routing para que cambie el contenido segun la ruta en la que estamos.

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container-fluid">
    <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
      <div class="navbar-nav">
        <span class="navbar-text"
          style="color:red; font-size: 18px; margin-right:
            10px; font-family: Arial, Helvetica, sans-serif;">
          Musical Hendrix
        </span>
        <a class="nav-link active" aria-current="page" href="#">Home</a>
        <a class="nav-link" href="/catalogo">Catalogo</a>
        <a class="nav-link" href="/donde-estamos">Donde Estamos</a>
      </div>
      <div class="navbar-nav" style="position: absolute; margin: 0 auto; right:10px;">
        <a class="nav-link" href="/admin/lista-productos">Administracion</a>
      </div>
    </div>
  </div>
</nav>
```

```
<router-outlet></router-outlet>
```

Componente Home

Archivo **home.component.html** contiene una simple descripcion.

```
<div class="container" style="margin: 0 auto; font-size: 20px; margin-top: 50px;">
  <p>
    Musical Hendrix es una tienda de instrumentos musicales con ya más de 15 años de
    experiencia. Tenemos el conocimiento y la capacidad como para informarte acerca
    de las mejores elecciones para tu compra musical.
  </p>
</div>
```

Componente Mapa

Archivo **mapa.component.html** contiene un mapa de google con una direccion especifica.

```
<iframe
  title="mapadegoogle"
  style=" border: 0; width: 100%; height: 350px "
  src="https://maps.google.com/maps?q=-32.886153, -68.8386639&zoom=5&output=embed"
></iframe>
```

Componente Pipe

Archivo **filter.pipe.ts** filtra el nombre de un arreglo de instrumentos con el texto ingresado por el usuario.

```
@Pipe( {name: 'filter'} )
export class FilterPipe implements PipeTransform {
  transform(instrumento: Instrumento[], texto: string): Instrumento[] {
    if (texto.length === 0) {
      return instrumento;
    }
    texto = texto.toLocaleLowerCase();
    return instrumento.filter(
      instrumento => {
        return instrumento.nombre.toLocaleLowerCase().includes(texto);
      }
    );
  }
}
```

Agregarlo al **app.module.ts** dentro de declarations junto con los imports **FormsModule** y **ReactiveFormsModule**

Componente Catalogo

Archivo **catalogo.component.ts** trae un arreglo de instrumentos desde el servidor usando el service y redirige al detalle del instrumento seleccionado.

```
@Component({
  selector: 'app-catalogo',
  templateUrl: './catalogo.component.html',
  styleUrls: ['./catalogo.component.css']
})
export class CatalogoComponent implements OnInit {
  filterPost = '';
  instrumento: Instrumento[] = [];

  constructor(
    private service: Service,
    private router: Router
  ) { }

  ngOnInit(): void {
    this.getAll();
  }
}
```

```

    getAll(){
      this.service.getAll().subscribe( data => {
        this.instrumento = data;
      });
    }

    detail(id:number){
      this.router.navigate(['catalogo/'+ id]);
    }
  }
}
```

Archivo **catalogo.component.html** muestra una serie de cartas con la informacion traída. También tiene el filtro Pipe, donde un buscador filtra la lista de cartas con el texto ingresado

```

<div class="m-3">
  <form class="form-group">
    <input class="form-control margen-top-5" type="text" placeholder="Buscar instrumento..." style="width: 255px;"
      name="filterPost" [(ngModel)]="filterPost">
    </form>
  </div>

