LABORATORIO DE INTEGRACION EMPRESARIAL

1. **Compilación y Ejecución de servicio pryservice123 construido en Python.**
   1. Accedemos a la ruta.

**/laboratorio\_integracion\_empresarial**

Procedemos a lanzar el comando para habilitar el entono virtual de Python, para lo cual abrimos una terminal o un CMD en Windows.

**source pryservice123/bin/activate**

* 1. Luego navegamos hacia el proyecto de Python.

**/laboratorio\_integracion\_empresarial/pryservice123**

* 1. Una vez dentro de este habilitamos procedemos a instalar las dependencias con el comando.

**pip install -r pryservice123/requirements.txt**

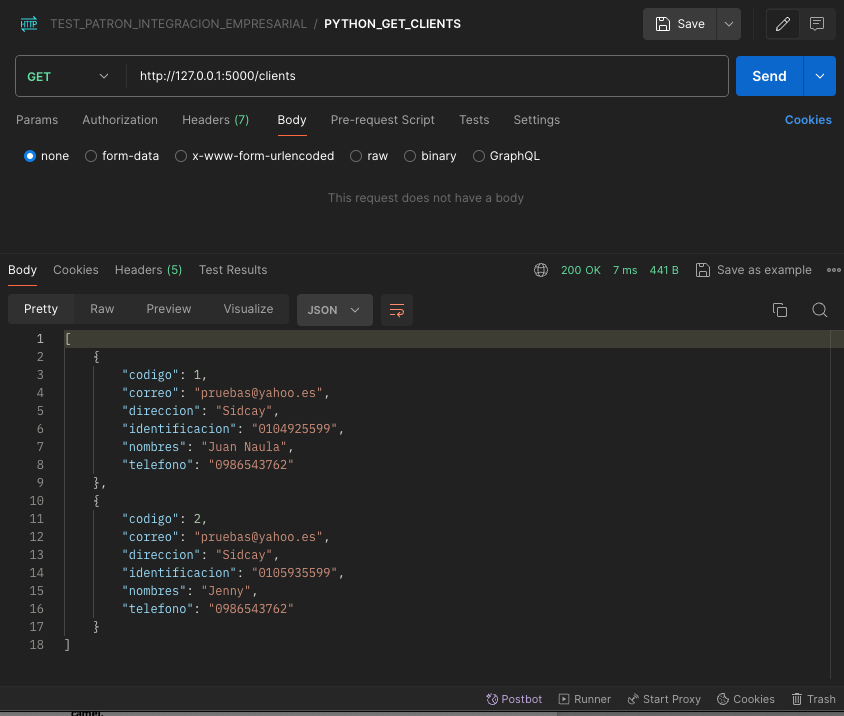
* 1. Una vez instalado ya podemos ejecutar nuestra aplicación para lo cual ejecutamos el comando.

**python main.py**

* 1. Una vez ejecutada la aplicación ya podremos probar nuestro servicio para lo cual lo aremos con el método GET.

[**http://127.0.0.1:5000/clients**](http://127.0.0.1:5000/clients)

Este servicio inicialmente nos mostrara un listado de clientes inicial.



1. **Compilación y Ejecución de Servicio netmicro construido en NetCore 7.**
   1. Requisitos previos tener instalado netcore 7.0 tomado del link.

**https://learn.microsoft.com/es-es/dotnet/core/install/**

* 1. Limpieza, Construcción y ejecución del proyecto.

