

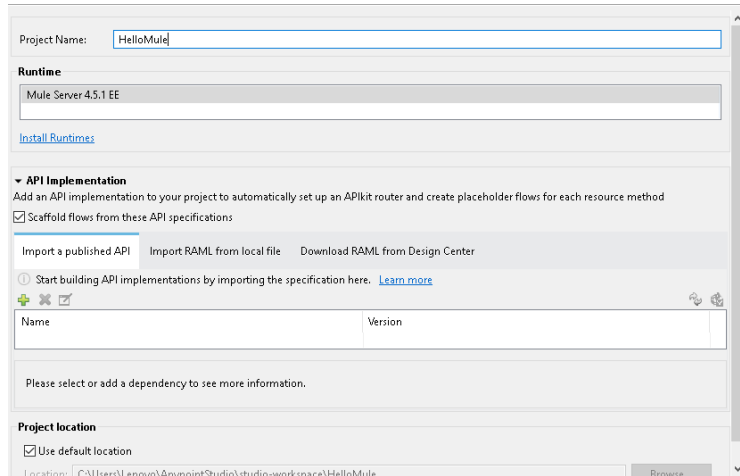
Contenido

PRACTICA 1. HELLO MULE	2
Implementación Localmente.....	2
Implementación en CloudHube	4
PRACTICA 2. Configurar elementos globales y archivos de propiedades en Anypoint Studio.	7
Cree un archivo global.xml para mantener los elementos globales.	7
Externalizar los valores codificados en propiedades.....	8
Configurar el elemento global Propiedades de configuración	10
PRACTICA 3. Cómo proteger las propiedades antes de la implementación en Anypoint Studio.	13
Crear los archivos de propiedades seguros de sus entornos	13
Configure un registrador para leer sus credenciales seguras.....	15
Cifre los valores de sus archivos de propiedades	16
PRACTICA 4. Cómo proteger las propiedades antes de la implementación en Anypoint Studio.....	19
Cree una nueva API en el Administrador de API	19
Configurar el Administrador de API	21
Configurar Anypoint Studio con descubrimiento automático de API.....	21
Pruebe su aplicación en su computadora local	22
PRACTICA 5. Cómo aplicar la política de cumplimiento de ID de cliente a su aplicación Mule en API Manager.....	26
Establecer la política de cumplimiento de ID de cliente en API Manager	26

PRACTICA 1. HELLO MULE

Implementación Localmente.

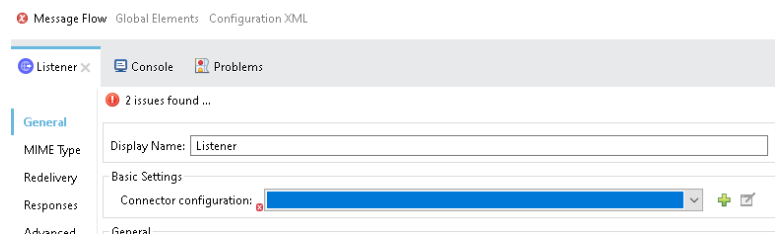
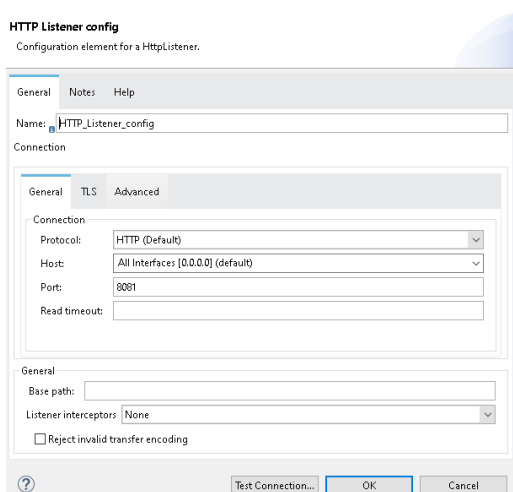
PASO 1. Crear un nuevo Proyecto en AnyPoint Studio y llamarlo “HelloMule”.



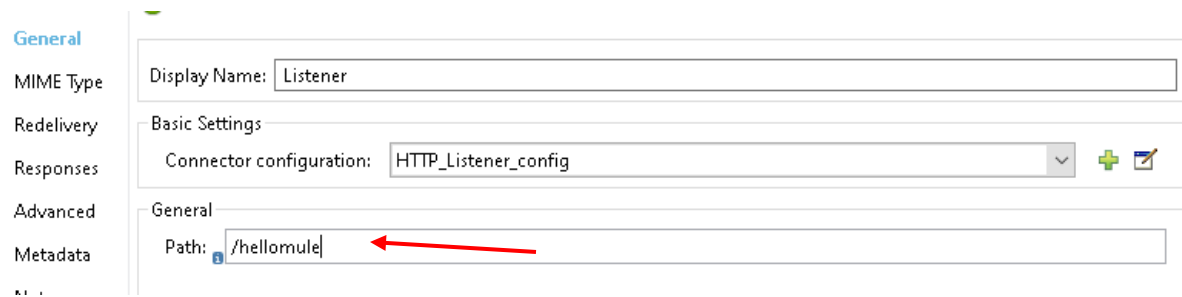
PASO 2. Arrastrar un **Listener HTTP** desde la **Mule Palette** a la hoja en blanco.