<div class="m-3">
  <div class="row">
    <div class="col-xs-4" *ngFor="let i of instrumento | filter:filterPost">
      <div class="card mb-3">
        <div class="card-body">
          <h3 class="card-title"> {{i.nombre}} </h3>
          <h6 class="card-subtitle"> {{i.marca}} </h6>
          <img class="card-imagen" [src]="i.imagen">
          <p class="card-text">$ {{i.precio}} </p>
          <p class="card-text" *ngIf="i.costoEnvio != 'G' " style="color:orange">
            Costo de Envio Interior de Argentina: ${{i.costoEnvio}}
          </p>
          <p class="card-text" *ngIf="i.costoEnvio == 'G' " style="color:green">
            Envio gratis a todo el pais
          </p>
          <button class="btn btn-primary m-l" (click)="detail(i.id)"> Ver detalles </button>
        </div>
      </div>
    </div>
  </div>
</div>
</div>
```

Componente Artículo

Archivo **articulo.component.ts** recibe el id enviado desde el catalogo para buscar los datos del instrumento seleccionado

```

export class ArticuloComponent implements OnInit {
  i: Instrumento = {
    id:null,
    nombre:null,
    marca:null,
    imagen:null,
    precio:null,
    costoEnvio:null,
    cantVendida:null,
    descripcion:null,
  }
  constructor(
    private service: Service,
    private rout: ActivatedRoute,
    private router: Router
  ) { }

  ngOnInit(): void {
    this.rout.params.subscribe(data => {
      if(data['id'] != '0'){
        this.getOne(data['id']);
      }
    });
  }

  getOne(id:number){
    this.service.getOne(id).subscribe( data => {
      this.i = data;
    });
  }

  volver() { this.router.navigate(['catalogo']); }
}
```

Archivo **articulo.component.html** muestra los todos datos del instrumento seleccionado en una carta

```
<div class="card">
  <div class="card-body">
    <h3 class="card-title"> {{i.nombre}} </h3>
    <h6 class="card-subtitle"> {{i.marca}} </h6>
    <img class="card-imagen" [src]="i.imagen">
    <p class="card-text">$ {{i.precio}} </p>
    <p class="card-text" *ngIf="i.costoEnvio != 'G' " style="color:orange">
      Costo de Envio Interior de Argentina: ${{i.costoEnvio}}
    </p>
    <p class="card-text" *ngIf="i.costoEnvio == 'G' " style="color:green">
      Envio gratis a todo el pais
    </p>
    <p class="card-text">
      Cantidad vendida: {{i.cantVendida}}
    </p>
    <p class="card-text"> {{i.descripcion}} </p>
    <button type="button" class="btn btn-primary m-2" (click)="volver()">Volver</button>
  </div>
</div>
```

Archivo **articulo.component.css** le da estilo a la carta

```
.card {
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);
  width: 500px;
  margin: 10px;
}
```

Componente Lista

Archivo **lista.component.ts** trae un listado de instrumentos para gestionar sus datos pudiendo eliminar el registro y crear un nuevo registro de instrumentos. Permite redireccionar al detalle del instrumento seleccionado (en caso de recibir un id diferente a 0), sino redirecciona al mismo formulario pero vacio.

```
@Component({
  selector: 'app-lista',
  templateUrl: './lista.component.html',
  styleUrls: ['./lista.component.css']
})
export class ListaComponent implements OnInit {
  instrumento: Instrumento[] = [];

  constructor(
    private service: Service,
    private router: Router
  ) { }

  ngOnInit(): void {
    this.getAll();
  }

  getAll(){
    this.service.getAll().subscribe( data => {
      this.instrumento = data;
    });
  }

  detail(id:number){
    this.router.navigate(['admin/detalle-producto/'+ id]);
  }

  delete(id:number){
    const opcion = confirm('¿Eliminar registro?');
    if(opcion === true){
      this.service.delete(id).subscribe( data => {
        alert('Registro eliminado');
        location.reload();
      });
    }
  }
}
```

Archivo **lista.component.html** muestra una tabla enlistando los registros.

