Luis Eduardo Peralta Martinez 2520029460101

Repositorio de GitHub: `https://github.com/LEPM7/JES-ANPRO-001-2019.git`

# JES-ANPRO-001-2019

El Ministerio Publico es una institución que cuenta con diversas fiscalías en todo el país, para lo cual tiene la necesidad de contar con la información de la ubicación física de cada una de ellas así como el número de teléfono. Información se deberá poder modificar en el transcurso del tiempo ya que cambien de ubicación constantemente.

**Requisitos backend**

- Instalar docker
- Instalar docker-compose

**Iniciar backend**

- `docker-compose up`
- En otra terminal ejecutar `docker exec -ti sql-server-db "bash"`
    - dentro del contenedor ejecutar `bash -c "/bin/bash /opt/bin/sql.sh"`

**Requisitos frontend**

- Instalar nodejs
- Instalar cliente de angular `npm install -g @angular/cli`

**Iniciar frontend**

- ng serve --open

**Screenshots funcionalidad**

**Archivos mas importantes**

```
package api;

import static spark.Spark.get;
import static spark.Spark.post;
import static spark.Spark.options;
import static spark.Spark.before;
import static spark.Spark.delete;
import static spark.Spark.put;

import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.databind.SerializationFeature;
import lombok.Data;

import java.io.IOException;
```

```
import java.io.StringWriter;
import java.sql.SQLException;
import java.util.LinkedList;

public class API {

    private static final int HTTP_BAD_REQUEST = 400;

    private static void enableCORS(final String origin, final String
methods, final String headers) {

        options("/*", (request, response) -> {

            String accessControlRequestHeaders = request.headers("Access-
Control-Request-Headers");
            if (accessControlRequestHeaders != null) {
                response.header("Access-Control-Allow-Headers",
accessControlRequestHeaders);
            }

            String accessControlRequestMethod = request.headers("Access-
Control-Request-Method");
            if (accessControlRequestMethod != null) {
                response.header("Access-Control-Allow-Methods",
accessControlRequestMethod);
            }

            return "OK";
        });

        before((request, response) -> {
            response.header("Access-Control-Allow-Origin", origin);
            response.header("Access-Control-Request-Method", methods);
            response.header("Access-Control-Allow-Headers", headers);
            // Note: this may or may not be necessary in your particular
application
            response.type("application/json");
        });
    }

    @Data
    static class FiscaliaPayload {
        private String nombre;
        private String descripcion;
        private String telefono;
        private String direccion;
        private Double latitud;
        private Double longitud;
    }

    //from spark docs
    public static String dataToJson(Object data) {
        try {
            ObjectMapper mapper = new ObjectMapper();
```

```
            mapper.enable(SerializationFeature.INDENT_OUTPUT);
            StringWriter sw = new StringWriter();
            mapper.writeValue(sw, data);
            return sw.toString();
        } catch (IOException e){
            throw new RuntimeException("IOException from a StringWriter?");
        }
    }

    public static void main(String[] args) {
        API.enableCORS("*", "*", "*");
        get("/fiscalias", (request, response) -> {
            System.out.println("GET /fiscalias");
            try {
                response.status(200);
                response.type("application/json");
                return dataToJson(new FakeORM().obtenerFiscaliasActivas());
            } catch (SQLException e) {
                response.status(HTTP_BAD_REQUEST);
                response.type("application/json");
                e.printStackTrace();
                return dataToJson(new LinkedList<>());
            }
        });

        //insertar nueva fiscalia
        post("/fiscalias", (request, response) -> {
            System.out.println("POST /fiscalias");
            try {
                ObjectMapper mapper = new ObjectMapper();
                FiscaliaPayload creation = mapper.readValue(request.body(),
FiscaliaPayload.class);
                new FakeORM().insert(creation.nombre, creation.descripcion,
creation.telefono,
                creation.direccion, creation.latitud, creation.longitud);
                response.status(200);
                response.type("application/json");
                return true;
            } catch (IOException | SQLException e) {
