

Angular-Parte 2

En esta parte del curso se crea una App la cual gestiona información sobre restaurantes que interactúa con una base de datos MySQL para obtener, añadir, modificar o eliminar datos a través de un API en PHP y llamadas Ajax de tipo *observable*. Tanto para el listado como para el detalle y eliminación, se usará el método GET, para la edición o nuevos registros se usa el método POST.

Comunicaciones http en Angular 2

Estos dos artículos en castellano explican el concepto de los observables y las ventajas respecto AngularJS y promises.

<http://academia-binaria.com/comunicaciones-http-observables-con-angular2/>

<https://blog.enriqueoriol.com/2016/09/servicio-http-angular-2.html>

Este artículo (inglés) profundiza un poco más y comenta el tema de la cancelación de la petición con el método unsubscribe.

<https://jaxenter.com/reactive-programming-http-and-angular-2-124560.html>

Preparar API rest

El API rest está programado en PHP el cual espera ser llamado a través de Ajax, en función de la ruta especificada se solicita una interacción con la base de datos devolviendo el resultado en formato JSON.

Crear en www carpeta Slim y descomprimir api-rest.zip

En phpMyAdmin ejecutar el siguiente sql y añadir 4 registros

```
CREATE DATABASE IF NOT EXISTS webapp;  
USE webapp;  
  
CREATE TABLE restaurantes(  
  id          int(255) auto_increment not null,  
  nombre      varchar(255),  
  direccion   TEXT,  
  descripcion TEXT,  
  imagen      varchar(255),  
  precio      varchar(255),  
  CONSTRAINT pk_restaurantes PRIMARY KEY(id)  
)ENGINE=InnoDB;
```

Se testea: <http://localhost/slim/restaurantes-api.php/restaurantes> y devuelve un Json con los restaurantes.

Acentos y caracteres especiales.

Con tal de evitar errores/problemas con la codificación UTF8, añadir esta línea justo debajo de la conexión a la BBDD en el archivo restaurantes-api.php

```
$setUtf8= $db->query("SET CHARACTER SET utf8;");
```

Preparando el App

Main.ts

```
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { AppModule } from './app.module';
const platform = platformBrowserDynamic();
platform.bootstrapModule(AppModule);
```

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { HttpClientModule, JsonpModule } from '@angular/http';
import { FormsModule } from '@angular/forms';

import { AppComponent } from './app.component';
import { routing, appRoutingProviders } from './app.routing';

import {RestaurantesListComponent} from "./components/restaurantes-list.component";
import {RestaurantesDetailComponent} from "./components/restaurantes-detail.component";
import {RestauranteAddComponent} from "./components/restaurante-add.component";
import {RestauranteEditComponent} from "./components/restaurante-edit.component";

@NgModule({
  imports:      [ BrowserModule, HttpClientModule, FormsModule, routing ],
  declarations: [
    AppComponent,
    RestaurantesListComponent,
    RestaurantesDetailComponent,
    RestauranteEditComponent,
    RestauranteAddComponent
  ],
  providers:    [ appRoutingProviders ],
  bootstrap:    [ AppComponent ]
})

export class AppModule { }
```

app.routing.ts

```
import { ModuleWithProviders } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import {RestaurantesListComponent} from "../components/restaurantes-list.component";
import {RestaurantesDetailComponent} from "../components/restaurantes-detail.component";
import {RestauranteAddComponent} from "../components/restaurante-add.component";
import {RestauranteEditComponent} from "../components/restaurante-edit.component";

const appRoutes: Routes = [
  {
    path: '',
    redirectTo: '/',
    pathMatch: 'full'
  },
  {path: "", component: RestaurantesListComponent},
  {path: "restaurante/:id", component: RestaurantesDetailComponent},
  {path: "crear-restaurante", component: RestauranteAddComponent},
  {path: "editar-restaurante/:id", component: RestauranteEditComponent},
  {path: "donde-como-hoy/:random", component:
RestaurantesDetailComponent},
];
export const appRoutingProviders: any[] = [];
export const routing: ModuleWithProviders = RouterModule.forRoot(appRoutes);
```

Crear views y components en blanco

App.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Router, ActivatedRoute, Params } from '@angular/router';

@Component({
  selector: "mi-app",
  templateUrl: "app/view/home.html"
})
export class AppComponent{
  public titulo:string = "Restaurantes";
}
```