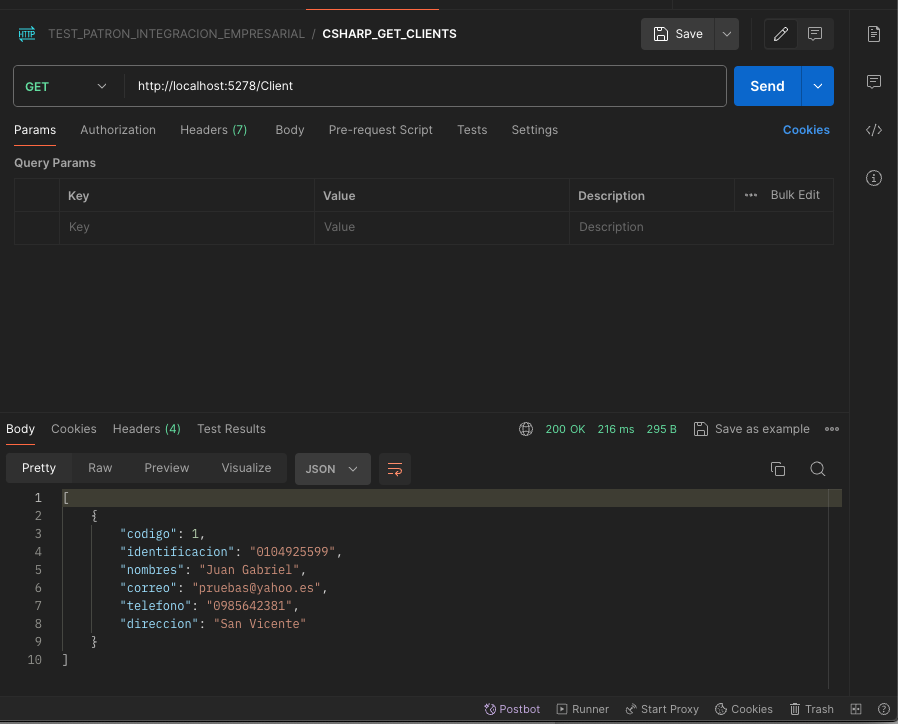
Para esto debemos ingresar a la carpeta contenedora del proyecto.

**/laboratorio\_integracion\_empresarial/netmicro**

* 1. Una vez dentro procedemos a ejecutar los siguientes comandos en una terminal en Linux o CMD Windows.
* **dotnet clean**
* **dotnet build**
* **dotnet run**
  1. Una vez ejecutado los comandos ya tendremos operativo nuestro servicio haciendo uso del endpoint haciendo uso del método GET.

<http://localhost:5278/Client>

* 1. Una vez iniciado el servicio ya podremos probar el servicio el cual nos presentara un listado json inicial.



1. **Ejecución de Proyecto spring-camel-integration, construido con Sprint boot y camel.**
   1. El primer paso es ingresar a nuestro proyecto de Sprint Camel Integration.
   2. Procedemos a lanzar los comandos para construcción del servicio y su respectiva ejecución, para esto ejecutamos los siguientes comandos.

**/Users/juannaula/Documents/CLASES\_MAESTRIA/PATRONES\_INTEGRACION\_EMPRESARIAL/apache-maven-3.9.6/bin/mvn clean package -DskipTests (Mac)**

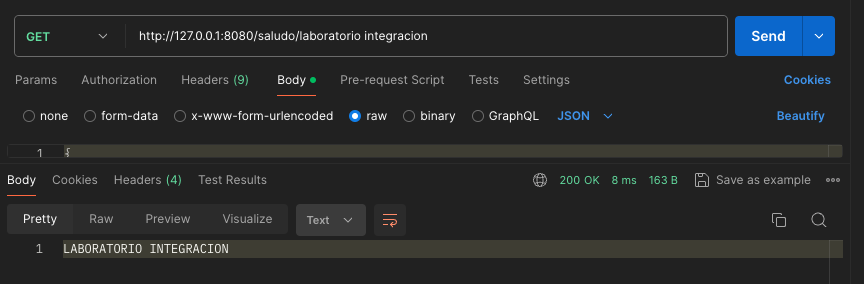
**mvn clean package -DskipTests (Windows)**

* 1. Iniciamos nuestro servicio con el siguiente comando.

**java -jar target/spring-camel-integration-1.0.0-SNAPSHOT.jar**

* 1. Una vez iniciada el servicio procedemos a probar que este operativo para lo cual consumimos el servicio con el uso del método GET

**http://127.0.0.1:8080/saludo/laboratorio integracion**



1. **Pruebas funcionales de servicios integrados.**
   1. Ejecución del servicios para crear clientes en base a la condición ingresada en el **Json ("codigoEmpresa":1), en este caso cuando el código empresa sea 1 el nuevo cliente se registrada en el Servicio de python para lo cual se ejecuta mediante el link de tipo POST.**

[**http://localhost:8080/addClienteEmp**](http://localhost:8080/addClienteEmp)

