

Documentación de la Práctica

Mulesoft Trainee

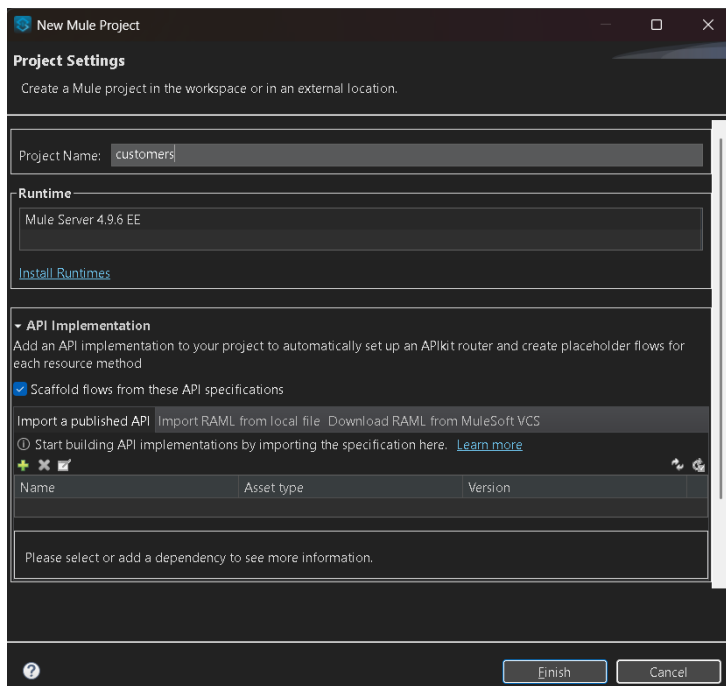
Proyecto Creado en AnypointStudio
para realizar la consuta a una base de
datos mediante una API

Realizado por Julio César Méndez
Torres

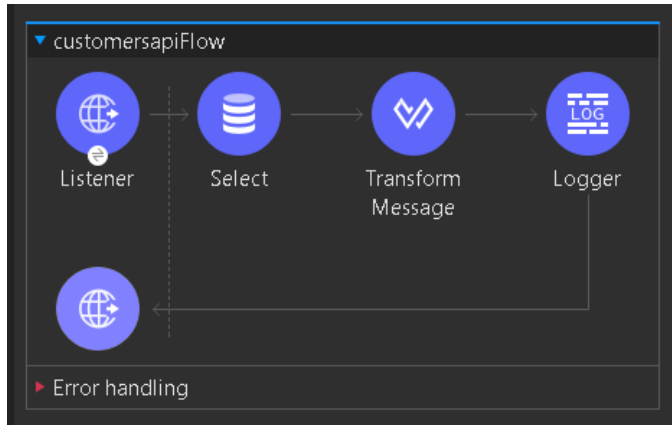
Indice

1. Creación del proyecto.....	1
2. Curtomers.xml.....	2
3. Creación de archivos local y dev.....	5
4. Global.xml.....	6
5. Especificación API.....	7
6. Desplegar el proyecto en Cloudhub.....	8
7. Api Manager.....	9
8. API AutoDiscovery.....	10
9. Client-id Enforcement.....	11