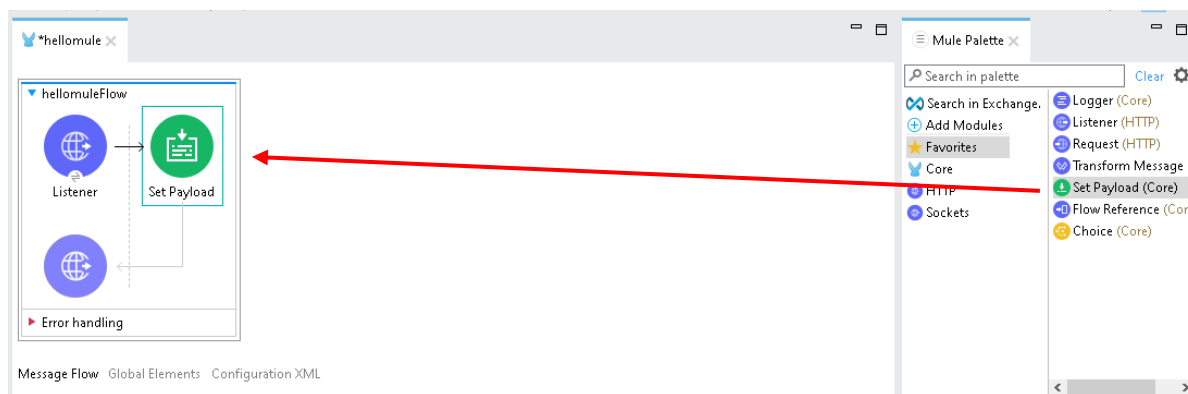
PASO 3. Dar clic en el conector Listener HTTP , dirigirse a donde se encuentran Basic Settings, ahí mismo darle en el botón verde de agregar en donde se abrirá una configuración global y verificar los valores predeterminados ‘Anfitrión : 0.0.0.0’ y ‘Puerto: 8081’ .



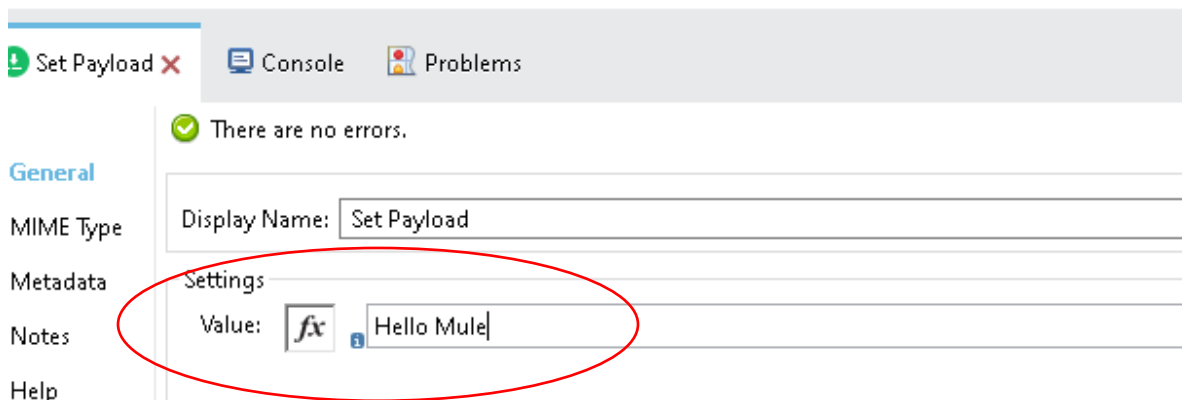
PASO 4. En la parte de General agregar la siguiente ruta: **/hellomule**.



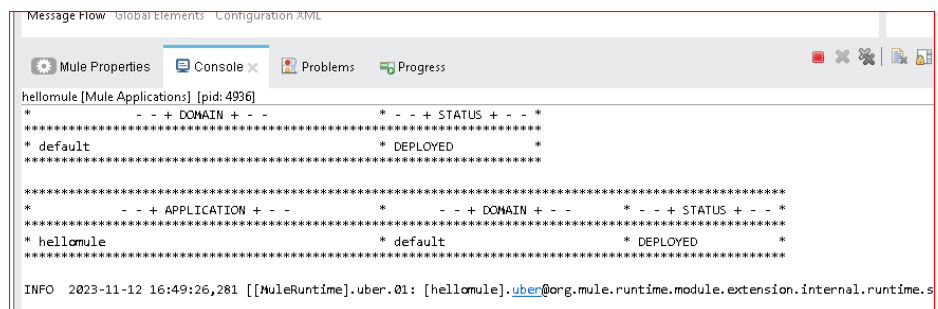
PASO 5. Arrastrar el conector **SetPayload** desde la **Mule Palette** al flujo dentro del Proceso.



