ALEJANDRO SAINZ SAINZ

Vamos primero a ver como he hecho el código del controller y del back de springboot.

```java
@RestController
@RequestMapping("/api")

public class MainController {


    public ArrayList<Moroso> lista = new ArrayList<>();

    @GetMapping("/principal")
    public ArrayList<Moroso> obtenerLista() {
        return lista;
    }

    @PostMapping("/add")
    public void addMoroso(@RequestBody Moroso moroso){
        lista.add(moroso);
    }

    @DeleteMapping("/remove/{id}")
    public void removeMoroso(@PathVariable("id") String id){
        for (Moroso moroso : lista) {
            if(moroso.getId().equals(id)){
                lista.remove(moroso);
            }
        }
    }
}
```

@RestController porque el postman se supone que está en otro dominio distinto al del back.

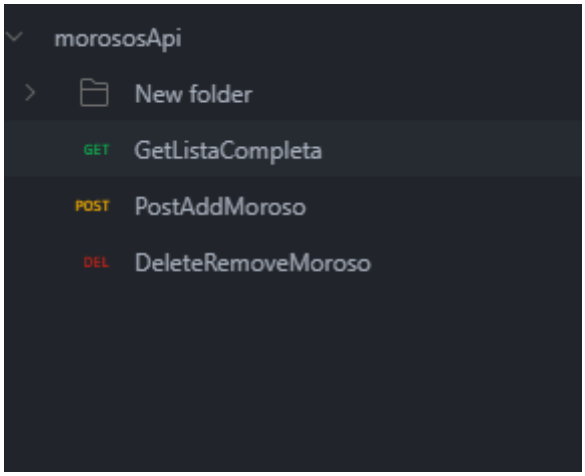
@RequestMapping(/api) para inidicar cual es el endpoint principal de mi servidor.

Como es un ejercicio de prueba, ya se que no debería de ser así, esta vez no he definido un @Service, no es indispensable para realizar las pruebas, así que lo que he hecho es definir primero el arrayList dentro del controller para almacenar los objetos con los que vamos a trabajar.

Con @GetMapping(/principal) indicamos el endpoint la que nos vamos a conectar al hacer peticiones cuando hacemos un get. Nos devuelve el array completo.

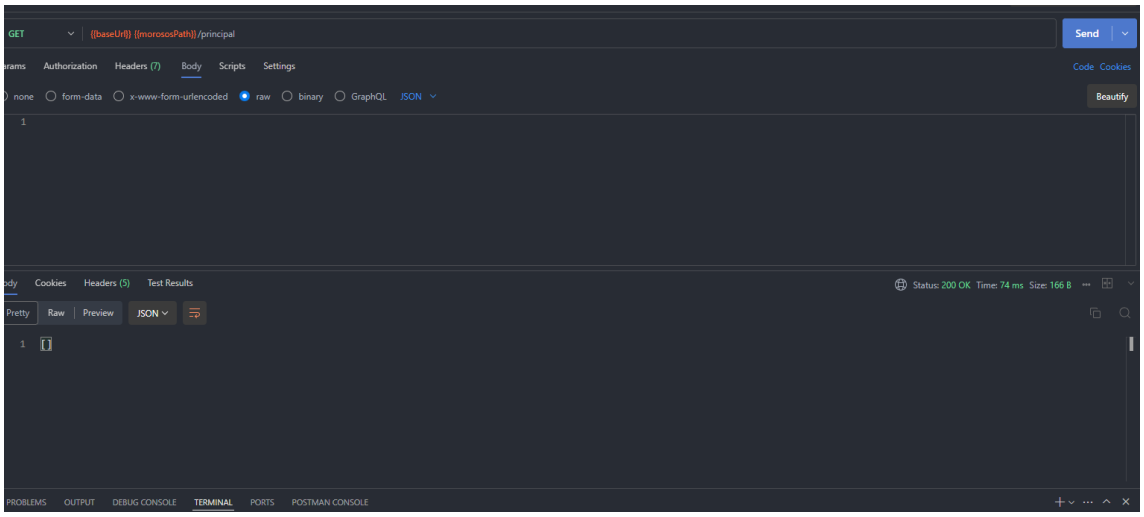
con@PostMapping(/add) marco el endpoint al que postman se tiene que dirigir al hacer peticiones Post.

