

21-1-2021

Proyecto de mini código.

Servidor REST de empleado.



David Hernández Andrade.

Descripción de proyecto:

Es una API de servidor REST de empleado, conectada a base de datos.

Se demuestra:

Datos guardados externamente (base de datos).

Uso de java 8.

Como se interactúa con la API a través del navegador y Postman para mostrar las operaciones REST.

La API es capaz de:

- Obtener empleados por un ID.
- Crear nuevos empleados.
- Actualizar a los empleados existentes.
- Eliminar empleados.
- Mostrar todos los empleados.

Datos:

La API maneja datos de empleados donde se componen de los siguientes datos:

- ID: Identificador único de empleado.
- FirstName: Nombre del empleado.
- MiddleInitial: Inicial del segundo nombre del empleado.
- LastName: Apellido del empleado.
- DateBirth: fecha de nacimiento de empleado.
- DateEmp: fecha de inicio del empleado.
- Status: ACTIVO o INACTIVO.

Herramientas Utilizadas:

- Java 8. (Java 1.8).
- Spring Tool Suite. (Spring Boot).
- Postman.
- Google Chrome.
- Servido MySQL.
- MySQL Workbench.

Demostrando Uso:

Uso de Java 8. Instalado en equipo y uso en FrameWork.

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.18363.1316]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Dave HA>java -version
java version "1.8.0_271"
Java(TM) SE Runtime Environment (build 1.8.0_271-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.271-b09, mixed mode)

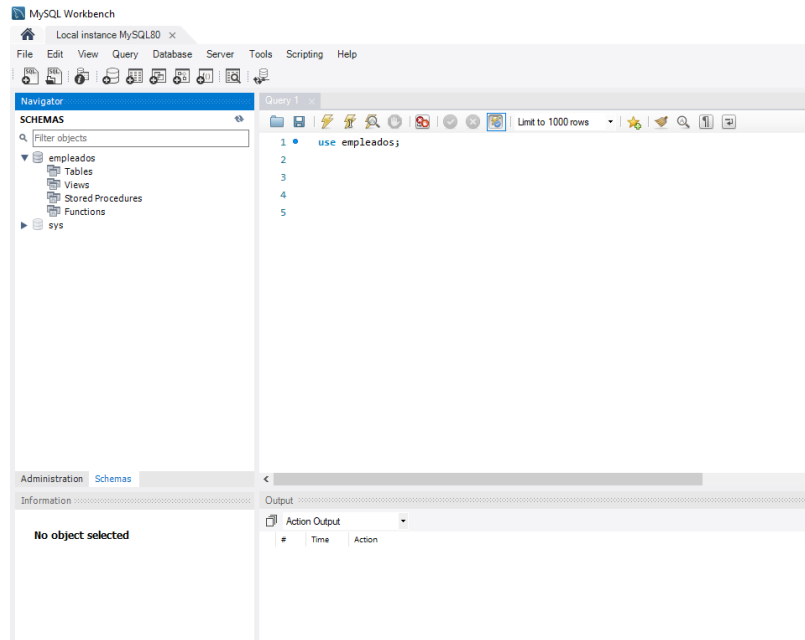
C:\Users\Dave HA>
```

```
4)