**{**

**"codigo": 4,**

**"identificacion": "0104925594",**

**"nombres": "Karolain Naula",**

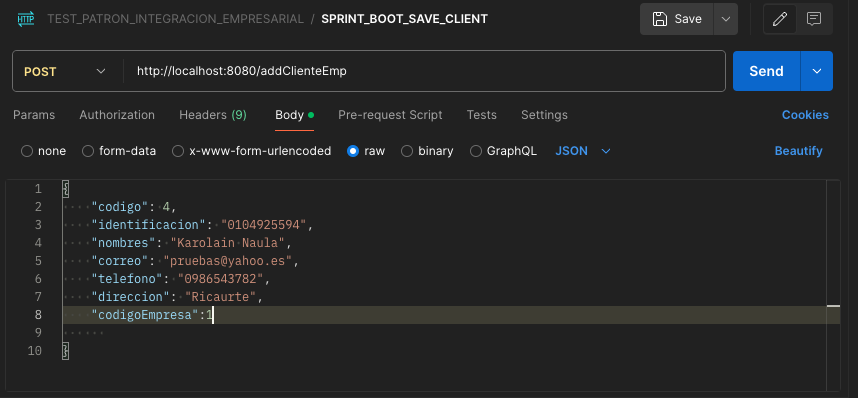
**"correo": "pruebas@yahoo.es",**

**"telefono": "0986543782",**

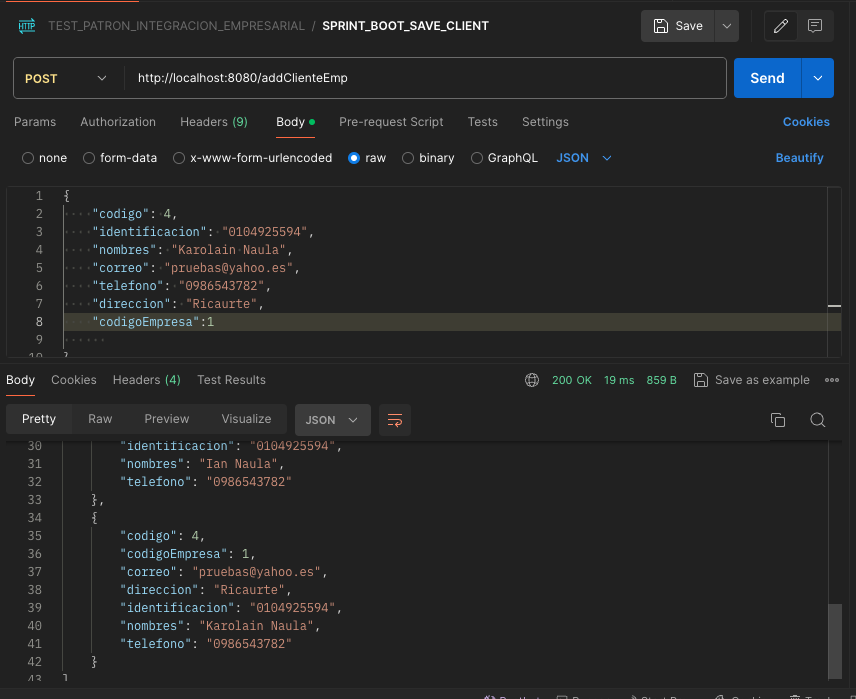
**"direccion": "Ricaurte",**

**"codigoEmpresa":1**

**}**

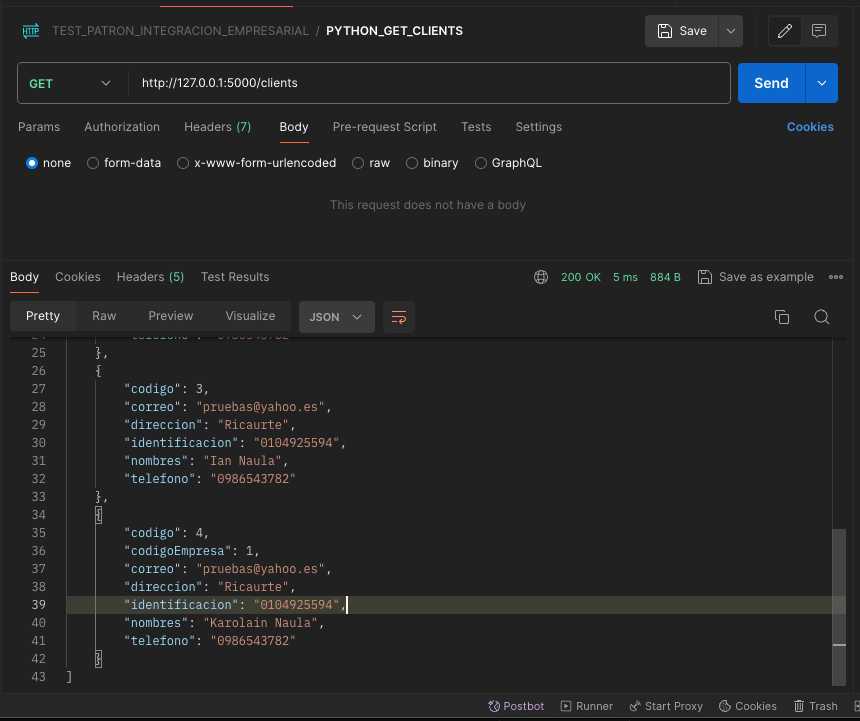
****

* 1. Resultado registro nuevo cliente, para este caso el servicio retornara una lista de clientes incluida el nuevo cliente creado.



