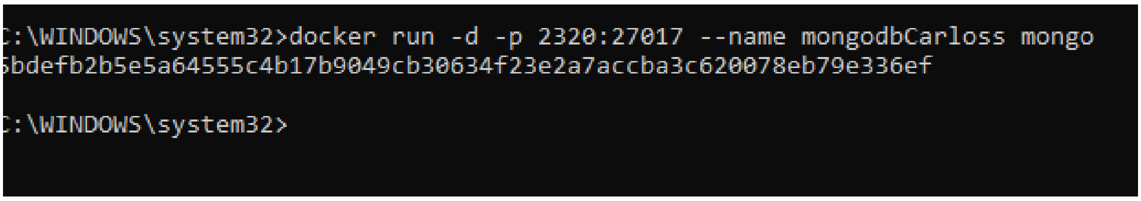
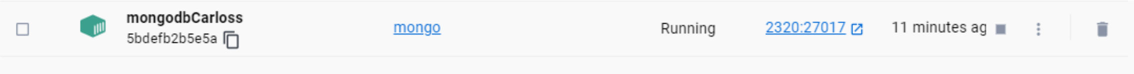
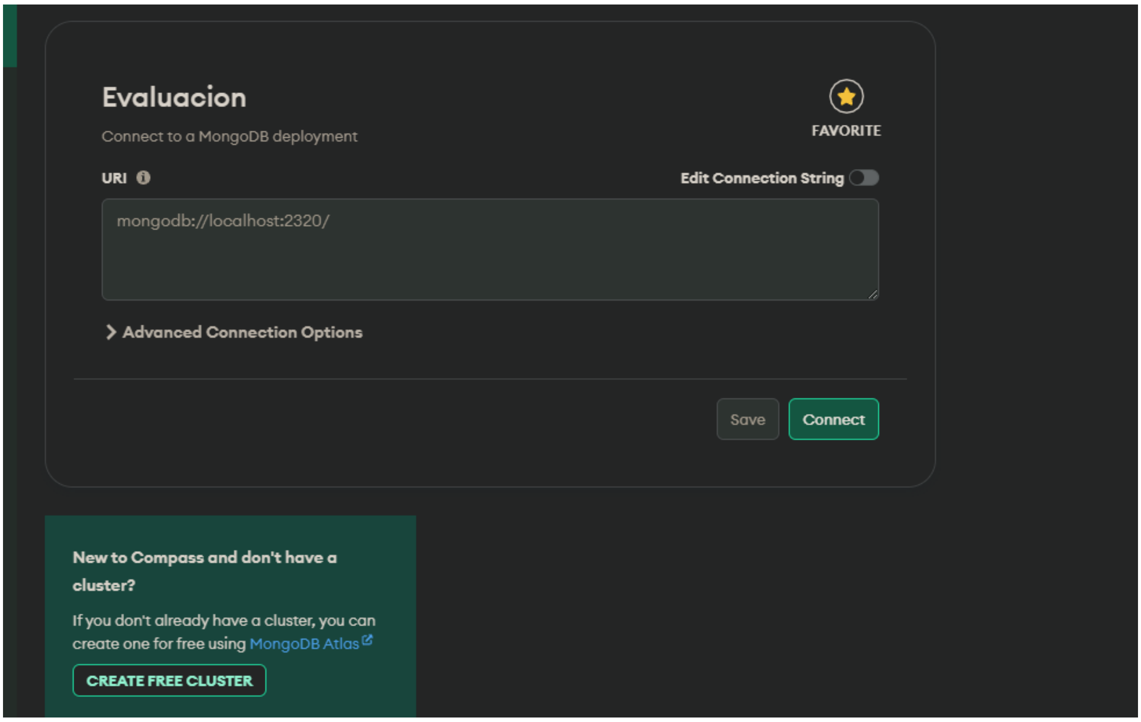


```
C:\WINDOWS\system32>docker run -d -p 2320:27017 --name mongodbCarloss mongo
5bdefb2b5e5a64555c4b17b9049cb30634f23e2a7accba3c620078eb79e336ef

C:\WINDOWS\system32>
```