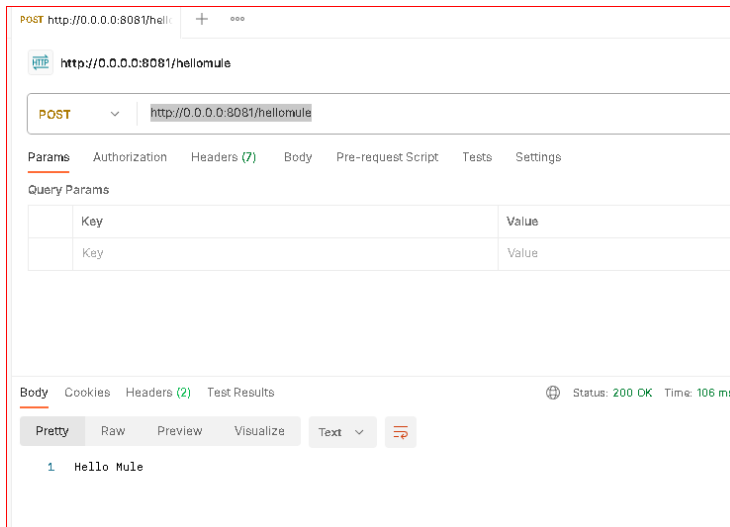
PASO 5. Dirigirse al apartado de **Settings**, Desmarcar el botón fx y poner la cadena **Hello Mule**.



PASO 6. Guardaremos el proyecto en la **Pestaña de File** y daremos clic en **Save**. Después de esto ejecutaremos la aplicación en modo local. Daremos clic derecho dentro del Lienzo y escogemos donde dice **Run Project HelloMule**. Si la aplicación funciona de manera correcta deberá salir esta ventana.

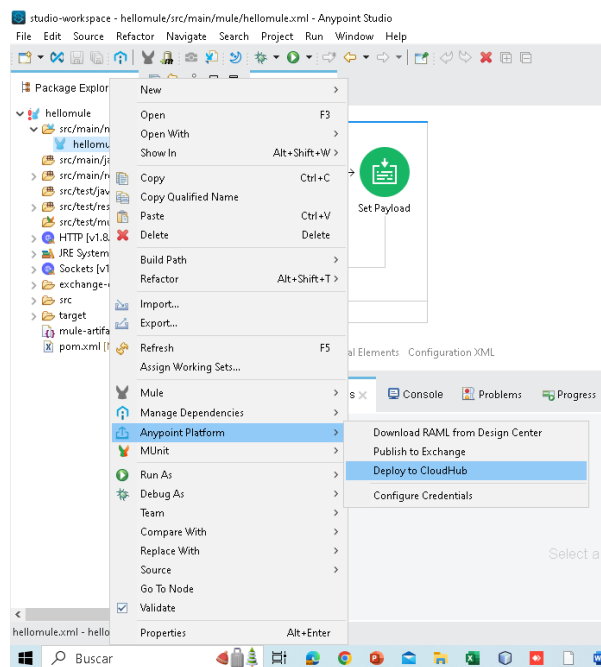


PASO 7. Nos dirigimos a Postman , donde escogeremos **Post** donde pondremos la siguiente ruta <http://0.0.0.0:8081/hellomule> para consultar localmente.

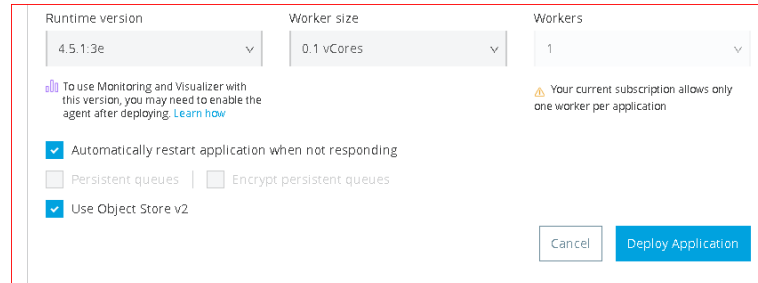
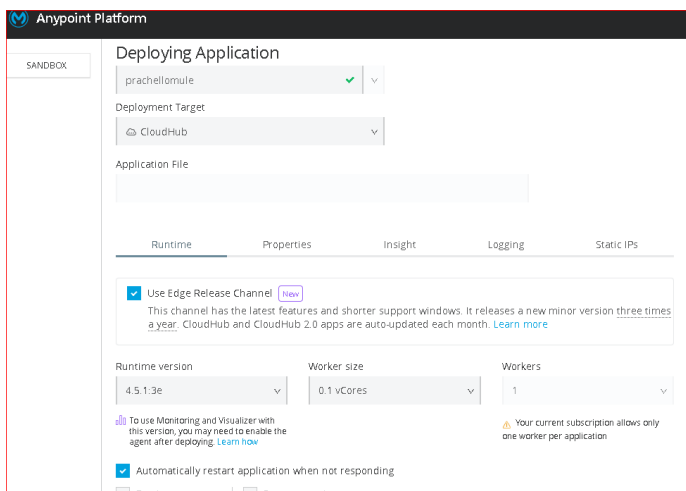