Home.html

```
{{titulo}}
<nav id="nav">
  <ul>
    <li><a [routerLink]="['/']">Inicio</a></li>
    <li><a [routerLink]="['/crear-restaurante']">Crear
restaurantes</a></li>
    <li><a [routerLink]="['/donde-como-hoy', 'true']">¿Donde como
hoy?</a></li>
  </ul>
</nav>
<section class="main">
  <router-outlet></router-outlet>
</section>
```

Restaurants-list.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Router, ActivatedRoute, Params } from '@angular/router';

@Component({
  selector: "restaurantes-list",
  templateUrl: "app/view/restaurantes-list.html"
})
export class RestaurantesListComponent {
  private titulo:string;

  constructor(){
    this.titulo="Listado"
  }
}
```

restaurantes-list.html

```
{{titulo}}
```

El resto de componententes/vistas son:

RestauranteAddComponent-restaurante-add.html

RestauranteEditComponent-restaurante-edit.html

RestaurantesDetailComponent-restaurante-detail.html

Model

Crear modelo de datos para restaurantes

```
export class Restaurante{
  constructor(
    public id:number,
    public nombre:string,
    public direccion:string,
    public descripcion:string,
    public imagen:string,
    public precio:string
  ){}
}
```

Services

En este archivo se irán añadiendo todos los servicios que se irán usando en la App, para empezar, se declara el método `getRestaurantes` que posteriormente es llamado desde el componente listado a través del método `Subscribe` (petición Ajax de tipo `Observable`)

```
import {Injectable} from "@angular/core";
import {Http, Response, Headers} from "@angular/http";
import "rxjs/add/operator/map";
import {Observable} from "rxjs/Observable";
import {Restaurante} from "../model/restaurante";

@Injectable()
export class RestauranteService{
    constructor(private _http: Http){}

    getRestaurantes(){
        return this._http.get("http://localhost/slim/restaurantes-api.php/restaurantes")
            .map(res => res.json());
    }
}
```

Listado de restaurantes (restaurantes-list.component.ts)

```
import {RestauranteService} from "../services/restaurante.service";
import {Restaurante} from "../model/restaurante";

@Component({
    selector: "restaurantes-list",
    templateUrl: "app/view/restaurantes-list.html",
    providers: [RestauranteService]
})
export class RestaurantesListComponent implements OnInit {
    private titulo:string;
    public restaurantes: Restaurante[];
    public status: string;
    public errorMessage;
    public confirmado;

    constructor(
        private _route: ActivatedRoute,
        private _router: Router,
        private _restauranteService: RestauranteService
    ){
        this.titulo="Listado"
    }
    ngOnInit() {
        this.getRestaurantes();
        console.log("restaurantes-list component cargado");
    }

    getRestaurantes(){

        this._restauranteService.getRestaurantes()
            .subscribe(
                result => {
```

```
this.restaurantes = result.data;

this.status = result.status;


if(this.status !== "success"){
    alert("Error en el servidor");
}


},
error => {

this.errorMessage = <any>error;

if(this.errorMessage !== null){
    console.log(this.errorMessage);
    alert("Error en la petición");
}

});
}
```

Comprobar a través de la consola que la petición al API se realiza

Se edita el template para mostrar listado

```
{{titulo}}
<ul id="restaurantes-list">
  <li *ngFor="let restaurante of restaurantes" class="restaurante-item">
    <div class="image">
      