□ mongodbCarloss 5bdefb2b5e5a mongo Running 2320:27017 11 minutes ag



Evaluacion

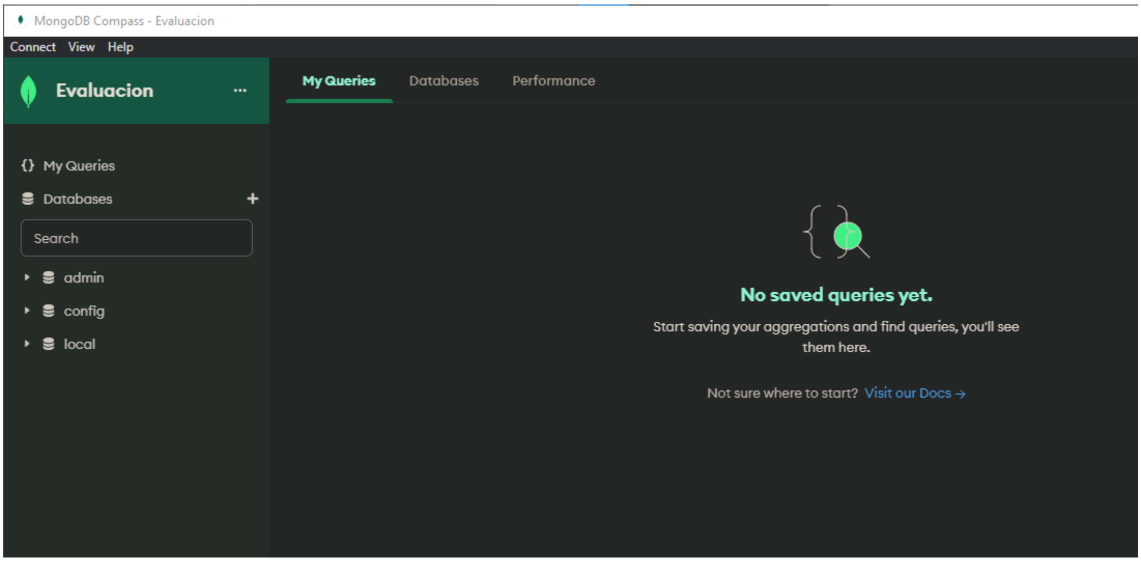
Connect to a MongoDB deployment

URI

mongodb://localhost:2320/

New to Compass and don't have a cluster?

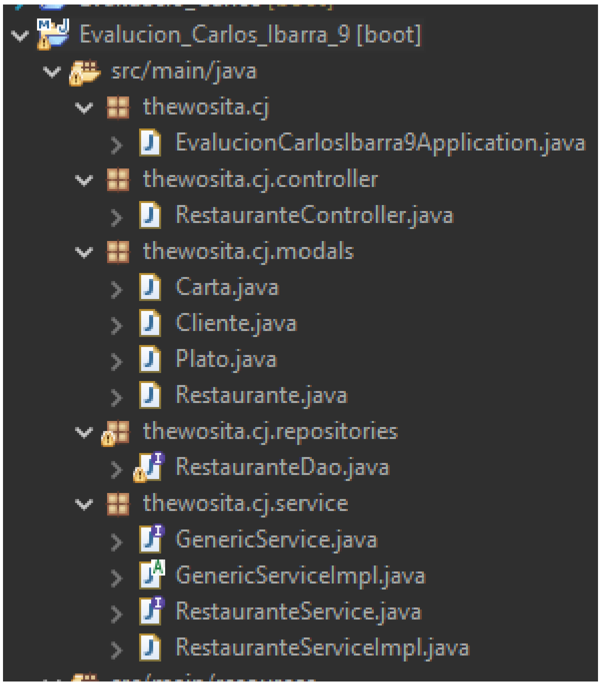
If you don't already have a cluster, you can create one for free using MongoDB Atlas



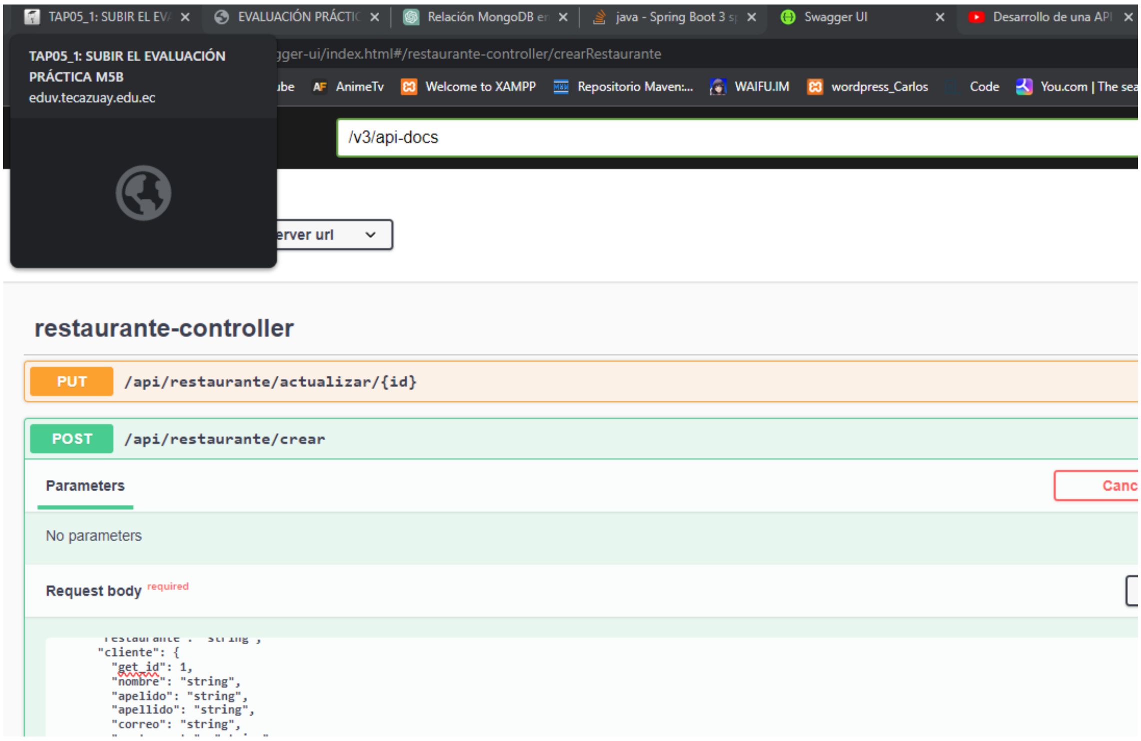
No saved queries yet.

Start saving your aggregations and find queries, you'll see them here.