                response.status(HTTP_BAD_REQUEST);
                response.type("application/json");
                e.printStackTrace();
                return false;
            }
        });

        delete("/fiscalias/:id", (request, response) -> {
            System.out.println("DELETE /fiscalias/:id");
            try {
                new
FakeORM().delete(Integer.parseInt(request.params("id")));
                response.status(200);
                response.type("application/json");
                return true;
```

```
            } catch (SQLException e) {
                response.status(HTTP_BAD_REQUEST);
                response.type("application/json");
                e.printStackTrace();
                return false;
            }
        });

        put("/fiscalias/:id", (request, response) -> {
            System.out.println("PUT /fiscalias/:id");
            try {
                ObjectMapper mapper = new ObjectMapper();
                FiscaliaPayload creation = mapper.readValue(request.body(),
FiscaliaPayload.class);
                new
FakeORM().update(Integer.parseInt(request.params("id")), creation.nombre,
creation.descripcion, creation.telefono,
                creation.direccion, creation.latitud, creation.longitud);
                response.status(200);
                response.type("application/json");
                return true;
            } catch (IOException | SQLException e) {
                response.status(HTTP_BAD_REQUEST);
                response.type("application/json");
                e.printStackTrace();
                return false;
            }
        });
    }

}
```

```
package api;

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.HashMap;
import java.util.Map;
import java.util.List;
import java.util.stream.Collectors;

// TODO: usar un ORM real
public class FakeORM {

  String userName = "sa";
  String password = "Qwerty1234";
  String url = "jdbc:sqlserver://sql-server-db:1433;database=master";
```

```
  public Connection getConnection() throws SQLException {
    return DriverManager.getConnection(this.url, this.userName,
this.password);
  }

  // TODO: manejar transaccionalidad
  public boolean insert(String nombre, String descripcion, String telefono,
String direccion, Double latitud,
      Double longitud) throws SQLException {
    boolean result;
    CallableStatement pstmt = this.getConnection().prepareCall("{ call
dbo.FiscaliaInsertar(?,?,?,?,?,?)}");
    pstmt.setString(1, nombre);
    pstmt.setString(2, descripcion);
    pstmt.setString(3, telefono);
    pstmt.setString(4, direccion);
    pstmt.setDouble(5, latitud);
    pstmt.setDouble(6, longitud);
    result = pstmt.execute();
    return result;
  }

  // https://programmaticponderings.com/2012/08/24/calling-sql-server-
stored-procedures-with-java-using-jdbc/
  // http://sparkjava.com/tutorials/reducing-java-boilerplate
  public List obtenerFiscaliasActivas() throws SQLException {
    CallableStatement pstmt = this.getConnection().prepareCall("{ call
dbo.FiscaliaObtenerActivas()}");
    pstmt.execute();
    ResultSet rs = pstmt.getResultSet();
    Map<Integer,Fiscalia> fiscalias = new HashMap<Integer,Fiscalia>();
    if (rs != null) {
      while (rs.next()) {
        Fiscalia f = new Fiscalia(
          rs.getInt("fiscaliaId"),
          rs.getString("nombre"),
          rs.getString("descripcion"),
          rs.getString("telefono"),
          rs.getString("direccion"),
          rs.getDouble("latitud"),
          rs.getDouble("longitud"),
          rs.getBoolean("activa"));
          fiscalias.put(f.id, f);
      }
    }
    return fiscalias.keySet().stream().sorted().map((id) ->
fiscalias.get(id)).collect(Collectors.toList());
  }

  public boolean delete(Integer id) throws SQLException {
    boolean result;
    CallableStatement pstmt = this.getConnection().prepareCall("{ call
dbo.FiscaliaBorrar(?)}");
```

```
    pstmt.setInt(1, id);
    result = pstmt.execute();
    return result;
  }

  public boolean update(Integer id, String nombre, String descripcion,
String telefono, String direccion, Double latitud,
      Double longitud) throws SQLException {
    boolean result;
    CallableStatement pstmt = this.getConnection().prepareCall("{ call
dbo.FiscaliaUpdate(?,?,?,?,?,?,?)}");
    pstmt.setInt(1, id);
    pstmt.setString(2, nombre);
    pstmt.setString(3, descripcion);
    pstmt.setString(4, telefono);
    pstmt.setString(5, direccion);
    pstmt.setDouble(6, latitud);
    pstmt.setDouble(7, longitud);
    result = pstmt.execute();
    return result;
  }

}
```