Implementación en CloudHube

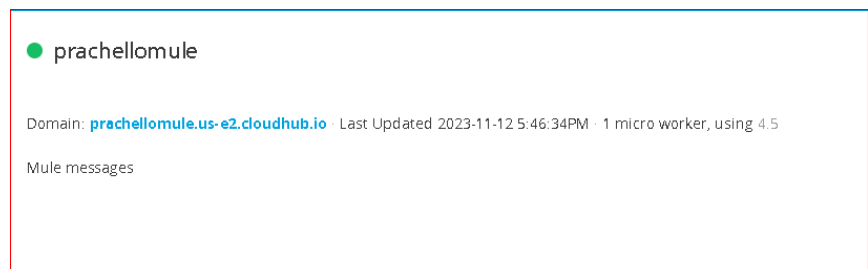
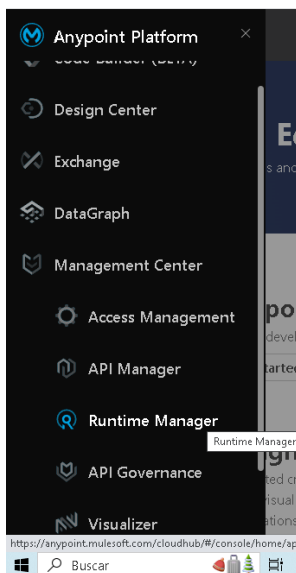
PASO 1. Nos al apartado donde se encuentra la aplicación y le daremos donde se encuentra el xml, en este caso **hellomule.xml**



PASO 2. Nos vamos a loggear con el usuario y contraseña que tenemos, escogeremos donde **SANBOX** y le agregaremos un nombre personalizado hasta que aparezca una palomita al lado del nombre. Para finalizar le daremos en Deploy Application.



PASO 3. Nos dirigimos a este link <https://anypoint.mulesoft.com/home/> y abrimos el apartado de **RunTime Manager**, buscamos la aplicación y le damos clic para abrir el panel , buscamos el URL y agregamos la ruta después de la URL , en este caso seria asi <prachellomule.us-e2.cloudhub.io/hellomule>



PASO 4. Nos dirigimos a **PostMan** y ponemos la URL <prachellomule.us-e2.cloudhub.io/hellomule>

POST prachellomule.us-e2.clo

prachellomule.us-e2.cloudhub.io/hellomule

Save

POST

prachellomule.us-e2.cloudhub.io/hellomule

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Body

Cookies

Headers (4)