Not sure where to start? Visit our Docs →



```
Evalucion_Carlos_Ibarra_9 [boot]
  src/main/java
    thewosita.cj
      EvalucionCarlosIbarra9Application.java
    thewosita.cj.controller
      RestauranteController.java
    thewosita.cj.modals
      Carta.java
      Cliente.java
      Plato.java
      Restaurante.java
    thewosita.cj.repositories
      RestauranteDao.java
    thewosita.cj.service
      GenericService.java
      GenericServiceImpl.java
      RestauranteService.java
      RestauranteServiceImpl.java
```



gger-ui/index.html#/restaurante-controller/crearRestaurante

/v3/api-docs

restaurante-controller

PUT /api/restaurante/actualizar/{id}

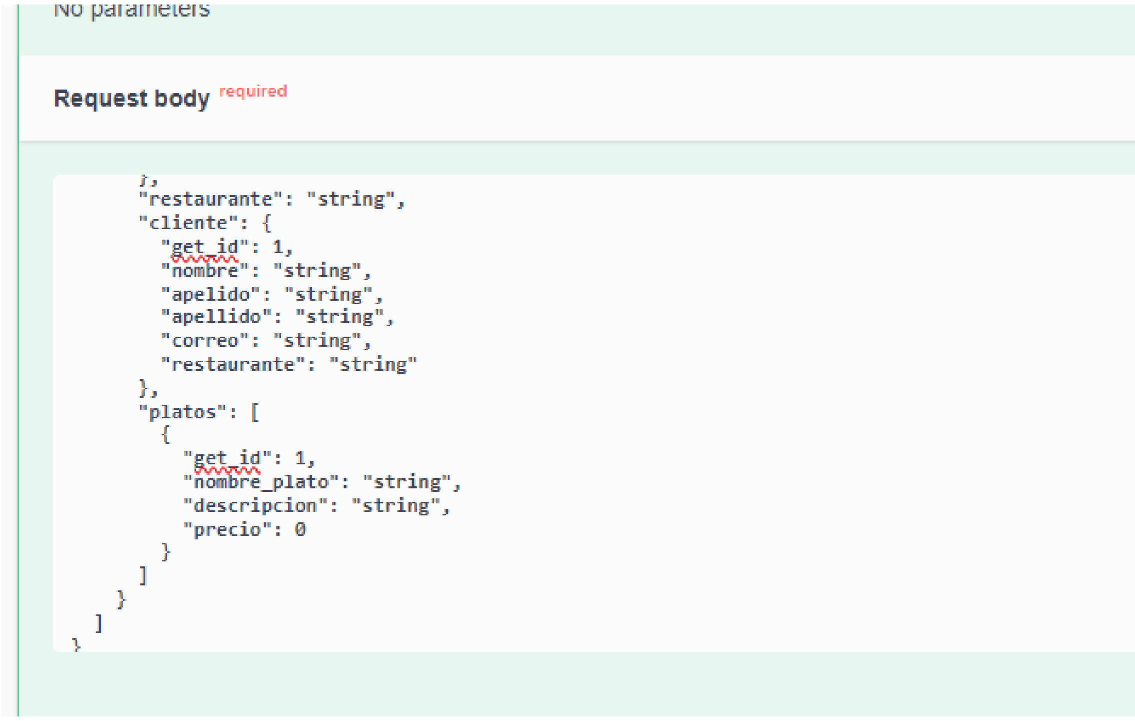
POST /api/restaurante/crear

Parameters

No parameters

Request body required

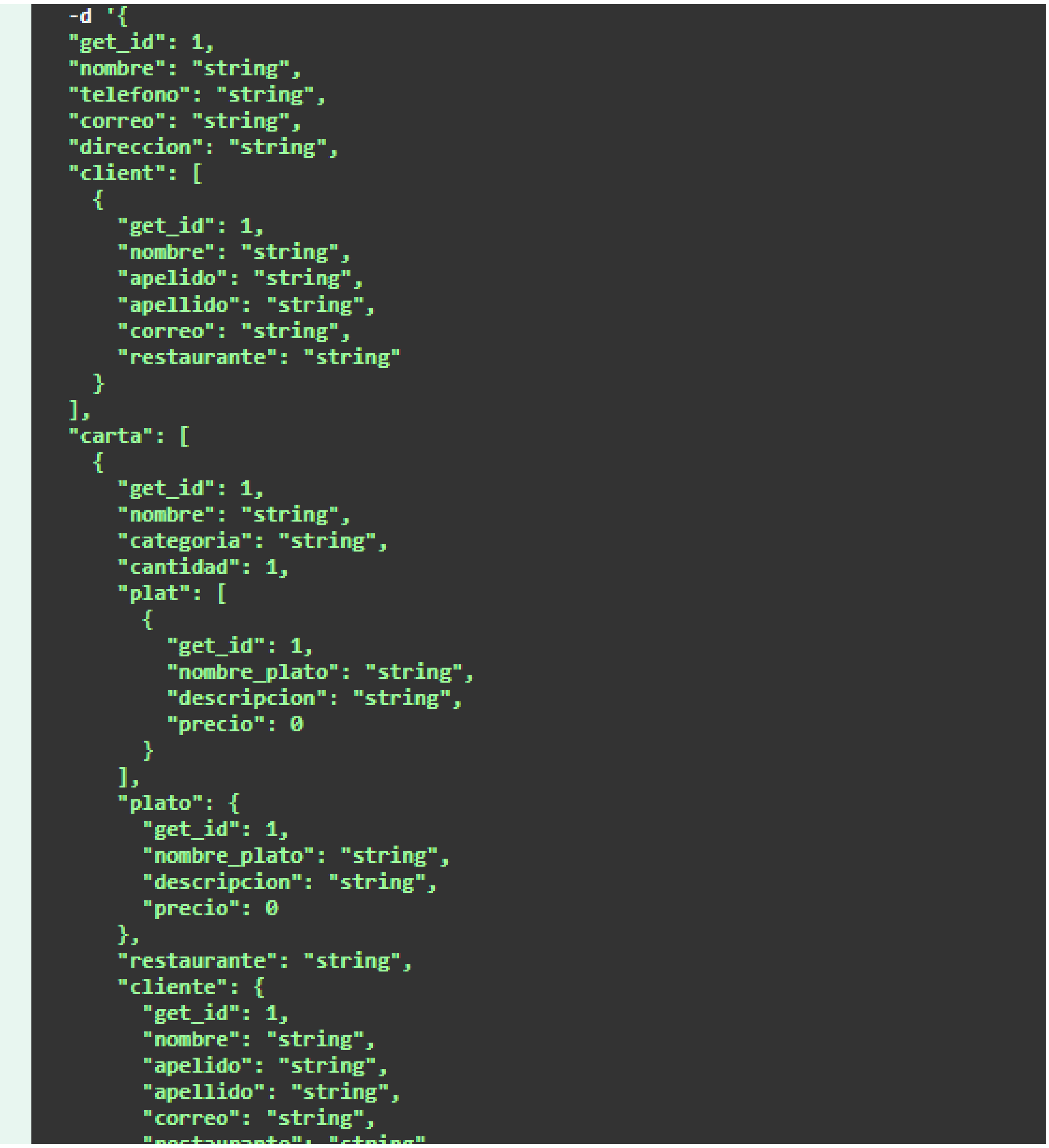
```
        "restaurante": "string",
        "cliente": {
          "get_id": 1,
          "nombre": "string",
          "apelido": "string",
          "apellido": "string",
          "correo": "string",
```