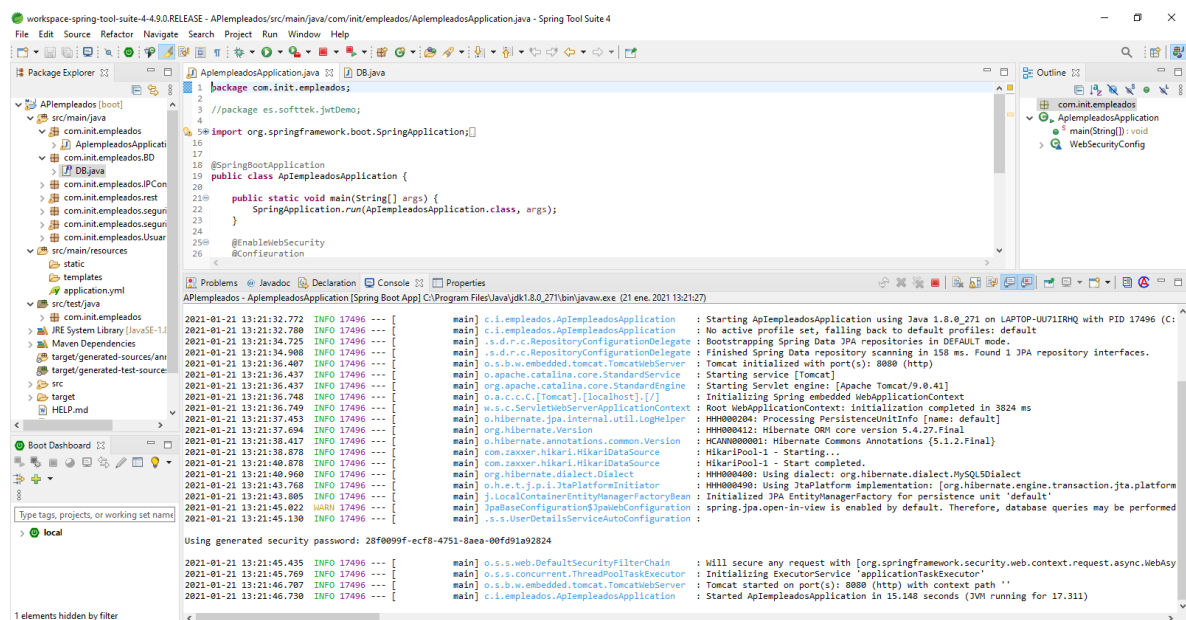
Starting ApiempleadosApplication using Java 1.8.0_271 on LAPTOP-UU71IRHQ with PID 19436 (C:
No active profile set, falling back to default profiles: default
Bootstrapping Spring Data JPA repositories in DEFAULT mode.
Finished Spring Data repository scanning in 182 ms. Found 1 JPA repository interfaces.
Tomcat initialized with port(s): 8080 (http)
Starting service [Tomcat]
Starting Servlet engine: [Apache Tomcat/9.0.41]