Test Results

Status: 200 OK

Time: 632 ms

Size: 125 B

Save Response

Pretty

Raw

Preview

Visualize

Text

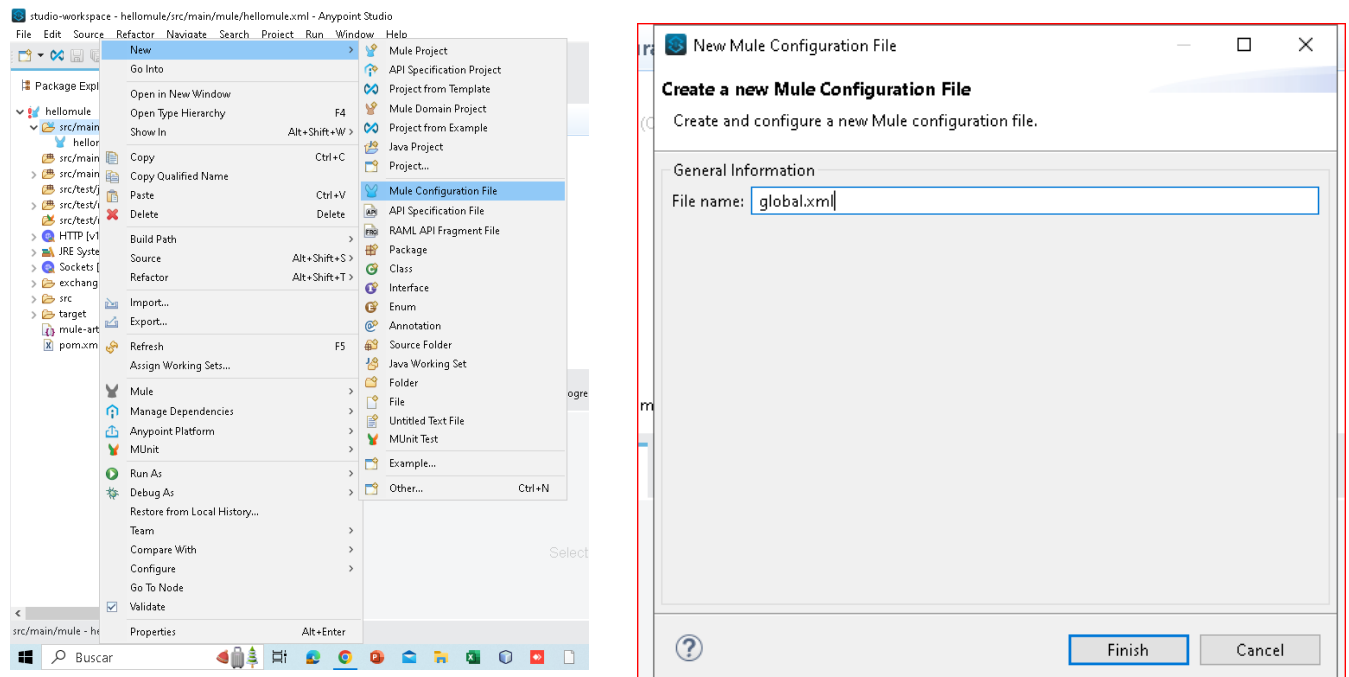
1

Hello Mule

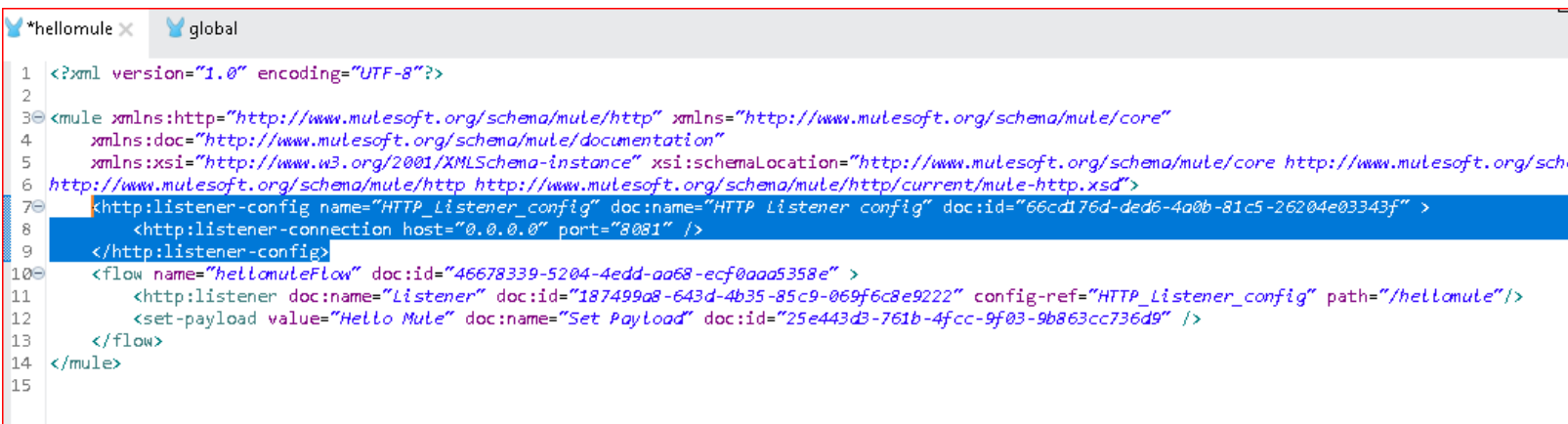
PRACTICA 2. Configurar elementos globales y archivos de propiedades en Anypoint Studio.

Cree un archivo `global.xml` para mantener los elementos globales.

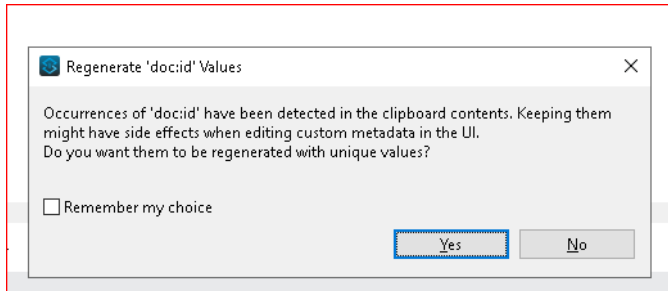
PASO 1. Nos dirigimos a la aplicación que creamos en el punto anterior y en la parte de `src/main/mule` le daremos clic derecho, damos en **New/Mule Configuration File** y creamos un archivo llamado **global.xml**.