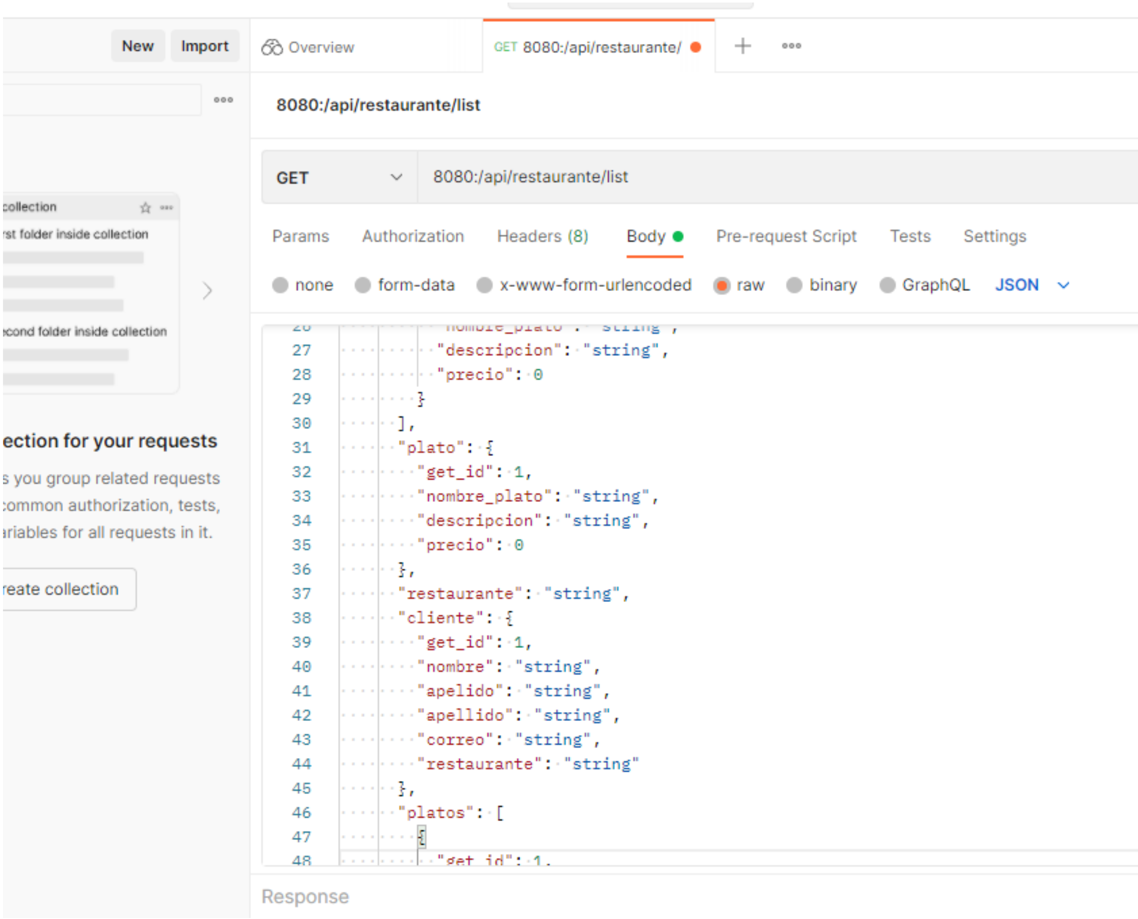
No parameters

Request body

```
      },
      "restaurante": "string",
      "cliente": {
        "get_id": 1,
        "nombre": "string",
        "apelido": "string",
        "apellido": "string",
        "correo": "string",
        "restaurante": "string"
      },
      "platos": [
        {
          "get_id": 1,
          "nombre_plato": "string",
          "descripcion": "string",
          "precio": 0
        }
      ]
    }
  ]
}
```



```
-d '{
"get_id": 1,
"nombre": "string",
"telefono": "string",
"correo": "string",
"direccion": "string",
"client": [
  {
    "get_id": 1,
    "nombre": "string",
    "apelido": "string",
    "apellido": "string",
    "correo": "string",
    "restaurante": "string"
  }
],
"carta": [
  {
    "get_id": 1,
    "nombre": "string",
    "categoria": "string",
    "cantidad": 1,
    "plat": [
      {
        "get_id": 1,
        "nombre_plato": "string",
        "descripcion": "string",
        "precio": 0
      }
    ],
    "plato": {
      "get_id": 1,
      "nombre_plato": "string",
      "descripcion": "string",
      "precio": 0
    },
    "restaurante": "string",
    "cliente": {
      "get_id": 1,
      "nombre": "string",
      "apelido": "string",
      "apellido": "string",
      "correo": "string",
      "restaurante": "string"
```