1. Se crea un proyecto en el IDE AnypointStudio con el nombre “customers”



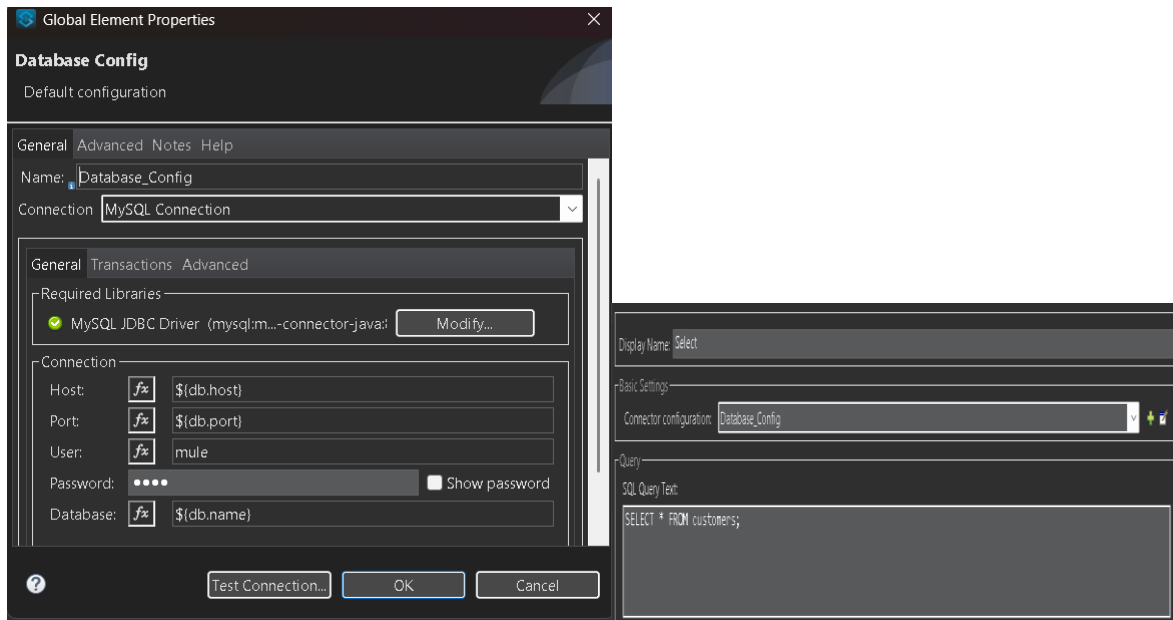
2. En customers.xml se crea un canvas con los componentes Listener, Select, Transform Message y Logger



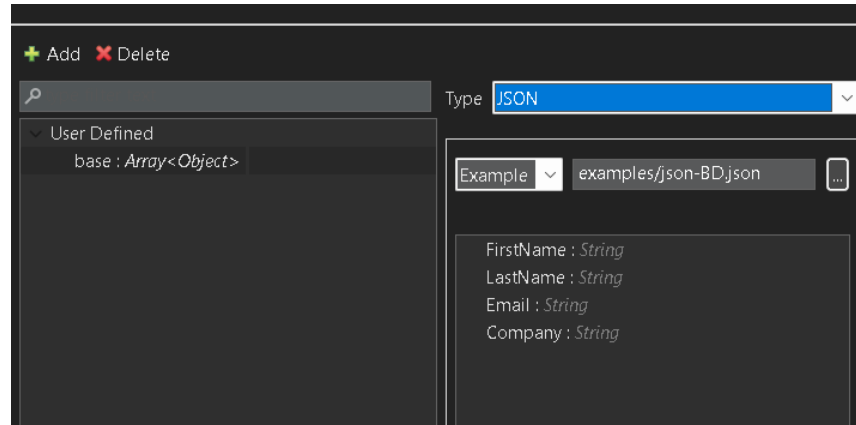
En Listener se asigna la ruta /api/v1/sps/customers donde se verán las peticiones HTTP



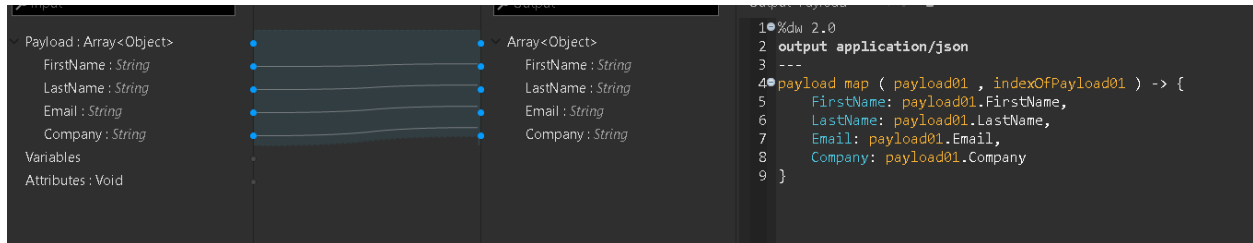
En Select se colocan los parámetros para realizar la conexión a la base de datos y también se añade el query para hacer la consulta a la tabla customers