    </div>
    <div class="data">
      <h4>{{restaurante.nombre}}</h4>
      <p>{{restaurante.direccion}}</p>
      <p>
        <a [routerLink]="['/restaurante', restaurante.id ]"
class="btn btn-ver">Ver</a>
        <a [routerLink]="['/editar-restaurante',
restaurante.id ]" href="" class="btn btn-editar">Editar</a>
        <a (click)="onBorrarConfirm(restaurante.id)"
class="btn btn-eliminar">Eliminar</a>
      </p>
      <div class="seguro" *ngIf="confirmado === restaurante.id">
        <a (click)="onBorrarRestaurante(restaurante.id)"
class="btn btn-eliminar">Eliminar</a>
        <a (click)="onCancelarConfirm(restaurante.id)"
class="btn btn-ver">Cancelar</a>
      </div>
    </div>
  </li>
</ul>
```

```

        </div>
      </div>

    </li>
  </ul>

```

Insertar loader

Declarar variable publica loading.

```
public loading;
```

En el ngOnInit

```
this.loading = 'show';
```

En el template, antes del primer li.

```

<li *ngIf="loading == 'show'" >
    
</li>

```

En la petición Ajax se oculta con

```
this.loading = 'hide';
```

Detalle de restaurante.

Se añade el servicio de obtener restaurante.

```
getRestaurante(id: string){
    return this._http.get("http://localhost/slim/restaurantes-
api.php/restaurante/"+id)
    .map(res => res.json());
}
```

En el componente-detail

```
import { Component, OnInit } from '@angular/core';
import { Router, ActivatedRoute, Params } from '@angular/router';
import { RestauranteService } from "../services/restaurante.service";
import { Restaurante } from "../model/restaurante";

@Component({
  selector: "restaurantes-detail",
  templateUrl: "app/view/restaurante-detail.html",
  providers: [RestauranteService]
})
export class RestaurantesDetailComponent implements OnInit {
  public restaurante: Restaurante[];
  public errorMessage: string;
  public status: string;
  constructor(
    private _restauranteService: RestauranteService,
    private _route: ActivatedRoute,
    private _router: Router
  ){}
  ngOnInit(){
    this.getRestaurante();
  }
  getRestaurante(){
    this._route.params.forEach((params: Params) => {
      let id = params["id"];
      this._restauranteService.getRestaurante(id)
        .subscribe(
          response => {
            this.restaurante = response.data;
            this.status = response.status;
            if(this.status !== "success"){
              this._router.navigate(["Home"]);
            }
          },
          error => {
            this.errorMessage = <any>error;
            if(this.errorMessage !== null){
              console.log(this.errorMessage);
              alert("Error en la petición");
            }
          }
        );
  }
  }
}
```


El template.

```
{{titulo}}
<div *ngIf="restaurante" class="restaurante-box">
  
  <div class="data-rest">
    <h1>Restaurante: {{restaurante.nombre}}</h1>
    <p class="precio">
      <span [ngSwitch]="restaurante.precio">
        <span *ngSwitchCase="'bajo'">Entre 2.50€ y 6€</span>
        <span *ngSwitchCase="'normal'">Entre 4.50€ y 9€</span>
        <span *ngSwitchCase="'alto'">Entre 7.50€ y 20€</span>
      </span>
    </p>
    <p class="direccion">{{restaurante.direccion}}</p>
    <p class="descripcion">{{restaurante.descripcion}}</p>
  </div>
</div>
```

Añadir Restaurante

En el servicio

```
addRestaurante(restaurante: Restaurante) {
    let json = JSON.stringify(restaurante);
    let params = "json="+json;
    let headers = new Headers({'Content-Type':'application/x-www-
form-urlencoded'});

    return this._http.post("http://localhost/slim/restaurantes-
api.php/restaurantes",
        params, {headers: headers}).map(res => res.json());
}
```

El componente

