# 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

1. Clonar el proyecto

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

Nos movemos al directorio que acabamos de crear

1. 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

1. Construir el proyecto

mvn clean package

1. 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

Texto

El contenido generado por IA puede ser incorrecto.

Generación de .jar

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Ejecución del .jar

Texto

El contenido generado por IA puede ser incorrecto.

cURL en Postman

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Ejecución desde Postman

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Swagger en funcionamiento

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

## Licencia

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

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