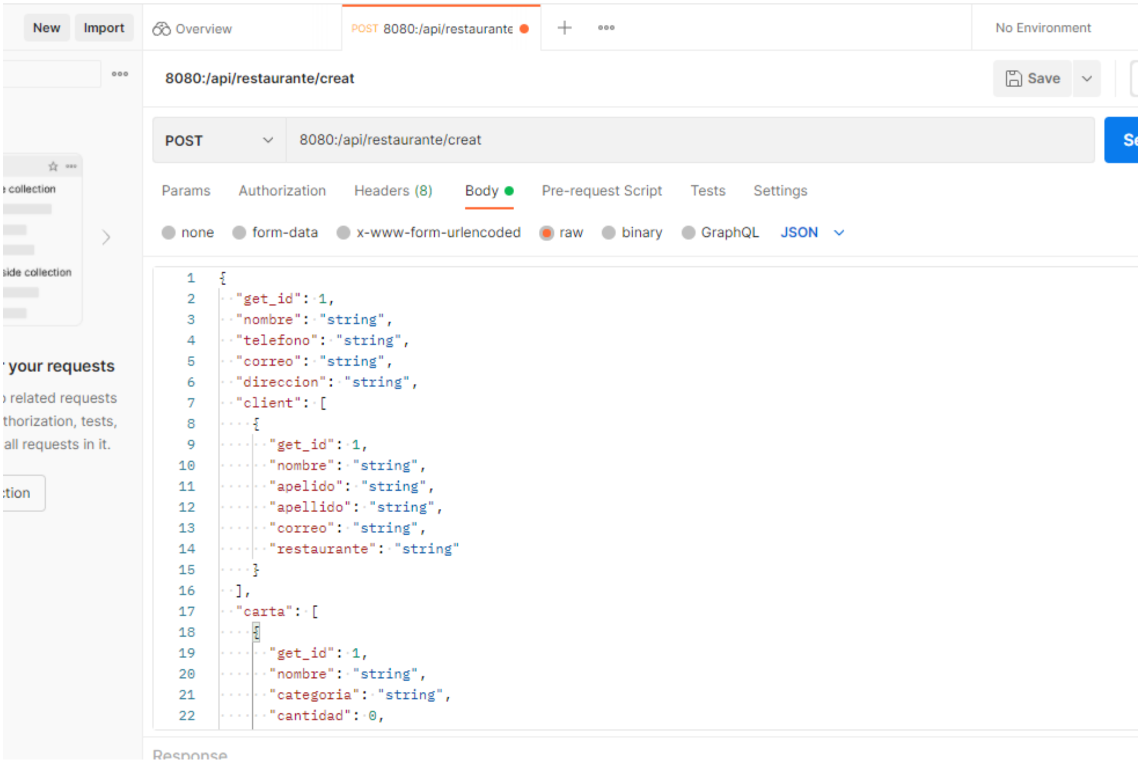

New
Import
Overview
GET 8080:/api/restaurante/
collection
rst folder inside collection
cond folder inside collection
ection for your requests
s you group related requests
common authorization, tests,
ariables for all requests in it.
reate collection
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Response