Initializing Spring embedded WebApplicationContext
Root WebApplicationContext: initialization completed in 4599 ms
HHH000204: Processing PersistenceUnitInfo [name: default]
HHH000412: Hibernate ORM core version 5.4.27.Final
HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
HikariPool-1 - Starting...
HikariPool-1 - Start completed.
HHH000400: Using dialect: org.hibernate.dialect.MySQL5Dialect
HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform
```

Ejecutando API:

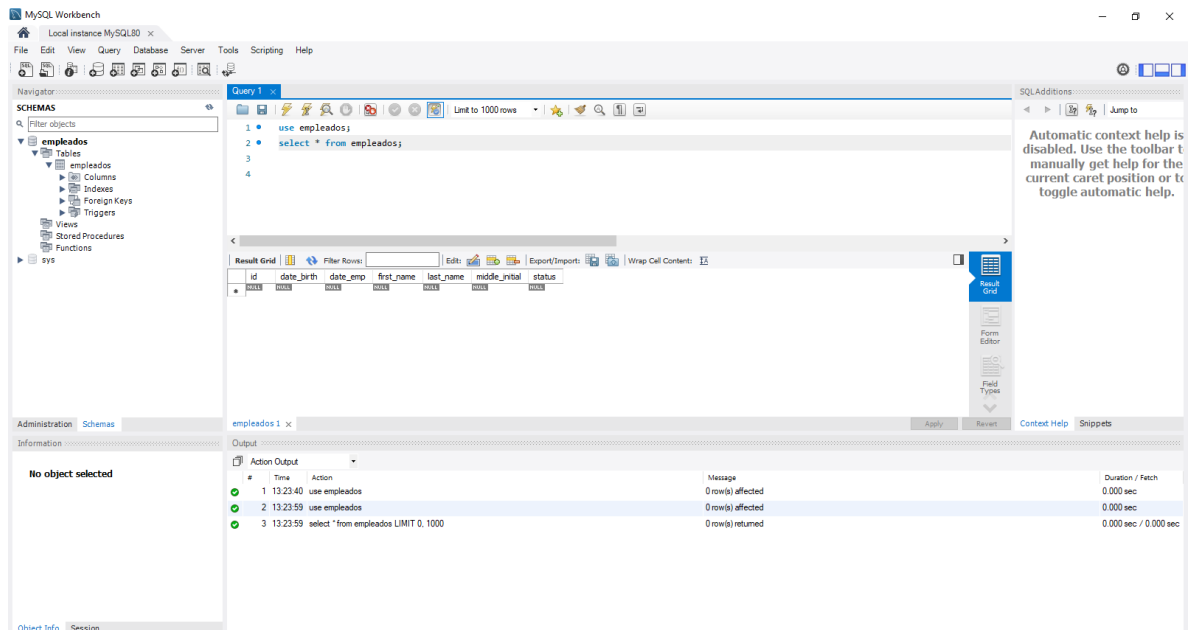
Al ejecutar la API, creará una tabla en una base de datos, hecha previamente, que se llamará “empleados”, una vez creada la tabla. Podremos ingresar datos de empleados, ya sea desde la base de datos o del método POST.



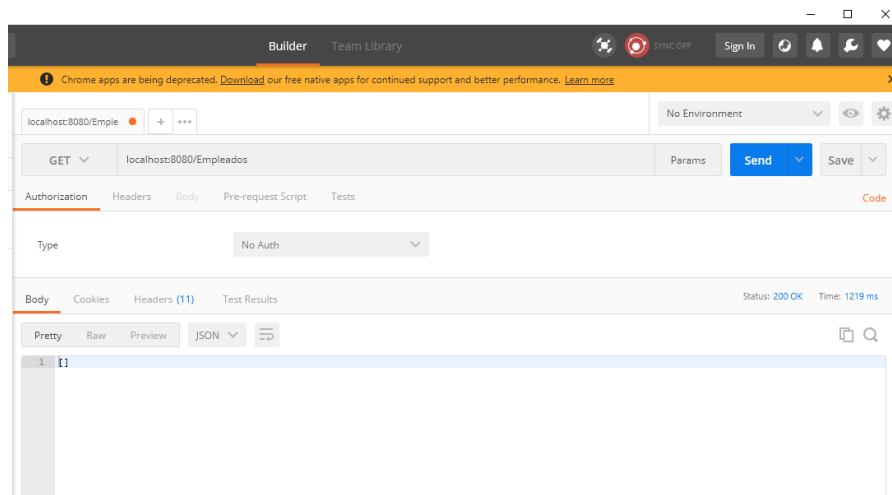
Ejecutamos la API para levantar el servidor:

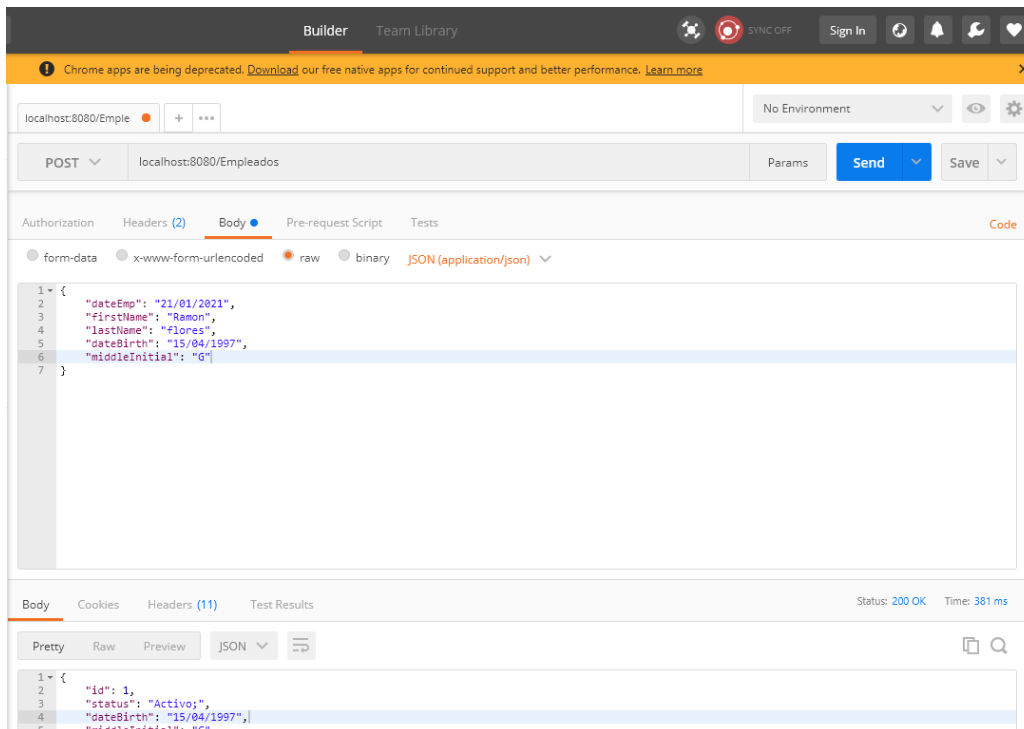
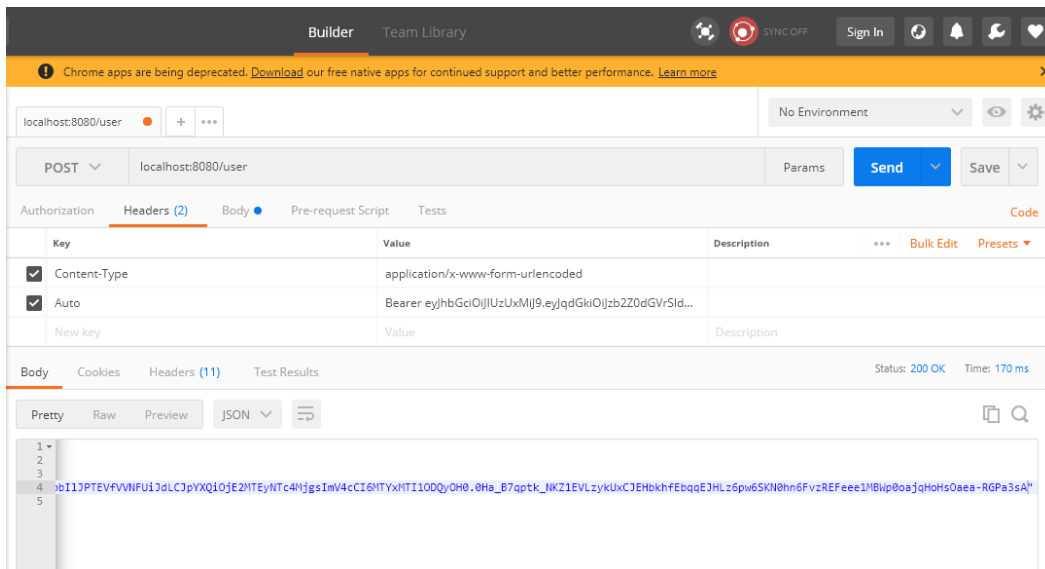