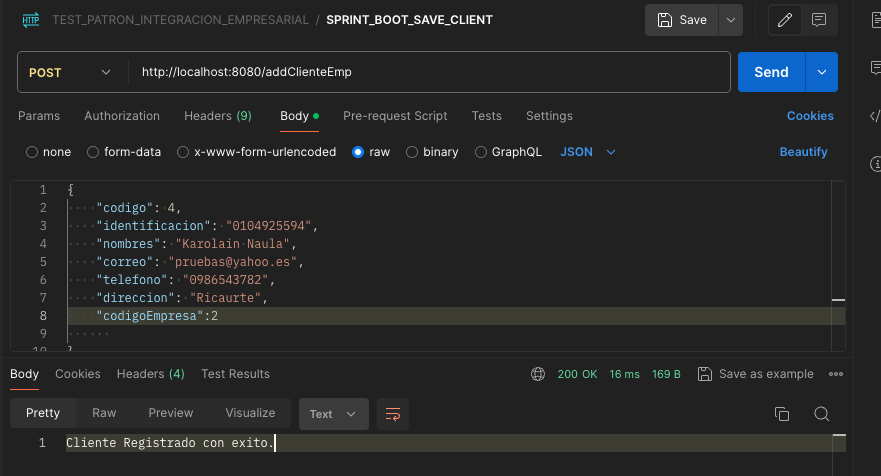
* 1. De igual manera procedemos a probar el registro existente dentro del servicio de Python, con el método GET.

<http://127.0.0.1:5000/clients>



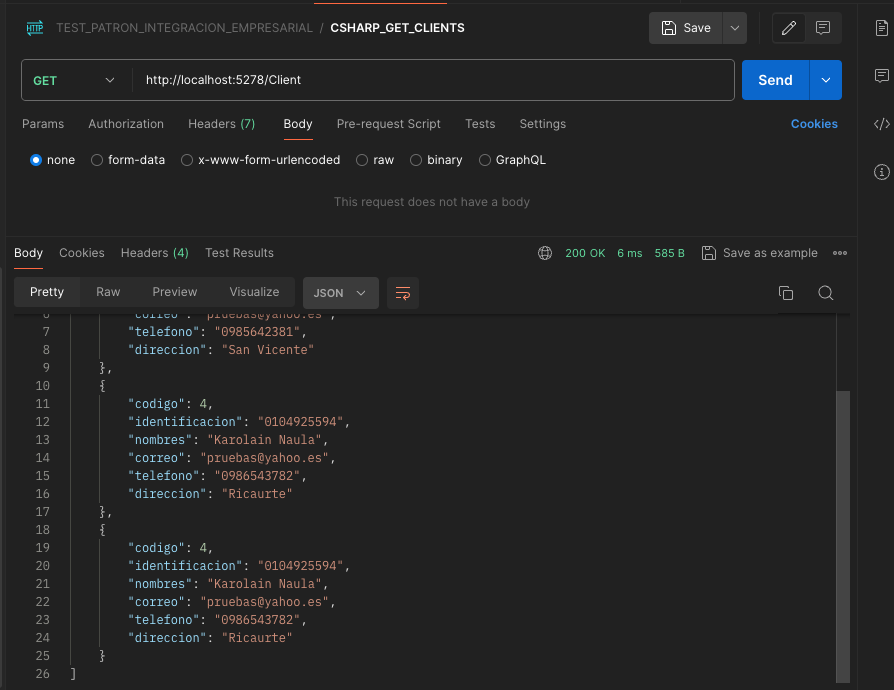
* 1. **Pruebas Funcionales cambiando el codigoEmpresa (2), para este caso el servicio realizara el registro en el servicio de .NetCore por lo cual al ejecutar el servicio de registro de nuevos Clientes retornara el mensaje (Cliente registrado con exito)**

[**http://localhost:8080/addClienteEmp**](http://localhost:8080/addClienteEmp)



* 1. **Validación registro en Servicio .net. con método GET.**

[**http://localhost:5278/Client**](http://localhost:5278/Client)

****

1. Fin.