8080:/api/restaurante/list

GET 8080:/api/restaurante/list

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
          "nombre_plato": "string",
          "descripcion": "string",
          "precio": 0
        }
      ],
      "plato": {
        "get_id": 1,
        "nombre_plato": "string",
        "descripcion": "string",
        "precio": 0
      },
      "restaurante": "string",
      "cliente": {
        "get_id": 1,
        "nombre": "string",
        "apelido": "string",
        "apellido": "string",
        "correo": "string",
        "restaurante": "string"
      },
      "platos": [
        {
          "get_id": 1,
```

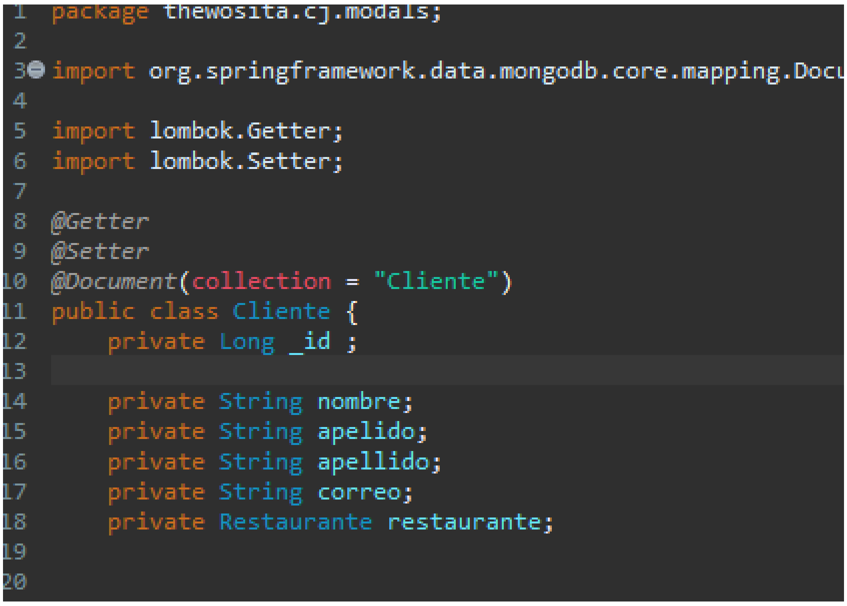