Después se revisa la base de datos y veremos que ha creado la tabla, aunque por el momento no tendrá datos de empleados:

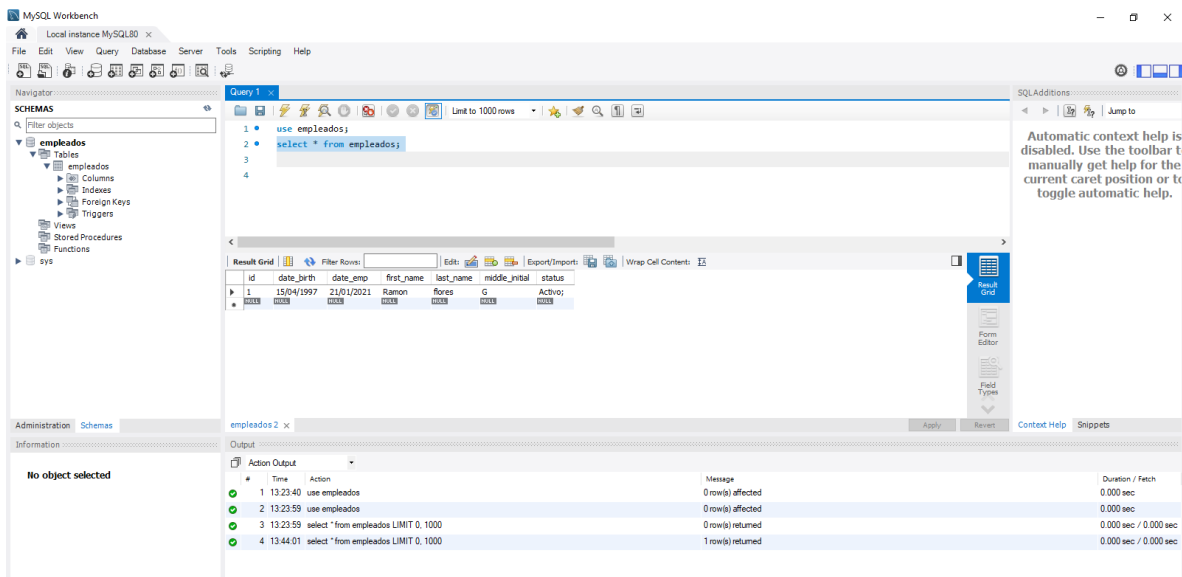
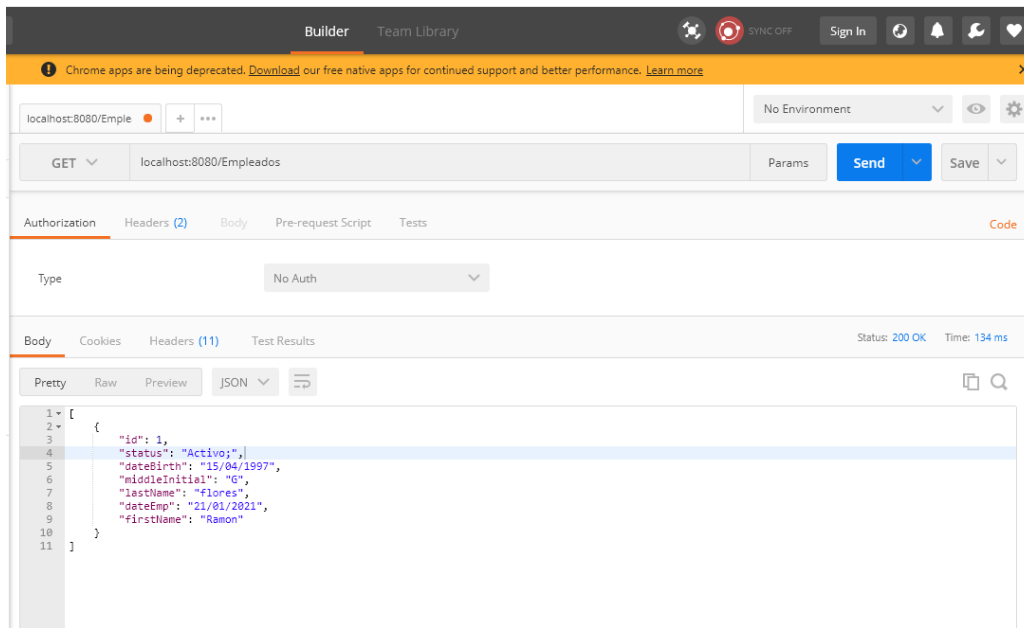


Abrimos Postman y accedemos a la URL: Localhots:8080/Empleados, en el método GET, para observar que no hay datos. Seguido de esto cambiamos al método POST, para agregar empleados, pero nos arroja un error 403, que significa que no estamos autorizados. Esto ya que para realizar los métodos