**Front End**

```
import { Injectable } from '@angular/core';
import { HttpClient, HttpHeaders } from '@angular/common/http';
import { Fiscalia } from './fiscalia';
import { Observable, throwError } from 'rxjs';
import { retry, catchError } from 'rxjs/operators';

@Injectable({
  providedIn: 'root'
})
export class FiscaliasService {

  constructor(private http: HttpClient) { }

  // Http Headers
  httpOptions = {
    headers: new HttpHeaders({
      'Content-Type': 'application/json'
    })
  };

  // POST
  CreateFiscalia(data): Observable<boolean> {
    return this.http.post<boolean>('http://localhost:4567/fiscalias',
JSON.stringify(data), this.httpOptions)
    .pipe(
```

```
      retry(1),
      catchError(this.errorHandl)
    );
  }

  // GET
  GetFiscalias(): Observable<Fiscalia[]> {
    return this.http.get<Fiscalia[]>('http://localhost:4567/fiscalias')
    .pipe(
      retry(1),
      catchError(this.errorHandl)
    );
  }

  // DELETE
  DeleteFiscalia(id): Observable<boolean>{
    return this.http.delete<boolean>
(`http://localhost:4567/fiscalias/${id}`, this.httpOptions)
    .pipe(
      retry(1),
      catchError(this.errorHandl)
    );
  }

  //UPDATE
  UpdateFiscalia(id, data): Observable<boolean> {
    return this.http.put<boolean>(`http://localhost:4567/fiscalias/${id}`,
JSON.stringify(data), this.httpOptions)
    .pipe(
      retry(1),
      catchError(this.errorHandl)
    );
  }

  // Error handling
  errorHandl(error) {
     let errorMessage = '';
     if(error.error instanceof ErrorEvent) {
       // Get client-side error
       errorMessage = error.error.message;
     } else {
       // Get server-side error
       errorMessage = `Error Code: ${error.status}\nMessage:
${error.message}`;
     }
     console.log(errorMessage);
     return throwError(errorMessage);
  }

}
```

```
import { Component, OnInit, Output, EventEmitter, Input } from
'@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';
import { NgxSpinnerService } from 'ngx-spinner';
import { FiscaliasService } from '../shared/fiscalias.service';
import { Fiscalia } from '../shared/fiscalia';

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

  @Output() finish = new EventEmitter();
  @Input() fiscalia;
  isEdit = false;

  fiscaliaForm: FormGroup;

  constructor(private spinner: NgxSpinnerService, private fiscaliaService:
FiscaliasService) { }

  ngOnInit(): void {
    this.fiscaliaForm = new FormGroup({
      nombre: new FormControl('',
Validators.compose([Validators.required])),
      descripcion: new FormControl('',
Validators.compose([Validators.required])),
      direccion: new FormControl('',
Validators.compose([Validators.required])),
      telefono: new FormControl('',
Validators.compose([Validators.required])),
      latitud: new FormControl(0.0,
Validators.compose([Validators.required])),
      longitud: new FormControl(0.0,
Validators.compose([Validators.required])),
    });
    if (this.fiscalia) {
      this.isEdit = true;
      this.fiscaliaForm.get('nombre').setValue(this.fiscalia.nombre);

this.fiscaliaForm.get('descripcion').setValue(this.fiscalia.descripcion);
      this.fiscaliaForm.get('direccion').setValue(this.fiscalia.direccion);
      this.fiscaliaForm.get('telefono').setValue(this.fiscalia.telefono);
      this.fiscaliaForm.get('latitud').setValue(this.fiscalia.latitud);
      this.fiscaliaForm.get('telefono').setValue(this.fiscalia.longitud);
    }
  }


  save() {
    console.log(this.fiscaliaForm.getRawValue());
```

```
    this.spinner.show();
    if (!this.fiscaliaForm.valid) {
      alert('Por favor, llena el formulario');
    } else if (this.isEdit) {
      this.fiscaliaService.UpdateFiscalia(this.fiscalia.id,
this.fiscaliaForm.getRawValue())
      .subscribe((data) => {
        console.log(data);
        this.finish.emit('');
        this.spinner.hide();
      }, (e) => {
        alert('Ocurrio un error, intente de nuevo mas tarde');
        this.spinner.hide();
        console.log(e);
      });
    } else {
      this.spinner.show();
      this.fiscaliaService.CreateFiscalia(this.fiscaliaForm.getRawValue())
      .subscribe((data) => {
        console.log(data);
        this.finish.emit('');
        this.spinner.hide();
      }, (e) => {
        alert('Ocurrio un error, intente de nuevo mas tarde');
        console.log(e);
      });
    }
  }

  back() {
    this.finish.emit('');
  }

}
```

```
import { Component, OnInit, Output, EventEmitter } from '@angular/core';
import { Fiscalia } from '../shared/fiscalia';
import { FiscaliasService } from '../shared/fiscalias.service';
import { NgxSpinnerService } from 'ngx-spinner';

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

  fiscalias: Fiscalia[] = [];
  @Output() selectedFiscalia = new EventEmitter();
```

```
  constructor(private fiscaliasService: FiscaliasService, private spinner:
NgxSpinnerService) {}

  async ngOnInit() {
    this.getAllFiscalias();
  }

  getAllFiscalias(){
    this.spinner.show();
    this.fiscaliasService.GetFiscalias()
    .subscribe((fiscalias: Fiscalia[]) => {
      this.fiscalias = fiscalias;
      this.spinner.hide();
    }, (e) => {
      console.trace(e);
      this.spinner.hide();
    });
  }

  deleteFiscalia(id:number){
    this.spinner.show();
    this.fiscaliasService.DeleteFiscalia(id)
    .subscribe((data) => {
      if(!data){
        alert('Ocurrio un error al borrar, intenta mas tarde');
      }
      this.getAllFiscalias();
      this.spinner.hide();
    }, (e) => {
      console.trace(e);
      this.spinner.hide()
    });
  }

  editFiscalia(f){
    this.selectedFiscalia.emit(f);
  }
}
```