```
public titulo = "Crear nuevo restaurante";
public restaurante: Restaurante;
public errorMessage: string;
public status: string;

public filesToUpload: Array<File>;

constructor(
    private _restauranteService: RestauranteService,
    private _route: ActivatedRoute,
    private _router: Router
){}

onSubmit(){
    this._restauranteService.addRestaurante(this.restaurante).subscribe(
        response => {
            this.status = response.status;
            if(this.status !== "success"){
                alert("Error en el servidor");
            }
            this._router.navigate([""]);
        },
        error => {
            this.errorMessage = <any>error;

            if(this.errorMessage !== null){
                console.log("no
chuta"+this.errorMessage);
                alert("Error en la petición");
            }
            this._router.navigate([""]);
        }
    );
}

ngOnInit(){
    this.restaurante = new Restaurante(0,"","","","null","bajo");
}
```

```

callPrecio(value){
    this.restaurante.precio = value;
}

fileChangeEvent(fileInput: any){
    this.filesToUpload = <Array<File>>fileInput.target.files;

    this.makeFileRequest("http://localhost/slim/restaurantes-
api.php/upload-file", [], this.filesToUpload).then((result) => {
        this.resultUpload = result;
        this.restaurante.imagen =
this.resultUpload.filename;
    }, (error) =>{
        console.log(error);
    });
}

makeFileRequest(url: string, params: Array<string>, files:
Array<File>){
    return new Promise((resolve, reject) => {
        var formData: any = new FormData();
        var xhr = new XMLHttpRequest();

        for(var i = 0; i < files.length; i++){
            formData.append("uploads[]", files[i],
files[i].name);
        }

        xhr.onreadystatechange = function(){
            if(xhr.readyState == 4){
                if(xhr.status == 200){

                    resolve(JSON.parse(xhr.response));
                }else{
                    reject(xhr.response);
                }
            }
        }
        xhr.open("POST", url, true);
        xhr.send(formData);
    });
}

```

El template

```
<h1>{{titulo}}</h1>
<form #formRestaurante="ngForm" (ngSubmit)="onSubmit()">

<label>Nombre:</label>
<p>
    <input type="text" #nombre="ngModel" name="nombre"
    [(ngModel)]="restaurante.nombre" required/>
    <span *ngIf="!nombre.valid">El nombre es obligatorio</span>
</p>

<label>Dirección:</label>
<p>
    <input type="text" #direccion="ngModel" name="direccion"
    [(ngModel)]="restaurante.direccion" required/>
    <span *ngIf="!direccion.valid">La dirección es obligatoria</span>
</p>

<label>Descripción:</label>
<p>
    <textarea #descripcion="ngModel" name="descripcion"
    [(ngModel)]="restaurante.descripcion" required/>
    <span *ngIf="!descripcion.valid">La descripción es obligatoria</span>
</p>

<label>Precio:</label>
<p>
    <select [(ngModel)]="restaurante.precio" #precio="ngModel"
    name="precio">
        <option value="bajo" [attr.selected]="restaurante.precio ===
        'bajo' ? 'selected' : null">Bajo</option>
        <option value="medio" [attr.selected]="restaurante.precio ===
        'medio' ? 'selected' : null">Normal</option>
        <option value="alto" [attr.selected]="restaurante.precio ===
        'alto' ? 'selected' : null">Alto</option>
    </select>
</p>

<label>Imagen:</label>
<p>
    <input type="file" placeholder="Subir imagen..."
    (change)="fileChangeEvent($event)" />
</p>

<input type="submit" value="{{titulo}}"
    [disabled]="!formRestaurante.form.valid" />
</form>
```

Editar Restaurante

En el servicio

```
editRestaurante(id: string, restaurante: Restaurante) {
    let json = JSON.stringify(restaurante);
    let params = "json="+json;
    let headers = new Headers({'Content-Type':'application/x-www-form-urlencoded'});

    return this._http.post("http://localhost/slim/restaurantes-api.php/update-restaurante/"+id,
        params, {headers: headers}).map(res => res.json());
}
```

En el component

```
import {RestauranteService} from "../services/restaurante.service";
import {Restaurante} from "../model/restaurante";