8080:/api/restaurante/creat

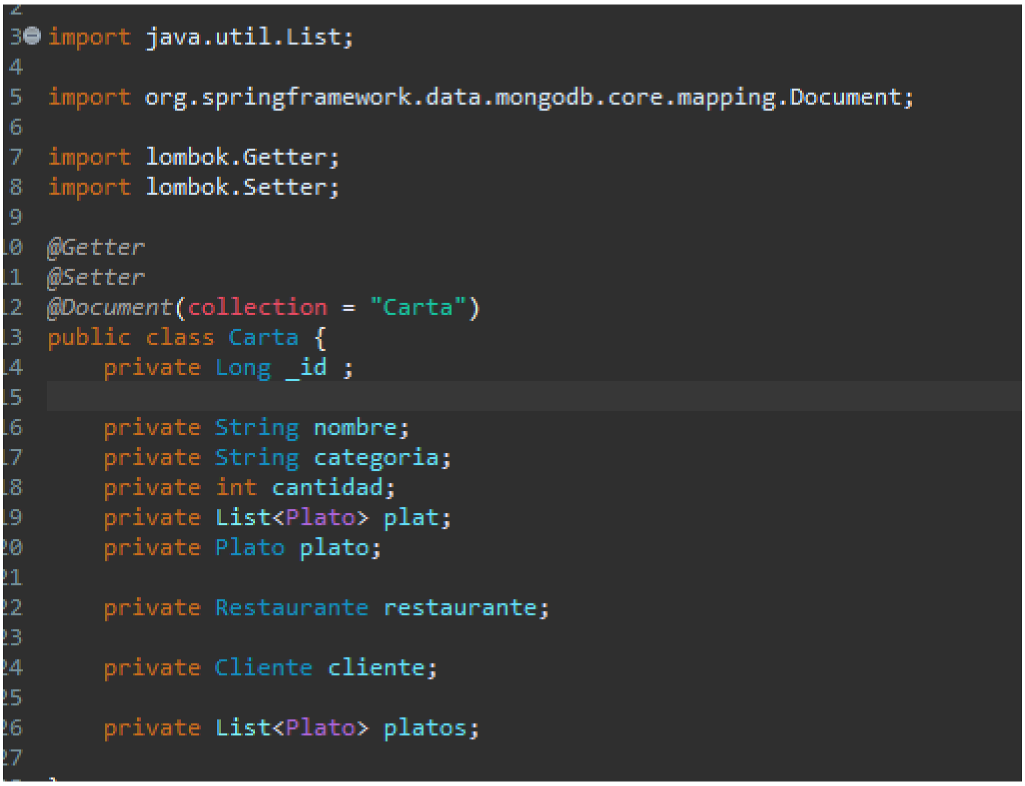
POST 8080:/api/restaurante/creat

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

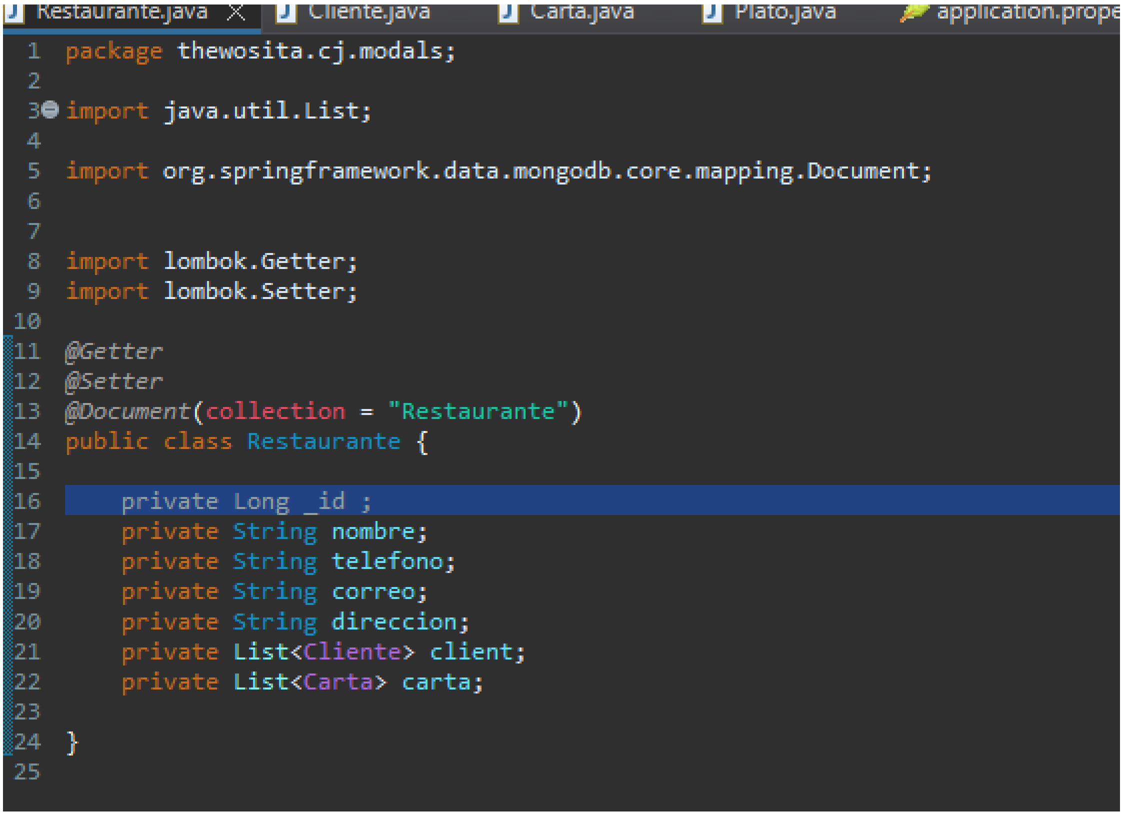
```
{
  "get_id": 1,
  "nombre": "string",
  "telefono": "string",
  "correo": "string",
  "direccion": "string",
  "client": [
    {
      "get_id": 1,
      "nombre": "string",
      "apelido": "string",
      "apellido": "string",
      "correo": "string",
      "restaurante": "string"
    }
  ],
  "carta": [
    {
      "get_id": 1,
      "nombre": "string",
      "categoria": "string",
      "cantidad": 0,
```



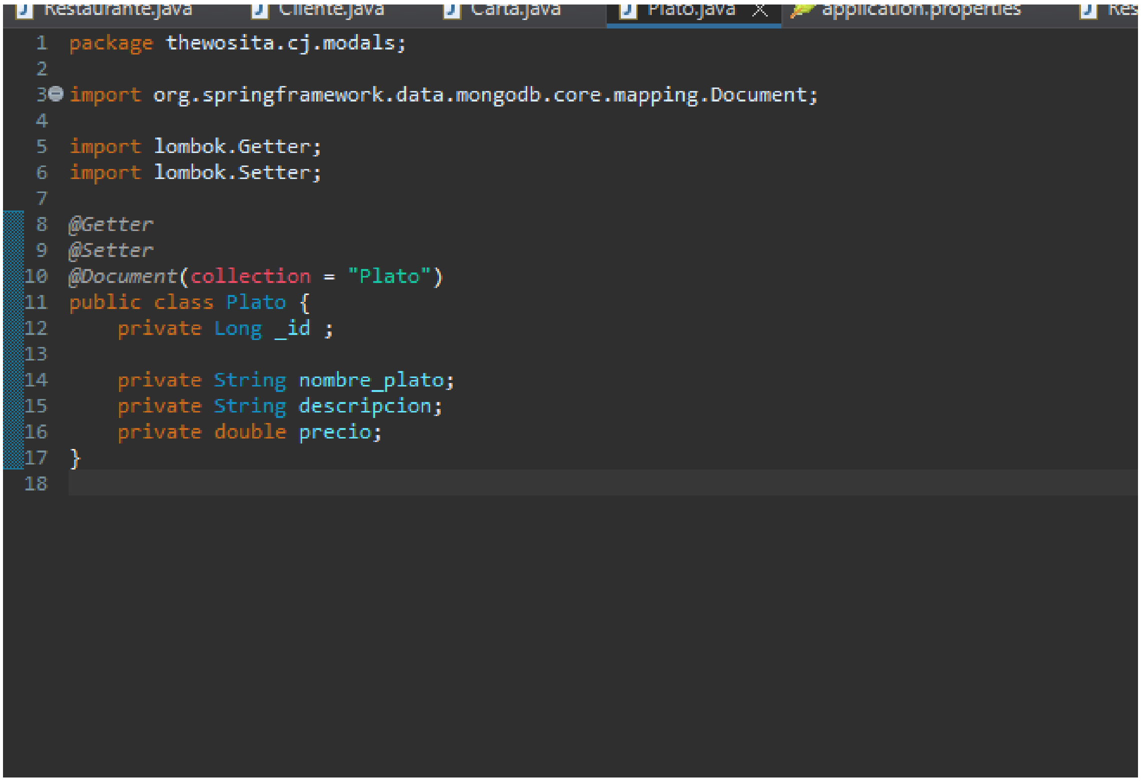
```java
package thewosita.cj.modals;