Y en metadata se creo un ejemplo de un JSON con las mismas columnas de la tabla de la base de datos que esperamos obtener



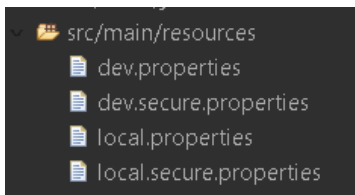
En Transform Message, la información obtenida de la base de datos se convierte a tipo JSON siguiendo el mismo ejemplo que creamos anteriormente



Y se crea un Logger con el mensaje “Información recibida”, que se mostrara en la consola si todo sale bien



3. En src/main/resources se crean 4 archivos



local.properties y dev properties donde crearemos variables de los parámetros de conexión de la base de datos, donde serán colocados en el Selct de nuestro canvas

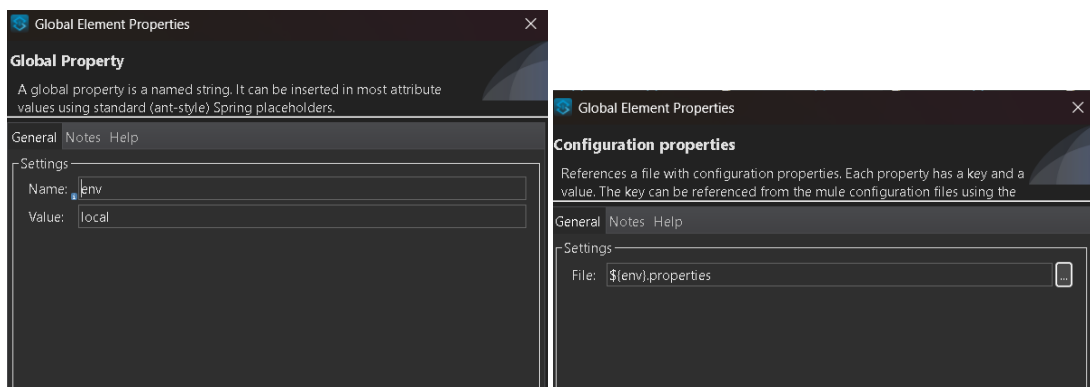
```
1 db.host=mudb.learn.mulesoft.com
2 db.port=3306
3 db.name=training
4 db.user=${secure::db.user}
5 db.password=${secure::db.password}
6
```

dev.secure.properties y local.secure.properties es donde crearemos las variables user y password cifradas previamente con secure-properties-tool.jar que ejecutaremos en la terminal

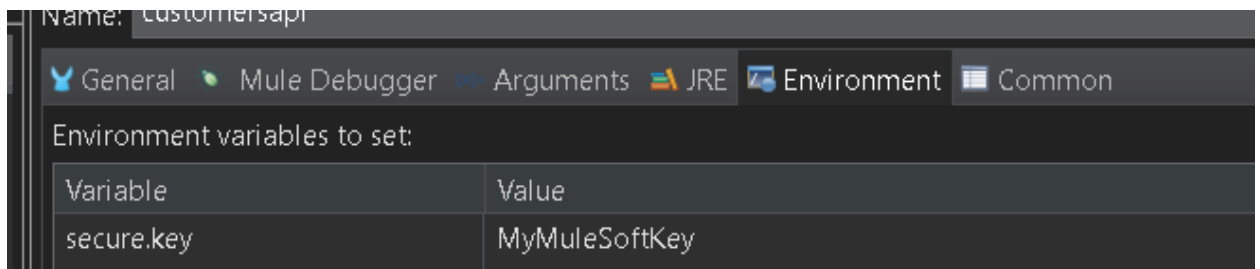
```
1 db.user=[hn5PdGLGVPw=]
2 db.password=[hn5PdGLGVPw=]
```

```
julio@Julio_Mendez MINGW64 ~/Downloads
$ java -cp secure-properties-tool.jar com.mulesoft.tools.SecurePropertiesTool \
string \
encrypt \
Blowfish \
CBC \
MyMuleSoftKey \
"mule"
hn5PdGLGVPw=
```

