

Employee Management API

Una API REST construida con **Spring Boot**, **Java 17** y **MongoDB**, diseñada para administrar empleados.

Características

- Obtener todos los empleados
- Crear uno o varios empleados
- Actualizar empleados
- Eliminar empleados
- Bitácora de eventos de todas las operaciones (event_log)
- Manejo de excepciones personalizadas
- Integración con MongoDB local o Atlas
- Manejo de logs
- Muestra todos los headers en el log

Estructura del proyecto

technical_test/

```
├── src/
│   └── main/
│       ├── java/com/example/employee/
│       │   ├── controller/ # Controlador REST
│       │   ├── exception/ # Excepciones personalizadas
│       │   ├── model/ # Modelo de datos
│       │   ├── repository/ # Repositorio MongoDB
│       │   ├── logging/ # Clase para el filtro de los headers en el log
│       │   ├── service/ # Interfaz del servicio
│       │   └── service/impl/ # Implementación del servicio
│       ├── resources/
│       ├── application.yaml # Configuración Spring Boot
│       ├── logback-spring.xml # Configuración de Log
│       └── test/java/com/example/employee/controller # Controlador de los test
├── pom.xml
└── README.md
```

Endpoints disponibles

Employee

Método	Endpoint	Descripción
GET	/api/employee/{id}	Obtener un empleado
GET	/api/employees	Obtener todos los empleados
POST	/api/employees	Guardar uno o múltiples empleados
POST	/api/employee	Guardar un empleado
PUT	/api/employees/{id}	Actualizar un empleado
DELETE	/api/employees/{id}	Eliminar un empleado

Event_log

Método	Endpoint	Descripción
GET	/api/event-log	Todos los eventos
GET	/api/event-log/entity/{entity}	Eventos por tipo de entity
GET	/api/event-log/action/{action}	Eventos por tipo de action
GET	/api/event-log/entity/{entity}/action/{action}	Eventos por tipo de entity y por tipo de action

Pruebas unitarias

Las pruebas están ubicadas en:

```
src/test/java/com/example/employee/
```

Ejecutar con:

```
mvn test
```

Postman

En la carpeta addons se agrega el archivo

```
Employee Management API.postman_collection.json
```

El cuál nos permite importar las colecciones para realizar las pruebas directamente en Postman.

Ejecución

1. Requisitos previos

- Java 17 instalado
- MongoDB local corriendo (si no usas Atlas)
- Maven instalado

2. Clonar el proyecto

git clone <https://github.com/edu5975/EmployeeManagementAPI>

Nos movemos al directorio que acabamos de crear

3. Configuración local del entorno

Renombra el archivo que se encuentra en la carpeta:

src/main/resources/application-template.yml.

Al nombre application.yml

Reemplaza los siguientes valores:

`#{MONGO_URI}` Sustituir por la cadena de conexión a mongodb

4. Construir el proyecto

mvn clean package

5. Ejecutar la aplicación

java -jar target/employee-0.0.1-SNAPSHOT.jar

La aplicación quedará disponible en: <http://localhost:8080>

Ruta para utilizar Swagger: <http://localhost:8080/swagger-ui/index.html>

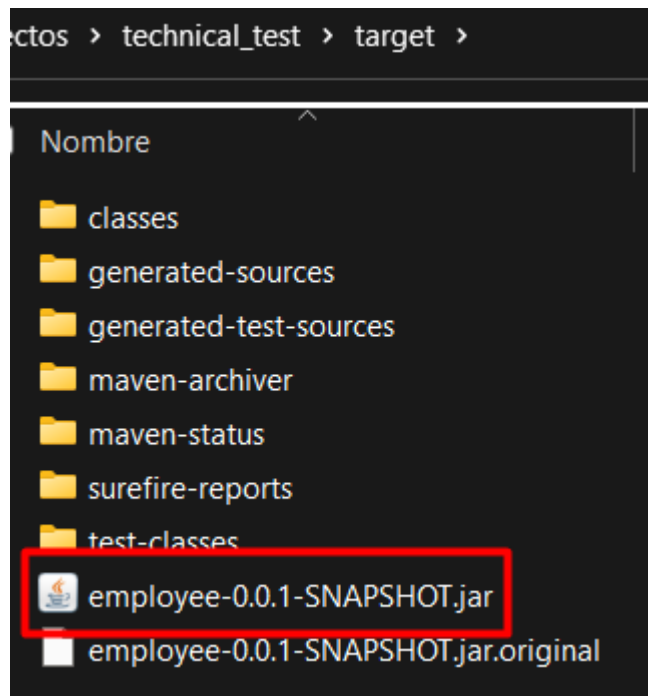
Evidencia de ejecución

Ejecución de pruebas

```
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.065 s
[INFO] Finished at: 2025-05-08T11:04:05-06:00
[INFO] -----
PS C:\Users\eduda\OneDrive\Documentos\Proyectos\technical_test>
```