@DeleteMapping(/remove/{id}) marcamos el endpoint de eliminar. Con id pasamos como parámetro la id del objeto que queremos eliminar.



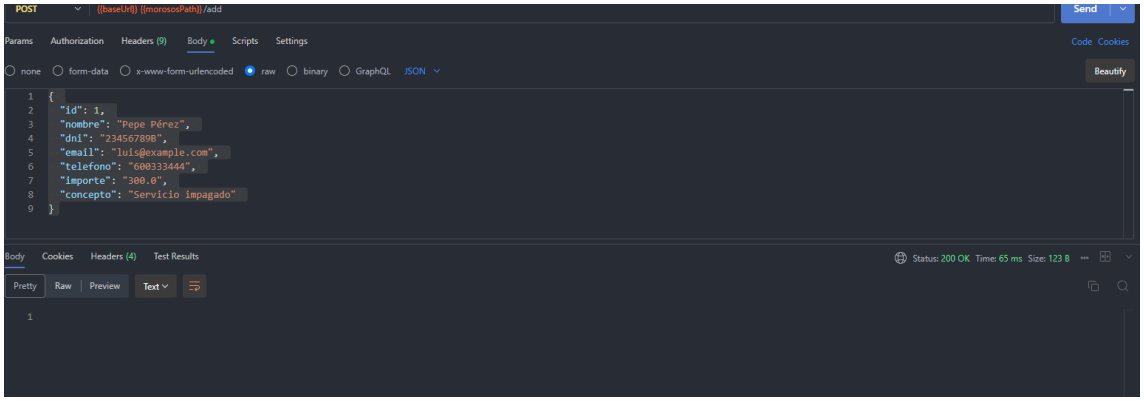
En postman debemos de crear las peticiones con las que vamos a trabajar. Primero ya hemos definidos nuestras variables de enviroment para que pueda ir dirigiéndose a los endpoint necesarios.

Primero vamos a hacer un get para ver que obtenemos.



Este es el resultado. Status 200 ok pero no devuelve nada porque no hay nada.

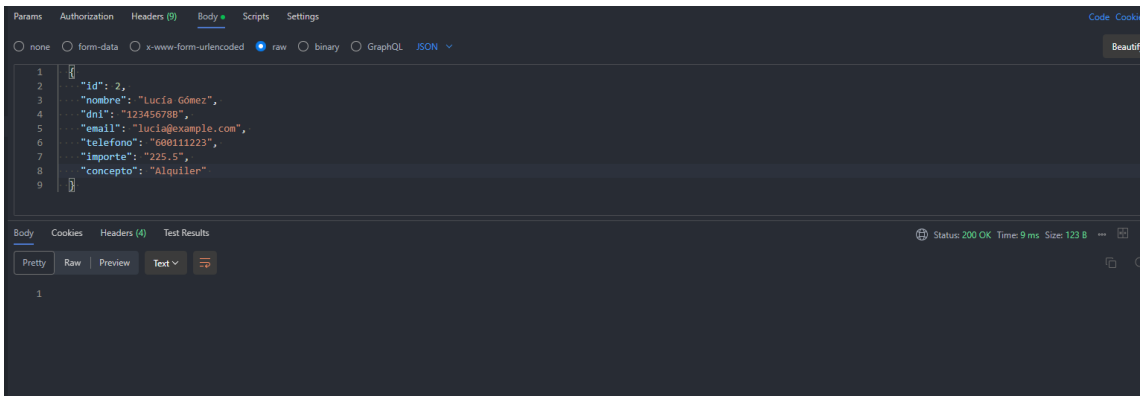
Hay que hacer primero un post.



