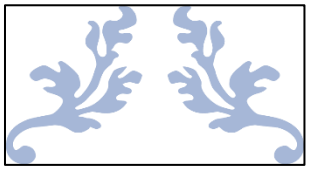

PROYECTO REST CONTROLLER

Academia Xideral, semana 1

27 DE JULIO DE 2025

JUAN CARLOS BERNAL SANDOVAL

Definición de un servicio REST

Durante la primer semana de la academia realizamos un proyecto demo acerca de un servicio “REST CONTROLLER” el cual es comúnmente utilizado en aplicaciones web para comunicar el Frontend (parte vidual) de nuestra aplicación con el backend y de esta forma nos pueda devolver la información necesaria para nuestra aplicación dependiendo del endpoint al cual hagamos la llamada.

Por endpoint decimos de una referencia o complemento del enlace de nuestra aplicación web a través del cual realizamos la comunicación con la lógica de nuestra app, por ejemplo.

https://www.mycoolapp.com/users

En el anterior url nos encontramos con la dirección de nuestra aplicación web en el cual se puede apreciar que termina con “/users”, este sería considerando nuestro endpoint, es decir aquella referencia mediante la cual le indicamos a nuestra aplicación web que queremos acceder a la información de los usuario.

De este forma desencadenamos una serie de acciones del lado del servidor las cuales nuestro backend se encargaría de resolver.

Para realizar esta comunicación es necesario un servicio REST el cual recibe, interpreta, resuelve y retorna información en cada llamada, se apoya de “verbos” http los cuales nos sirven para definir aquella acción que deseamos realizar en nuestra llamada, estos pueden ser POST, GET, UPDATE, PATCH, DELETE.

POST : Este verbo es comúnmente utilizado para realizar inserciones a la base de datos de nuestra aplicación web.

GET : Nos sirve como su nombre lo indica para obtener datos desde la base de datos de nuestra aplicación web.

UPDATE: Realiza la actualización de todos los campos de un registro en nuestra aplicación web.

PATCH : Realiza la actualización solo de los campos seleccionados en nuestra base de datos.

DELETE: Borra un registro en nuestra base de datos.

Claro que estos verbos no actúan por si solos, ya que en principio solo hacen alusión a la acción que se espera que se realice mediante su invocación, al final es la lógica que nosotros definimos en nuestro backend quien determinara el comportamiento de nuestros endpoints.

Herramientas utilizadas.

Springboot: Framework java más ampliamente utilizado para realizar el backend de aplicaciones web, se prefiere por su robustez y facilidad de aprendizaje, además de ser prácticamente el standard de la industria.

Docker: Se usa para crear contenedores con las herramientas necesarias para correr aquellas aplicaciones que necesitamos en nuestros desarrollos de forma no invasiva en nuestras maquinas, es decir si yo quiero instalar un motor de base de datos PostgreSQL en mi computadora bastara con descargar la imagen del contenedor de PostgreSQL y con la ayuda de Docker podre virtualizar los requerimientos necesarios para su instancia, quedando estos solo dentro del contenedor aislados de mi máquina, sin embargo pudiendo trabajar desde un workbench como dbeaver, tal cual como si PostgreSQL estuviera instalado nativamente en mi máquina.

MySQL: Motor de base de datos ampliamente utilizado, nos ayuda a persistir la información de nuestra aplicación web.

MySQL Workbench: Interfaz grafica para trabajar con MySQL de forma mas intuitiva.

Arquitectura.

En la parte del código de nuestro proyecto utilizamos una distribución de clases e interfaces que nos ayudan a mantener un orden concreto y nuestro código desacoplado, destacan 4 clases principales los cuales son, **controller, entity, repository, service.**

Controller: En el definimos los endpoints a los cuales podremos llamar desde nuestra aplicación web y que nos ayudaran a realizar nuestras funciones, en el hacemos uso de diferentes anotaciones como @RestController, @RequestMapping, @GetMapping, @PostMapping, @GetMapping, @PutMapping, @DeleteMapping.

Estas son de ayuda para identificar el tipo de vean con el que estamos trabajando y aquellas que hacen referencia a un verbo HTTP, nos sirven para definir la ruta de nuestros endpoints y si lleva variable de ruta.

Entity: En el paquete entity se encuentran las clases con las que identificamos las tablas de nuestra base de datos, los atributos que contienen estas clases normalmente son iguales a cada columna de cada tabla en nuestra base de datos.

Repository: Realiza las operaciones necesarias para interactuar con la base de datos.

Service: Es nuestro intermediario entre nuestro RestController y el repository, en el se encuentra la lógica de negocio.