```
<button class="btn btn-success m-3" (click)="detail(0)"> Nuevo Instrumento </button>
<br>
<div class="container" style="margin: 0 auto;">
  <table class="table table-striped">
    <thead>
      <tr>
        <th scope="col">id</th>
        <th scope="col">Nombre</th>
        <th scope="col">Marca</th>
        <th scope="col">Imagen Link</th>
        <th scope="col">Precio</th>
        <th scope="col">Costo del envio</th>
        <th scope="col">Cantidad vendida</th>
        <th scope="col">Descripcion</th>
        <th scope="col">Acciones</th>
      </tr>
    </thead>
    <tbody>
      <tr *ngFor="let i of instrumento">
        <td>{{i.id}}</td>
        <td>{{i.nombre}}</td>
        <td>{{i.marca}}</td>
        <td>{{i.imagen}}</td>
        <td>{{i.precio}}</td>
        <td>{{i.costoEnvio}}</td>
        <td>{{i.cantVendida}}</td>
        <td>{{i.descripcion}}</td>
        <td>
          <button class="btn btn-primary m-1" (click)="detail(i.id)"> Ver detalles </button>
          <button class="btn btn-danger m-2" (click)="delete(i.id)"> Borrar </button>
        </td>
      </tr>
    </tbody>
  </table>
</div>
```

Componente Detalle

Archivo **detalle.components.ts** trae los datos del registro seleccionado permitiendo actualizarlos. Sino permite la creacion de un nuevo registro en caso de que el id se detecte como '0'. Guarda lo traído desde el servidor en una interface temporal del mismo tipo que la entidad instrumento

```
@Component({
  selector: 'app-detalle',
  templateUrl: './detalle.component.html',
  styleUrls: ['./detalle.component.css']
})
export class DetalleComponent implements OnInit {

  instrumento: Instrumento = {
    id:null,
    nombre:null,
    marca:null,
    imagen:null,
    precio:null,
    costoEnvio:null,
    cantVendida:null,
    descripcion:null,
  }

  constructor(
    private service: Service,
    private rout: ActivatedRoute,
    private router: Router
  ) { }

  ngOnInit(): void {
    this.rout.params.subscribe(data => {
      if(data['id'] != '0'){
        this.getOne(data['id']);
      }
    });
  }

  getOne(id:number){
    this.service.getOne(id).subscribe( data => {
      this.instrumento = data;
    });
  }

  detail(){
    this.rout.params.subscribe( data => {
```

```

        (data['id'] === '0') ? this.save() : this.update(data['id']);
    });
    this.router.navigate(['admin/lista-productos']);
}

save(){
    this.service.save(this.instrumento).subscribe(data => {
        this.instrumento = data;
    });
}

update(id:number){
    this.service.update(id, this.instrumento).subscribe(data => {
        this.instrumento = data;
    });
}

canelar() { this.router.navigate(['admin/lista-productos']); }
}
```

Archivo **detalle.component.html** muestra los datos del registro (si es que hay)

```
<div class="card">
  <div class="card-body">
    <p class="title">Nombre</p>
    <input type="text" class="form-control" [(ngModel)]="instrumento.nombre">

    <p class="title">Marca</p>
    <input type="text" class="form-control" [(ngModel)]="instrumento.marca">

    <p class="title">Link de la Imagen</p>
    <input type="text" class="form-control" [(ngModel)]="instrumento.imagen">

    <p class="title">Precio</p>
    <input type="text" class="form-control" [(ngModel)]="instrumento.precio">

    <p class="title">Costo del Envio (gratis = G)</p>
    <input type="text" class="form-control" [(ngModel)]="instrumento.costoEnvio">

    <p class="title">Cantidad vendida</p>
    <input type="text" class="form-control" [(ngModel)]="instrumento.cantVendida">

    <p class="title">Descripcion</p>
    <textarea name="text" style="width:100%;" cols="2" rows="5"
    [(ngModel)]="instrumento.descripcion">
    </textarea>

    <button type="button" class="btn btn-success ml-2" (click)="detail()">Guardar</button>
    <button type="button" class="btn btn-danger m-2" (click)="canelar()">Cancelar</button>
  </div>
</div>
```