Generación de .jar

```
[INFO] Building jar: C:\Users\eduda\OneDrive\Documentos\Proyectos\technical_test\target\employee-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:3.3.11:repackage (repackage) @ employee ---
[INFO] Replacing main artifact C:\Users\eduda\OneDrive\Documentos\Proyectos\technical_test\target\employee-0.0.1-SNAPSHOT.jar with repackaged archive, adding nested dependencies in BOOT-INF/.
[INFO] The original artifact has been renamed to C:\Users\eduda\OneDrive\Documentos\Proyectos\technical_test\target\employee-0.0.1-SNAPSHOT.jar.original
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 15.029 s
[INFO] Finished at: 2025-05-08T11:04:49-06:00
[INFO] -----
PS C:\Users\eduda\OneDrive\Documentos\Proyectos\technical_test>
```



Ejecución del .jar

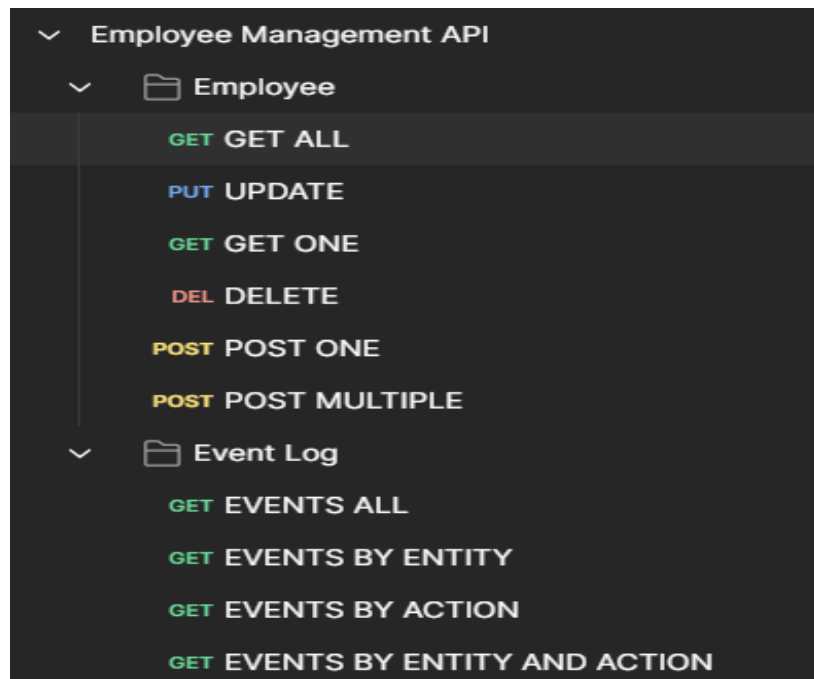
```
PS C:\Users\eduda\OneDrive\Documents\Proyectos\technical_test\target> java -jar .\employee-0.0.1-SNAPSHOT.jar
```

```

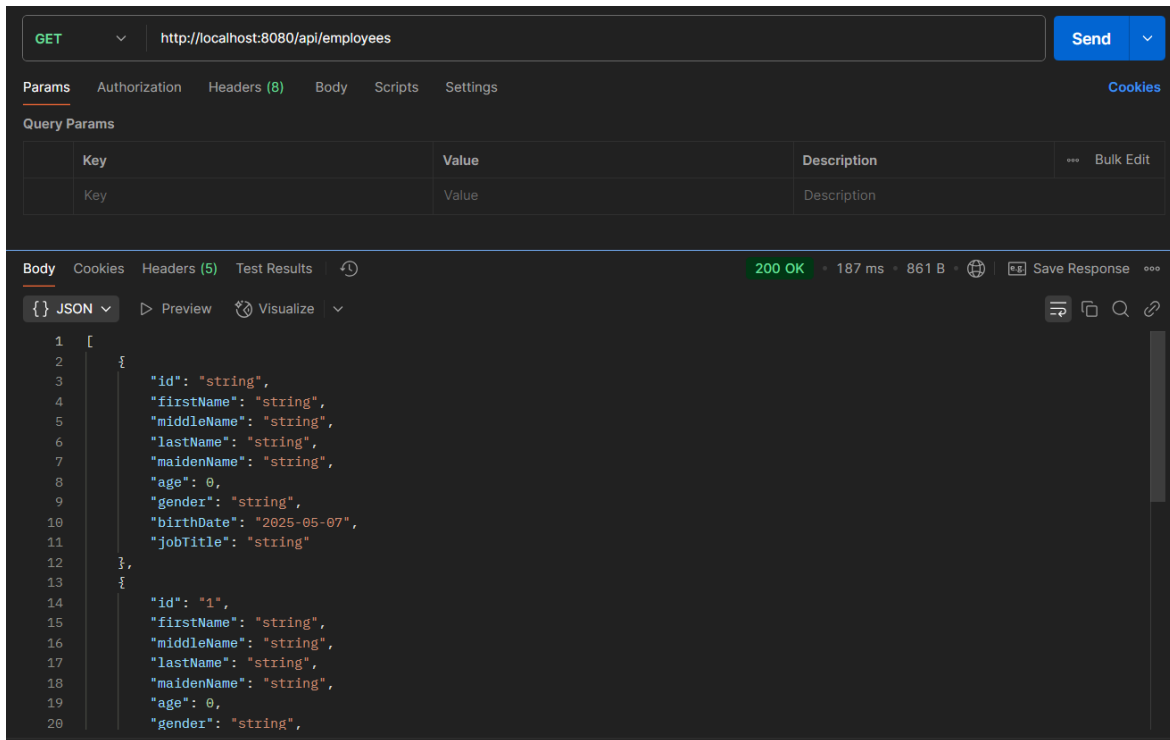
    ____ 
   / ___ \_ _/_ __| |_) |
  / /___ \| '_ \__| |_) |
 /_____/___/\___ \_____|
                          |
:: Spring Boot ::         (v3.3.11)

2025-05-08 11:05:47 - Starting EmployeeApplication v0.0.1-SNAPSHOT using Java 17.0.10 with PID 10876 (C:\Users\eduda\One
Drive\Documents\Proyectos\technical_test\target\employee-0.0.1-SNAPSHOT.jar started by eduda in C:\Users\eduda\OneDrive
\Documents\Proyectos\technical_test\target)
2025-05-08 11:05:47 - Running with Spring Boot v3.3.11, Spring v6.1.19
2025-05-08 11:05:47 - No active profile set, falling back to 1 default profile: "default"
```