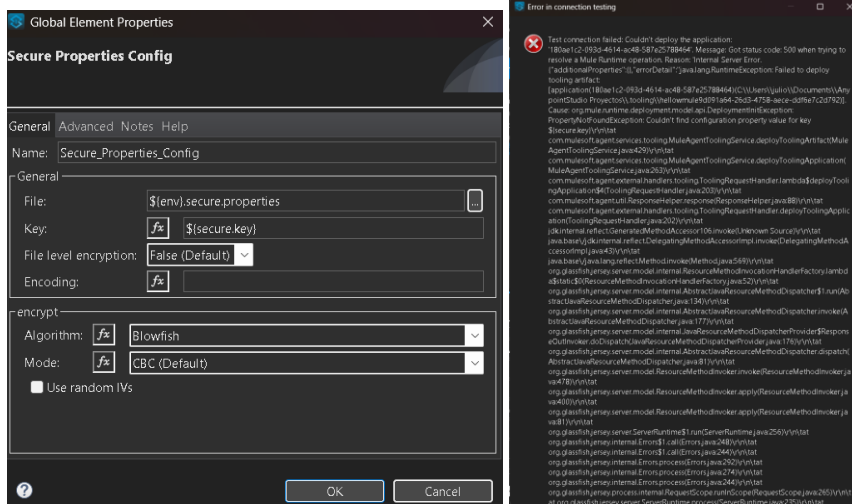
4. Se crea el archivo global.xml donde estará env para poder cambiar entre local o dev de acuerdo a su valor



En Run configurations crearemos la variable secure.key con el valor MyMuleSoftKey que servira como llave para desenscriptar las variables user y password



Y se coloca de igual manera en global.xml, aunque yo tuve que quitarlo debido a un error que no me dejaba hacer la conexión con la base de datos que no fui capaz de solucionar



5. Se crea la especificación de la API customers-api.raml

De nombre Customers API, con la ruta /api/v1/sps y con el mismo formato de la tabla customers, siendo un GET al ser solo de consulta

```
1 ##RAML 1.0
2 title: Customers API
3 version: v1
4 baseUri: /api/v1/sps
5
6 /customers:
7   get:
8     description: Returns all customers
9     responses:
10       200:
11         body:
12           application/json:
13             example: |
14               [
15                 {
16
17                   "FirstName": "Maura",
18                   "LastName": "Wagstaffe",
19                   "Email": "mwagstaffe@npr.org",
20                   "Company": "Kazu"
21                 }
22               ]
```

6. Lo desplegaremos en Cloudhub, antes debemos de iniciar sesión en la plataforma para poder desplegarlo, y ya solo le indicaremos que suba nuestros proyecto

Applying changes will create a new configuration for your application **Apply Changes**

Application File

Deployment Target Ingress Properties Logging

Runtime version

Release Channel **Edge** new

Runtime Version **4.9.6.See**

Java Version new

Using Java 17 may require a different application resource profile. [Learn more](#)