PASO 2. Nos vamos a la aplicación de `mule.xml`, nos cambiamos a la vista de xml localizamos la etiqueta `http:listener-config` y cortamos todo lo que esta seleccionado.

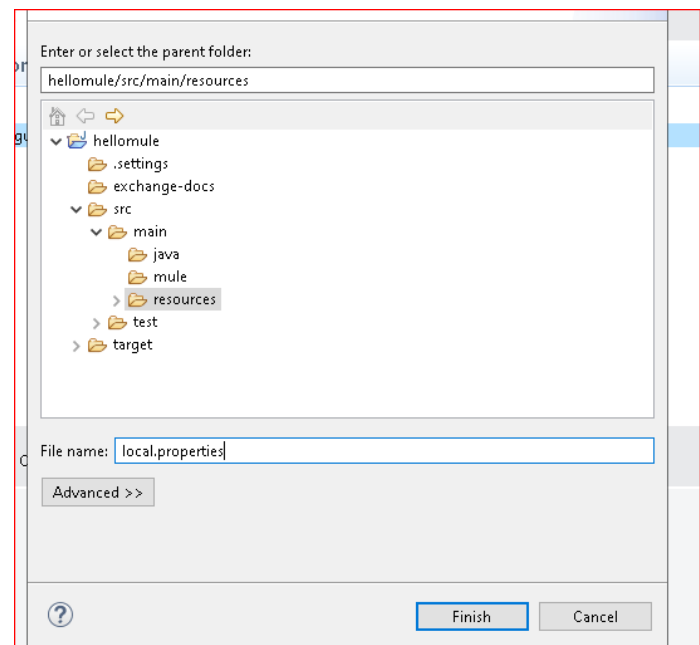
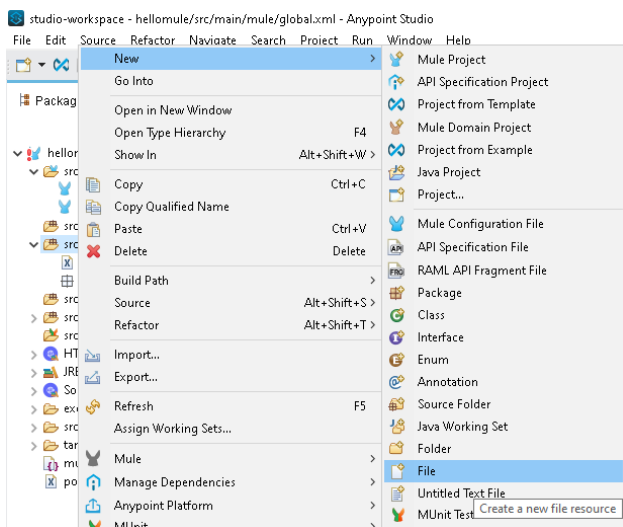


PASO 3. El código que se corto en el paso anterior lo pegaremos en el archivo **global.xml**, si sale un aviso le daremos que si y guardamos el archivo.



Externalizar los valores codificados en propiedades

PASO 1. Nos colocamos sobre **src/main/resources** , le daremos clic derecho y seguimos esta ruta **New/File** y creamos un archivo **local.properties**.

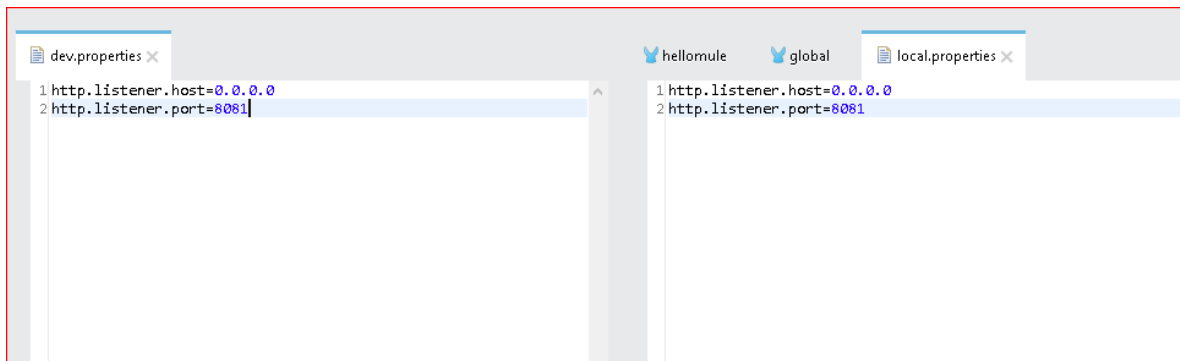


PASO 2. Dentro del archivo pegaremos este texto

http.listener.host=0.0.0.0

http.listener.port=8081

Después crearemos de la misma manera otro archivo llamado **dev.properties** y pegaremos el mismo texto.