@Component({
    selector: "restaurante-edit",
    templateUrl: "app/view/restaurante-add.html",
    providers: [RestauranteService]
})

export class RestauranteEditComponent implements OnInit {
    public titulo = "Editar restaurante";
    public restaurante: Restaurante;
    public errorMessage: string;
    public status: string;
    public filesToUpload: Array<File>;

    constructor(
        private _restauranteService: RestauranteService,
        private _route: ActivatedRoute,
        private _router: Router
    ){}

    onSubmit(){
        this._route.params.forEach((params: Params) => {
            let id = params["id"];
            this._restauranteService.editRestaurante(id,
this.restaurante).subscribe(
                response => {
                    this.status = response.status;
                    if(this.status !== "success"){
                        alert("Error en el servidor");
                    }
                    this._router.navigate(["/"]);
                },
                error => {
                    this.errorMessage = <any>error;

                    if(this.errorMessage !== null){
                        console.log(this.errorMessage);
                        alert("Error en la petición");
                    }
                }
            )
        })
    }
}
```

```

        });
    });
}

ngOnInit(){
    this.restaurante = new Restaurante(0,"","","","null","");
    this.getRestaurante();
}

getRestaurante(){
    this._route.params.forEach((params: Params) => {

        let id = params["id"];
        this._restauranteService.getRestaurante(id)
        .subscribe(
            response => {
                this.restaurante = response.data;
                this.status = response.status;

                if(this.status !== "success"){
                    this._router.navigate(["/"]);
                }

            },
            error => {
                this.errorMessage = <any>error;

                if(this.errorMessage !== null){
                    console.log(this.errorMessage);
                    alert("Error en la petición");
                }
            }
        );
    });
}

public resultUpload;

fileChangeEvent(fileInput: any){
    this.filesToUpload = <Array<File>>fileInput.target.files;

    this.makeFileRequest("http://localhost/slim/restaurantes-
api.php/upload-file", [], this.filesToUpload).then((result) => {
        this.resultUpload = result;
        this.restaurante.imagen =
this.resultUpload.filename;
    }, (error) =>{
        console.log(error);
    });
}

makeFileRequest(url: string, params: Array<string>, files:
Array<File>){
    return new Promise((resolve, reject) => {
        var formData: any = new FormData();
        var xhr = new XMLHttpRequest();
    });
}

```

```

        for(var i = 0; i < files.length; i++){
            formData.append("uploads[]", files[i],
files[i].name);
        }

        xhr.onreadystatechange = function(){
            if(xhr.readyState == 4){
                if(xhr.status == 200){

                    resolve(JSON.parse(xhr.response));
                }else{
                    reject(xhr.response);
                }
            }
        }
        xhr.open("POST", url, true);
        xhr.send(formData);
    });
}
}

```

Eliminar restaurante

En el servicio

```

deleteRestaurante(id: string){
    return this._http.get("http://localhost/slim/restaurantes-
api.php/delete-restaurante/"+id)
        .map(res => res.json());
}

```

En el componente listado

```

onBorrarConfirm(id){
    this.confirmado = id;
}

onCancelarConfirm(id){
    this.confirmado = null;
}

onBorrarRestaurante(id){

    this._restauranteService.deleteRestaurante(id)
        .subscribe(
            result => {
                this.status =
result.status;

                if(this.status !==
"success"){
                    alert("Error
en el servidor");
                }

                this.getRestaurantes();
            },
            error => {

```

```
<any>error;  
  
        if(this.errorMessage !==  
null){  
            console.log(this.errorMessage);  
            alert("Error en la  
petición");  
        }  
    }  
);  
}
```

En el template listado

En la navegación de cada restaurante

Eliminar

Se crear ventana confirmación

```
<div class="seguro" *ngIf="confirmado === restaurante.id">
  <a (click)="onBorrarRestaurante(restaurante.id)" class="btn btn-
  eliminar">Eliminar</a>
  <a (click)="onCancelarConfirm(restaurante.id)" class="btn btn-
  ver">Cancelar</a>
</div>
```