☐ Java 8 ☒ Java 17

Replicas

Replica Count **1**

Replica size **0.1 vCores**

Y en la pagina de Anypoint Runtime Manager podemos ver que ya esta corriendo

customersapi **Stop** **▼** **🔔**

Application status: **Running**

Configuration: 611c2a

Last updated: 2025-06-25 11:15:52AM

Replicas: 1/1 started

Public Endpoint: <https://customersapi-cxo7dt.5sc6y6-1.usa-e2.cloudhub.io>

Target name: Cloudhub-US-East-2

Target type: Shared Space

Configuration 611c2a **Apply Changes**

Applying changes will create a new configuration for your application

Application File **customersapi.jar** **Choose file**

Version 1.0.0

Deployment Target Ingress Properties Monitoring

Con la siguiente URL <https://customersapi-cxo7dt.5sc6y6-1.usa-e2.cloudhub.io/api/v1/sps/customers> verificamos que esta funcionando, trayendo en tipo JSON los datos de la tabla customers de la base de datos Training

```
[{"FirstName": "Maura", "LastName": "Wagstaffe", "Email": "mwagstaffe@npr.org", "Company": "Kazu"}, {"FirstName": "Myrwyn", "LastName": "Eliet", "Email": "meliet1@buzzfeed.com", "Company": "Skibox"}, {"FirstName": "Myrwyn", "LastName": "Eliet", "Email": "meliet1@buzzfeed.com", "Company": "Skibox"}, {"FirstName": "Jenn", "LastName": "White", "Email": "jwhite@wamu.org", "Company": "NPR"}, {"FirstName": "Kelly", "LastName": "McEvers", "Email": "kmcEvers@npr.org", "Company": "NPR"}, {"FirstName": "Audi", "LastName": "Cornish", "Email": "audie@npr.org", "Company": "NPR"}, {"FirstName": "Ari", "LastName": "Shapiro", "Email": "ari@npr.org", "Company": "NPR"}]
```


7. Api Manager

En la pagina de anypoint en la parte de Api Manager le damos a la opción
Add New API

Y añadimos el archivo raml

Con eso tendríamos creada la API, en base a la que hicimos en el proyecto

Once the API is created it will be published in Exchange in stable state.

Name: Customer-API

Asset types: REST API

Method: Upload a RAML

File upload: customers-api.raml

Main file: customers-api.raml

Advanced >

Instance Administration / Add API

Runtime API Downstream Upstream Review

Downstream

Configure the API instance settings related to inbound traffic.

* Required field

Client provider: Anypoint

Instance label (Optional):

Recommended if you have multiple managed instances of the same API

Advanced options >

Review

Review your selections before saving and deploying your API instance. You can also save the configuration and deploy it later.

Runtime

Runtime type: Mule Gateway

Proxy type: Basic Endpoint

API

API name: Customer-API

API version: v1

Asset version: 1.0.0

Downstream

Scheme: HTTP

API instance label: customers-api-dev

Upstream

Status	Name	Runtime	Label	Version	Instance	Error Rate	Total Requests	Client Applications	Creation Date
Unregistered	Customer-API	Mule 4	customers-api-dev	v1	20410740	No data	No data	0	06-25-2025

8. API AutoDiscovery

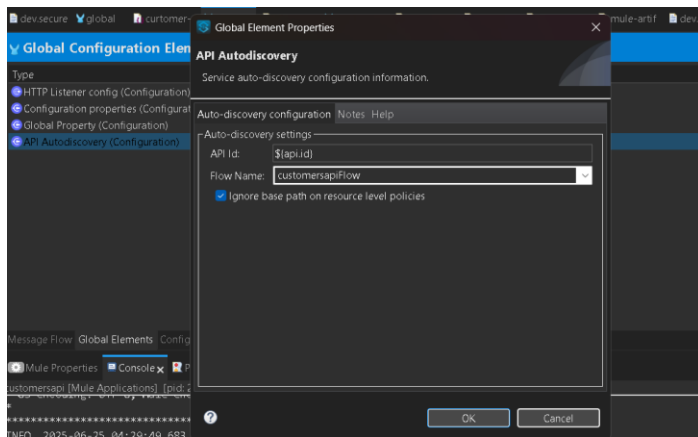
Una vez que ya se tiene la API subida en api manager se copia el ID

Implementation URI ⓘ	API Label ⓘ
https://customersapi-oxo7dt.5sc6y6-...	customers-api-dev
Consumer Endpoint	API Instance ID ⓘ
N/A	20410740
Instance Conformance ⓘ	
Not Validated	