PASO 3. Nos regresamos al global.xml en **Global Elements** al cual le daremos doble clic , en el panel de configuración pegamos este sintaxis **`${http.listener.host}`** y **`${http.listener.port}`** en la parte de **Host y Post**

General Notes Help

Name: HTTP_Listener_config

Connection

General TLS Advanced

Connection

Protocol: HTTP (Default)

Host: \${http.listener.host}

Port: \${http.listener.port}

Read timeout:

General

Base path:

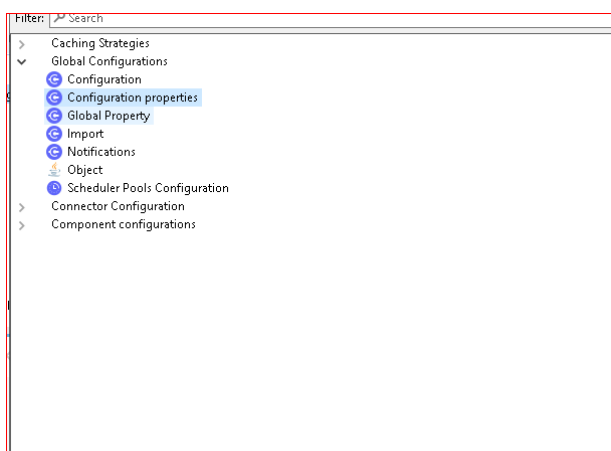
Listener interceptors: None

☐ Reject invalid transfer encoding

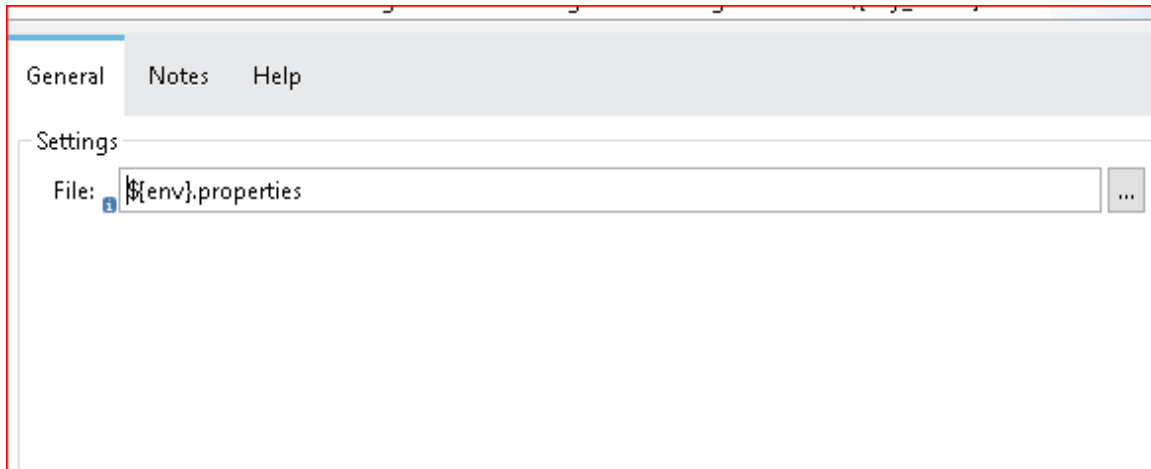
Test Connection... OK Cancel

Configurar el elemento global Propiedades de configuración

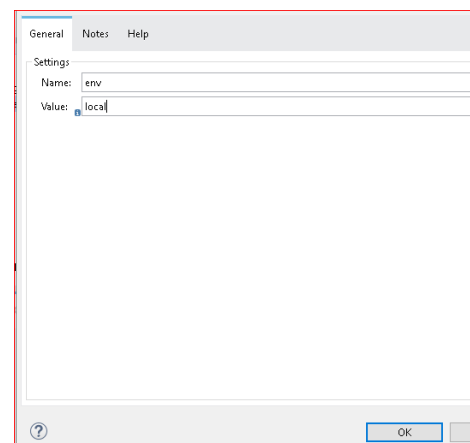
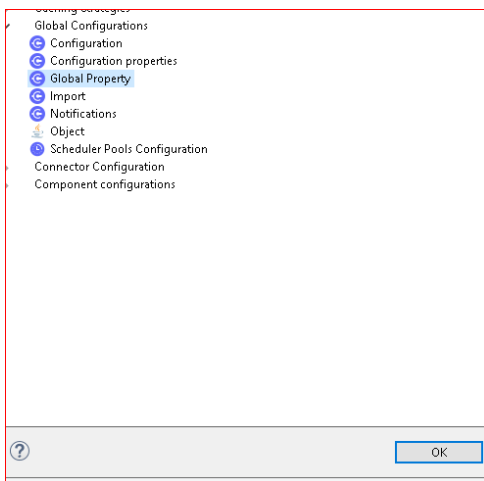
PASO 1. Nos dirigimos al archivo global.xml en el apartado de **Global Elements** daremos clic derecho y en la parte de **Crear** le damos clic para crear un nuevo elemento. En el apartado de **Global Elements** dentro de crear buscamos **Configuration Properties**, lo seleccionamos y le damos en **Ok**.