import org.springframework.data.mongodb.core.mapping.Docu

import lombok.Getter;
import lombok.Setter;

@Getter
@Setter
@Document(collection = "Cliente")
public class Cliente {
    private Long _id ;

    private String nombre;
    private String apelido;
    private String apellido;
    private String correo;
    private Restaurante restaurante;

```



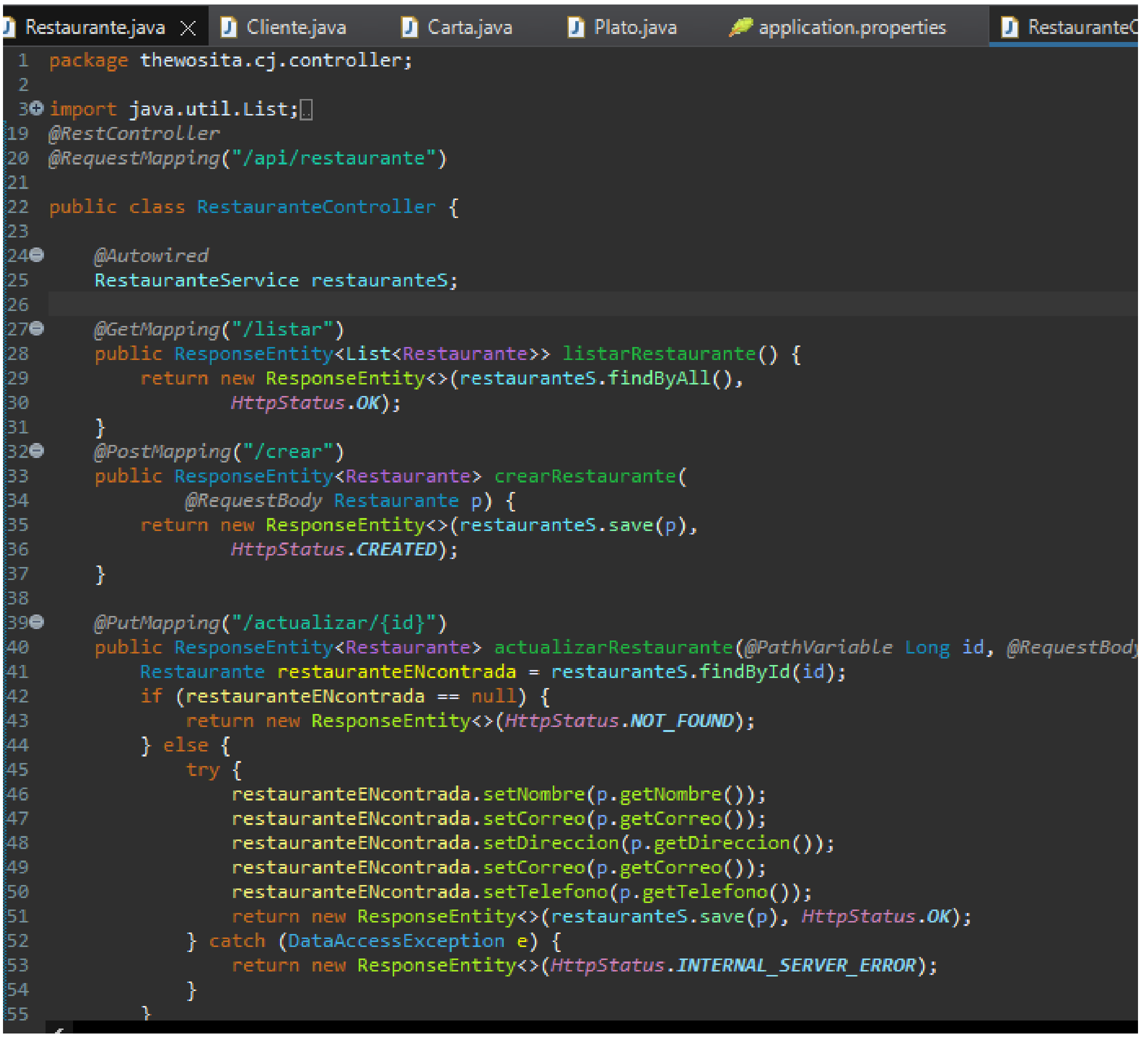
```java
import java.util.List;

import org.springframework.data.mongodb.core.mapping.Document;

import lombok.Getter;
import lombok.Setter;

@Getter
@Setter
@Document(collection = "Carta")
public class Carta {
    private Long _id ;

    private String nombre;
    private String categoria;
    private int cantidad;
    private List<Plato> plat;
    private Plato plato;

    private Restaurante restaurante;

    private Cliente cliente;

    private List<Plato> platos;

}
```



```java
package thewosita.cj.modals;

import java.util.List;

import org.springframework.data.mongodb.core.mapping.Document;


import lombok.Getter;
import lombok.Setter;

@Getter
@Setter
@Document(collection = "Restaurante")
public class Restaurante {

    private Long _id ;
    private String nombre;
    private String telefono;
    private String correo;
    private String direccion;
    private List<Cliente> client;
    private List<Carta> carta;

}

```



```java
package thewosita.cj.modals;

import org.springframework.data.mongodb.core.mapping.Document;

import lombok.Getter;
import lombok.Setter;

@Getter
@Setter
@Document(collection = "Plato")
public class Plato {
    private Long _id ;

    private String nombre_plato;
    private String descripcion;
    private double precio;
}

```



```java
package thewosita.cj.controller;

import java.util.List;
@RestController
@RequestMapping("/api/restaurante")

public class RestauranteController {

    @Autowired
    RestauranteService restauranteS;

    @GetMapping("/listar")
    public ResponseEntity<List<Restaurante>> listarRestaurante() {
        return new ResponseEntity<>(restauranteS.findByAll(),
                HttpStatus.OK);
    }
    @PostMapping("/crear")
    public ResponseEntity<Restaurante> crearRestaurante(
            @RequestBody Restaurante p) {
        return new ResponseEntity<>(restauranteS.save(p),
                HttpStatus.CREATED);
    }

    @PutMapping("/actualizar/{id}")
    public ResponseEntity<Restaurante> actualizarRestaurante(@PathVariable Long id, @RequestBody
        Restaurante restauranteENcontrada = restauranteS.findById(id);
        if (restauranteENcontrada == null) {
            return new ResponseEntity<>(HttpStatus.NOT_FOUND);
        } else {
            try {
                restauranteENcontrada.setNombre(p.getNombre());
                restauranteENcontrada.setCorreo(p.getCorreo());
                restauranteENcontrada.setDireccion(p.getDireccion());
                restauranteENcontrada.setCorreo(p.getCorreo());
                restauranteENcontrada.setTelefono(p.getTelefono());
                return new ResponseEntity<>(restauranteS.save(p), HttpStatus.OK);
            } catch (DataAccessException e) {
                return new ResponseEntity<>(HttpStatus.INTERNAL_SERVER_ERROR);
            }
        }
```