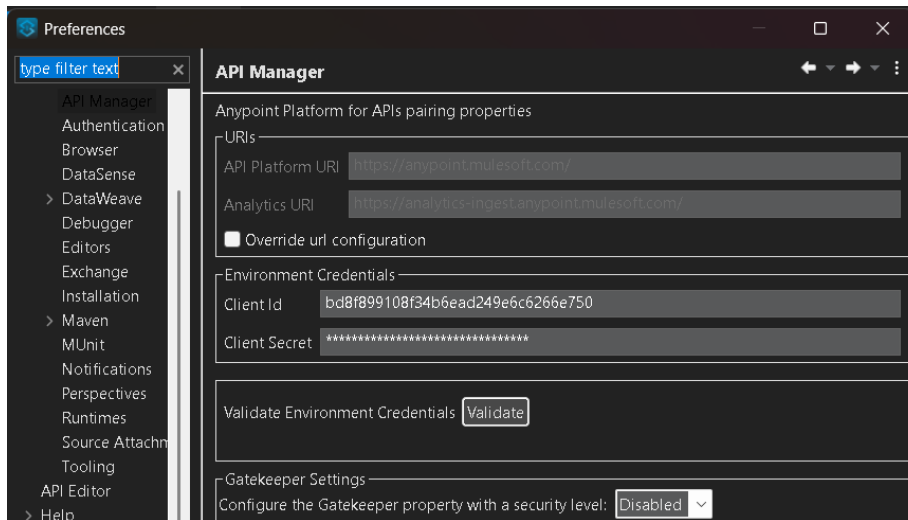
Y se coloca en una variable en dev.properties y local.properties

```
1 db.host=mudb.learn.mulesoft.com
2 db.port=3306
3 db.name=training
4 db.user=${secure::db.user}
5 db.password=${secure::db.password}
6
7 api.id=20410740
```

Y la utilizamos en global.xml



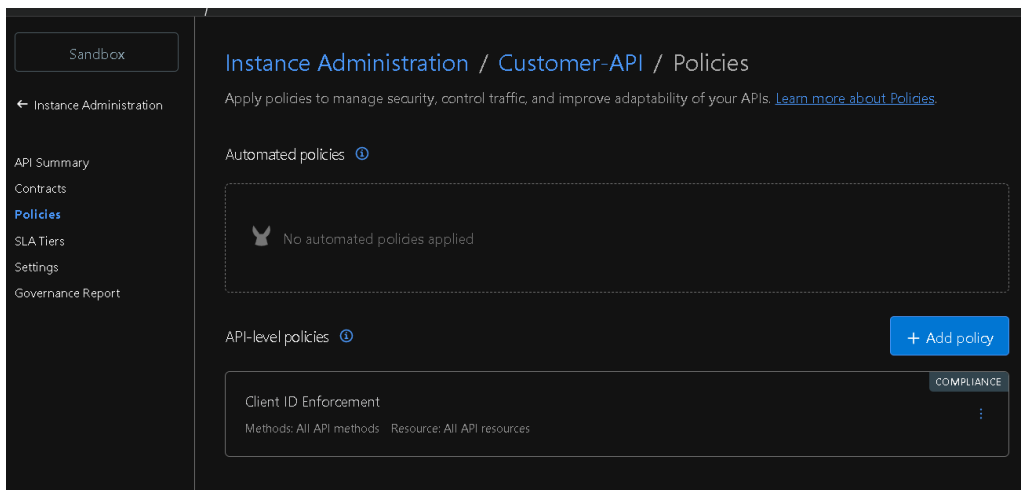
En preferencias añadimos el client id y secret, y con eso ya integrariamos la API al proyecto