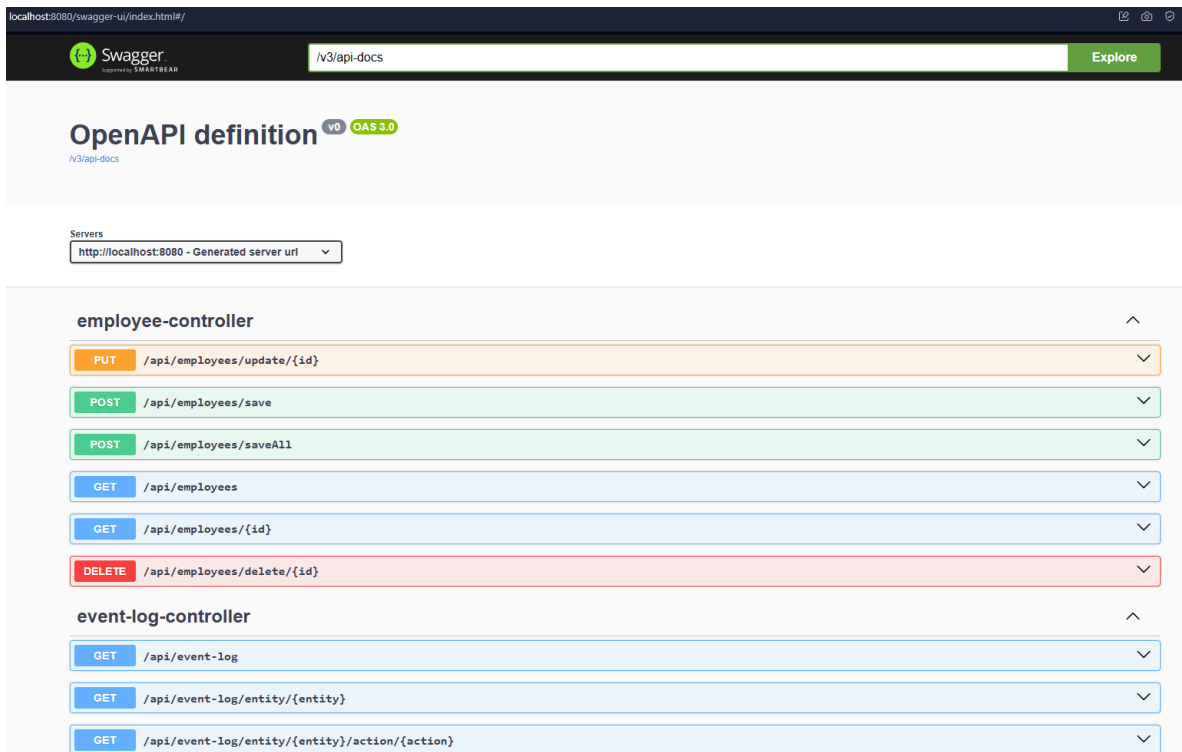
cURL en Postman

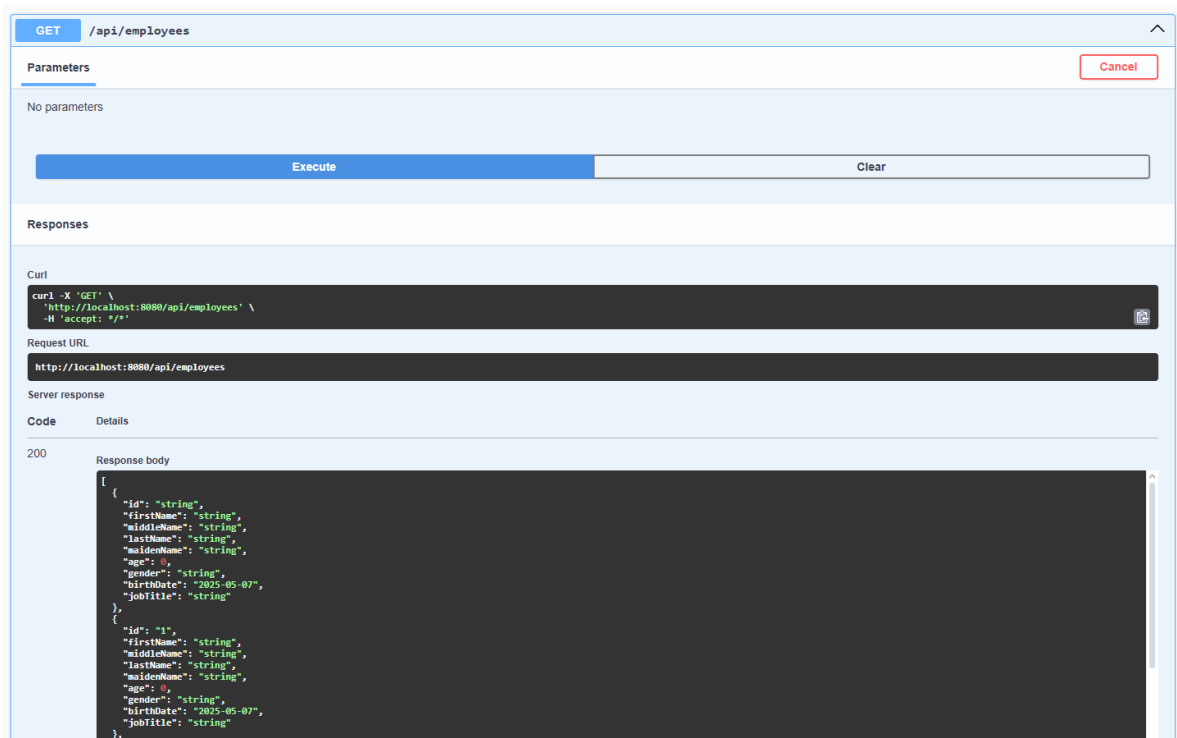


Ejecución desde Postman



Swagger en funcionamiento





Licencia

Este proyecto es de uso libre con fines académicos y profesionales.

Sugerencia: adapta a tus necesidades y da crédito si usas partes significativas.