PUT, POST y DELETE en los datos de los empleados, necesitamos autorización por token.

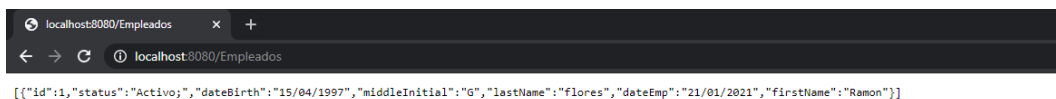




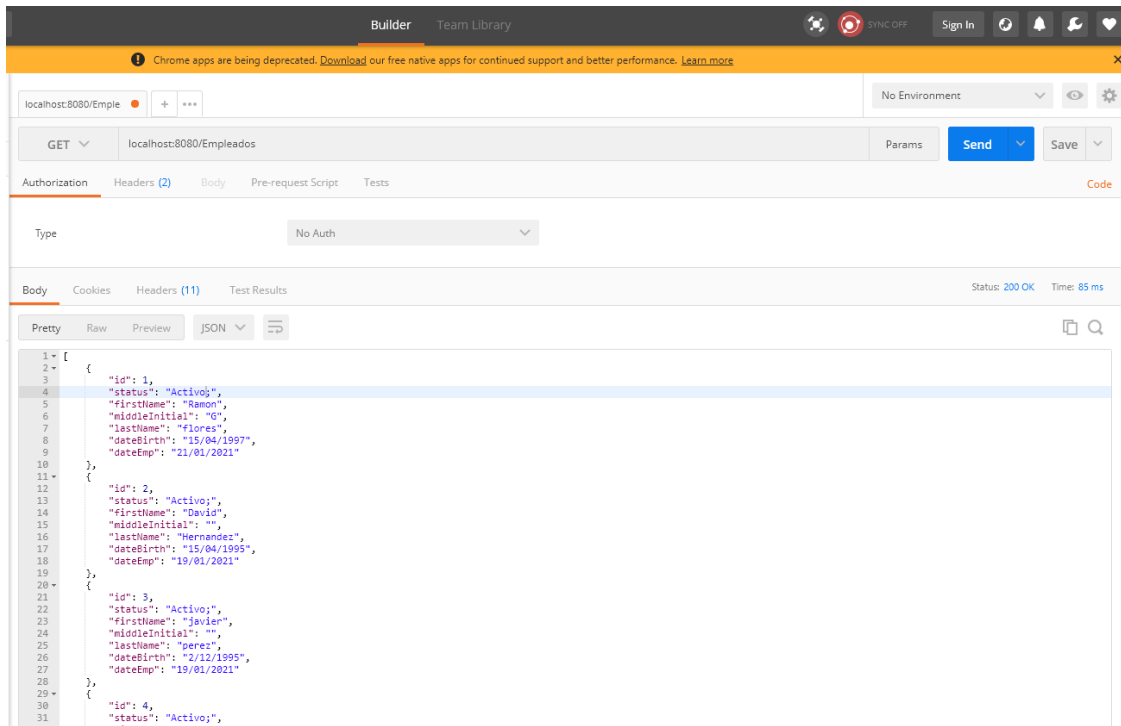
Ahora a través del método GET en la misma url, podemos observar que ya está el empleado dado de alta. Al igual que queda dado de alta en la base de datos.



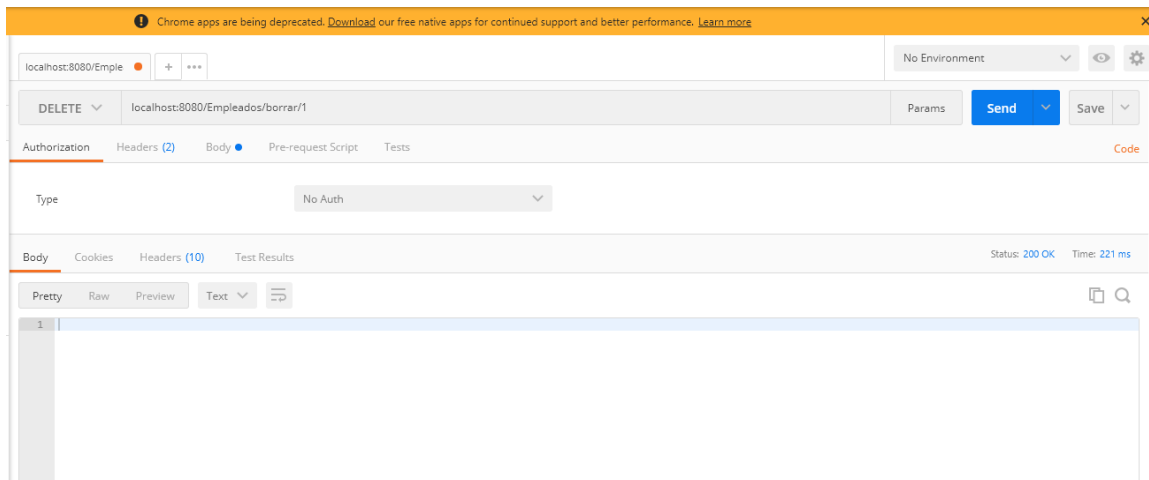
Desde el navegador, donde no tenemos la autorización se puede ver también:



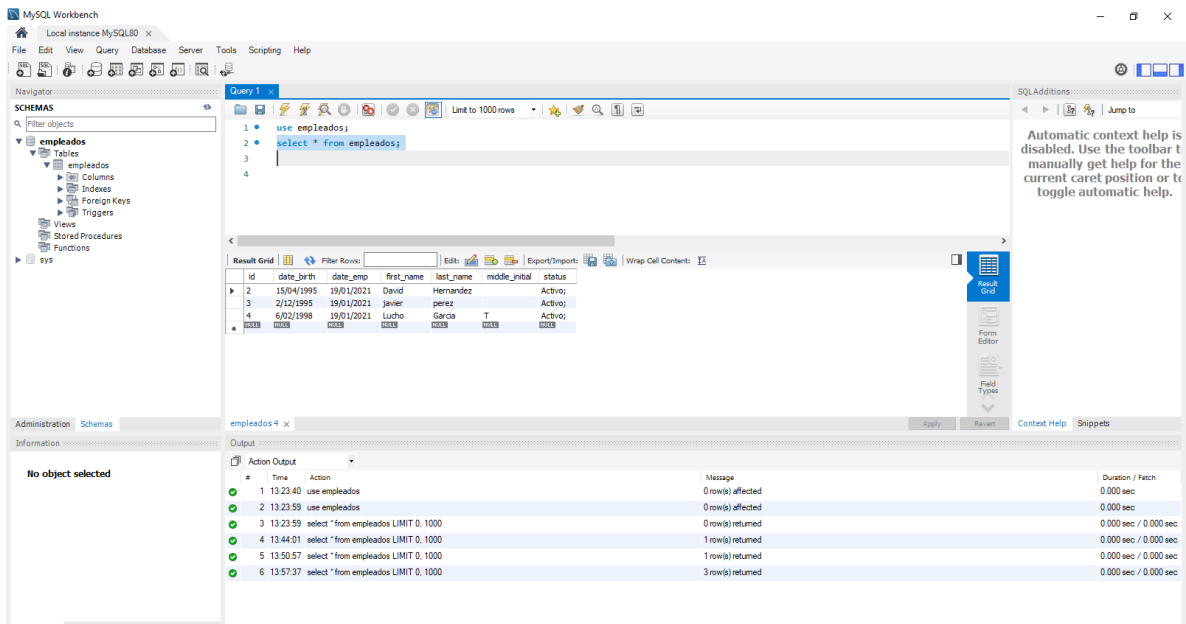
Después de dar de alta varios empleados, vamos a consultar por el método GET para ver la lista de empleados.



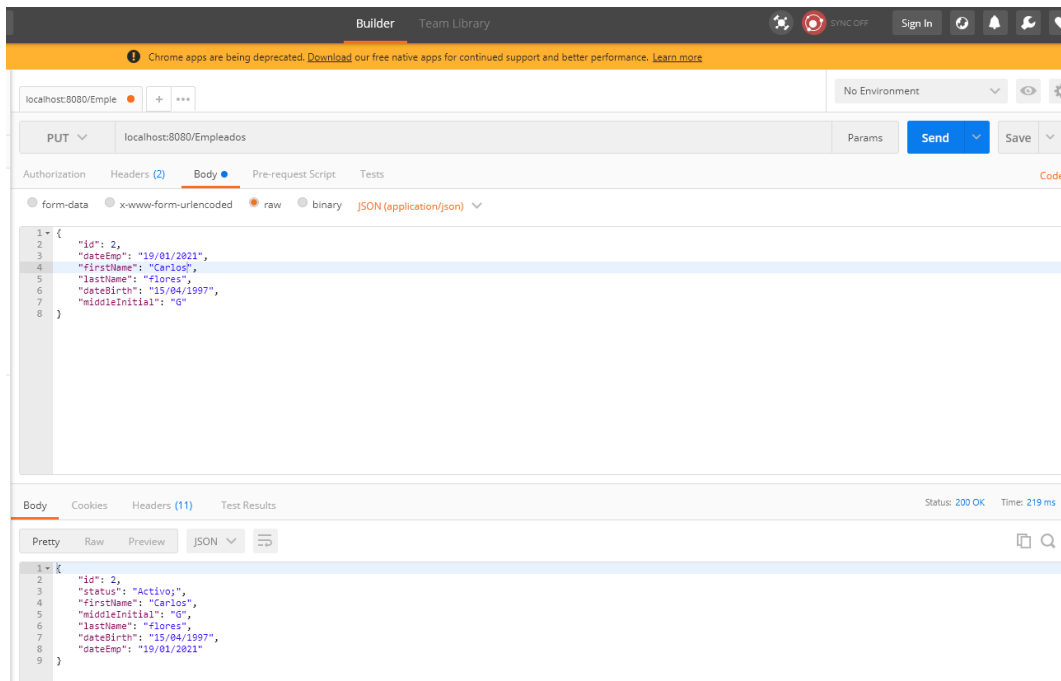
Ahora para pasar a inactivo un empleado que sería borrarlo, tenemos que realizar la autorización para el token, si es que no la hemos hecho, y usar la URL: `localhost/Empleados/borrar/{id del empleado}`, en el método DELETE.

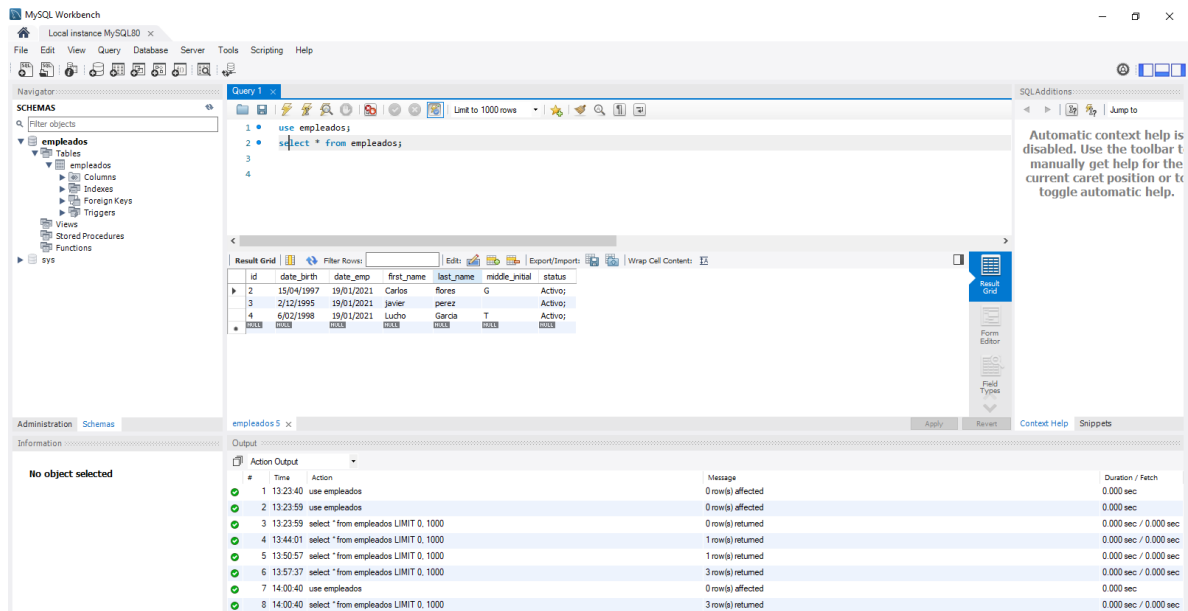


Y comprobamos en la base datos.



Si solo queremos hacer la modificación de datos con el método PUT, en la URL: localhost:8080/Empleados y el token. Colocando los datos que vamos a modificar y el id del empleado en el body.





Y para consultar los datos empleados ya sea todos o uno solo, con el método GET, sin token o con token estarán disponibles. Para uno solo la URL: localhost:8080/Empleados/{id del empleado}

Desde el navegador:



De esta forma es como trabaja el API presentada.