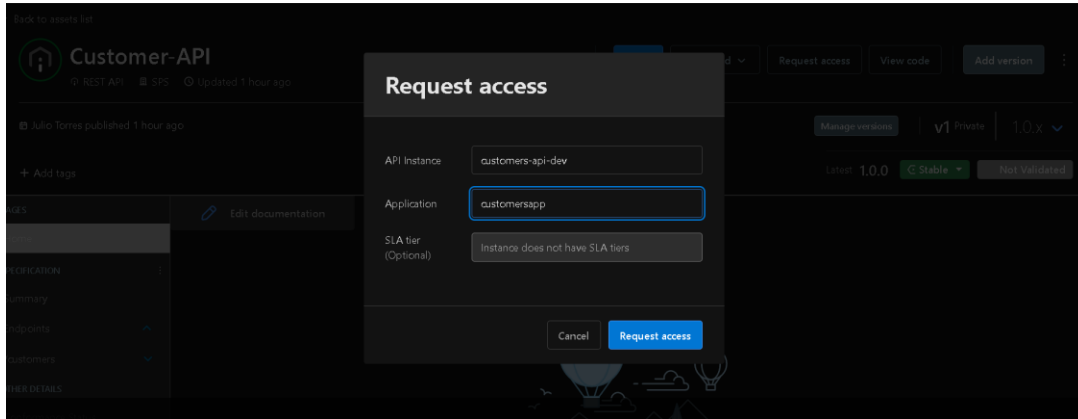
9. Client-id Enforcement

En API manager seleccionamos nuestra api y vamos a la parte de policies

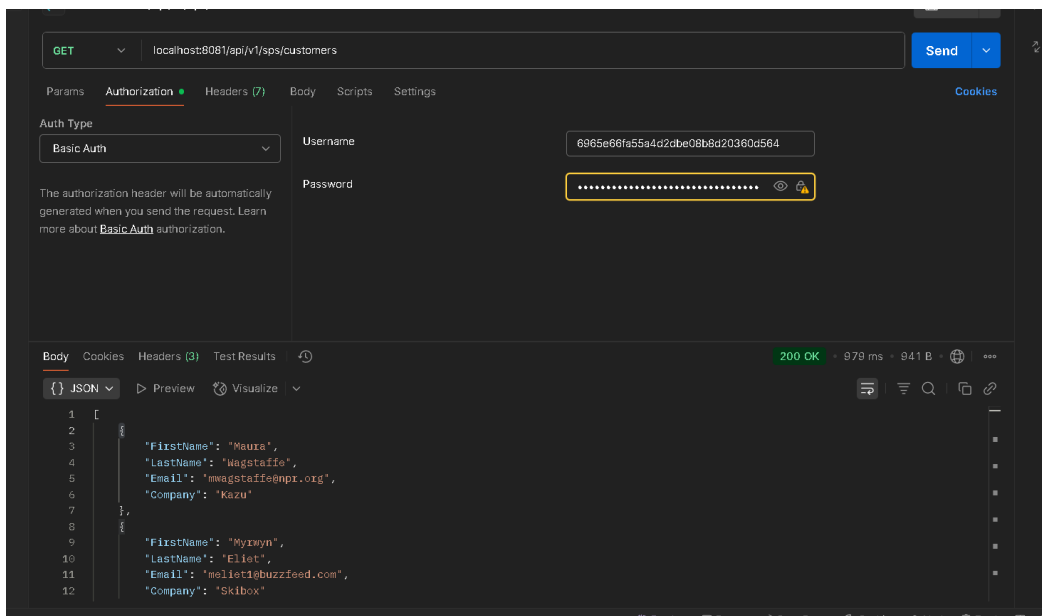
Donde añadimos Clien ID Enfocement



Ahora en Exchange seleccionamos la API y vamos a donde dice Request Access y creamos una aplicación



Colocando el username y el password ya podremos hacer peticiones http a la api



Adjunto el username y el password para que puedan ejecutar el proyecto y usar la API

Environment Information:

Client ID:

bd8f899108f34b6ead249e6c6266e750

Client secret:

D9797B8de8B347c0967110D44D8c97f1

Request Access

Client ID:

6965e66fa55a4d2dbe08b8d20360d564

Client Secret:

1ABB7Bb1D59949Df8D7455D3D9E3b351