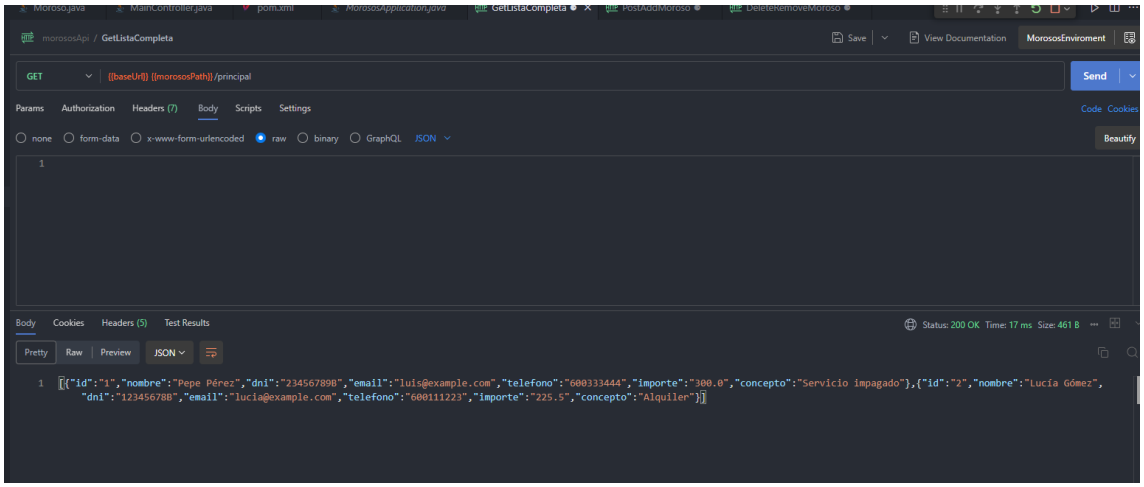
Hacemos primero el post de un objeto JSON y vemos la respuesta, 200 ok.

Ahora volvemos a hacer un GET para comprobar que se ha guardado en nuestro array.

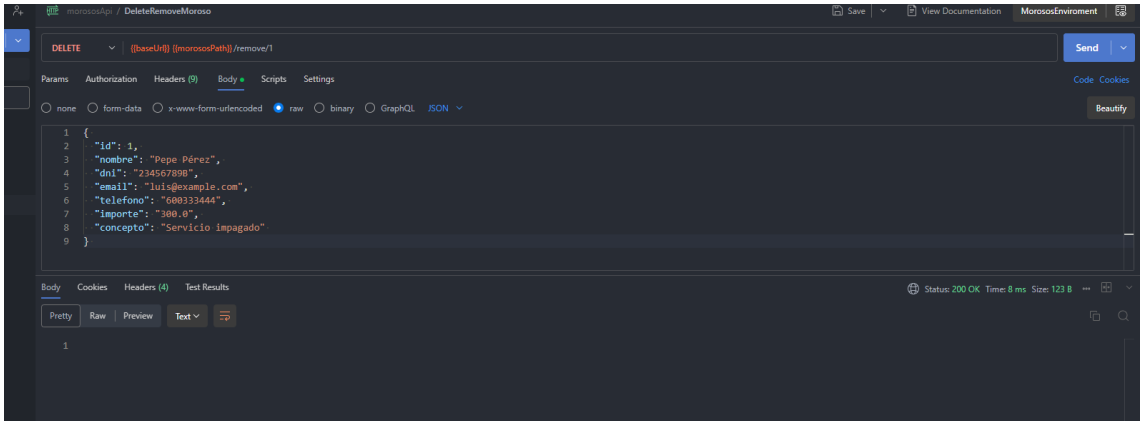
El nuevo GET da un resultado de OK y además nos responde con el mismo objeto que tenemos. Tendremos que hacer más POST para las siguientes partes.



Nuevo Post para añadir un nuevo elemento.



Al hacer el GET tenemos más elementos en la colección. Ahora hay que eliminar 1 para comprobar que funciona el elemento DELETE



Probamos eliminando el primer objeto, que así no hay sospechas de que pueda usar la foto del primer GET. Ahora al hacer un nuevo GET debería de salirnos el segundo elemento de nuestra colección.

Como vemos, con el nuevo GET, obtenemos el segundo elemento de nuestra colección.