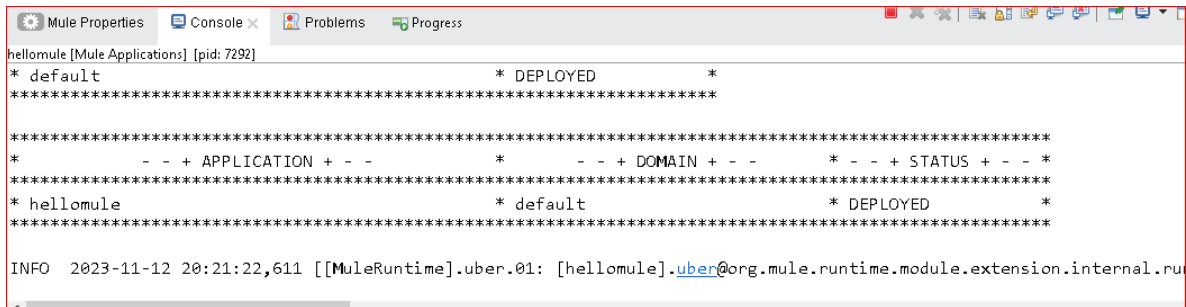
PASO 2. Agregamos el siguiente archivo de propiedades **`${env}.properties`** en donde dice **File**.



PASO 3. Ahora crearemos una **Global Property**, en donde en el **Name: env** y en **Value: local**



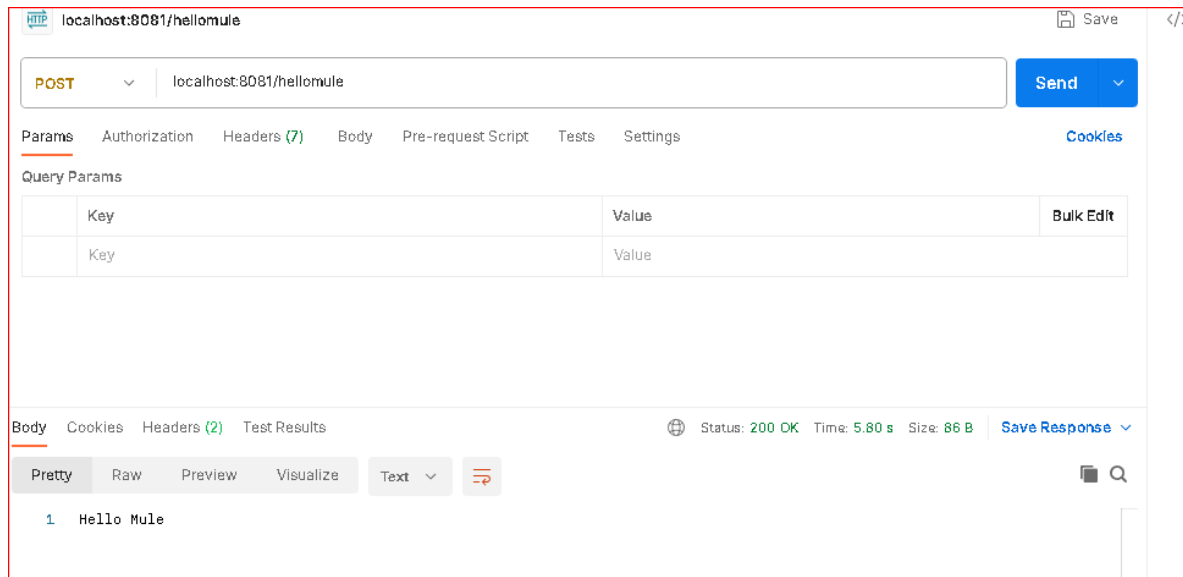
PASO 4. Verificaremos que la aplicación siga funcionando, para eso daremos clic derecho en el lienzo y clic en **Run Project hellomule**



The screenshot shows the Mule IDE console with the following output:

```
hellomule [Mule Applications] [pid: 7292]
* default * DEPLOYED *
*****
*****
* - - + APPLICATION + - - * - - + DOMAIN + - - * - - + STATUS + - - *
*****
* hellomule * default * DEPLOYED *
*****
INFO 2023-11-12 20:21:22,611 [[MuleRuntime].uber.01: [hellomule].uber@org.mule.runtime.module.extension.internal.ru
```

PASO 5. Por ultimo nos dirigimos a PostMan y ponemos esta URL **localhost:8081/hellomule** para verificar que funcione.



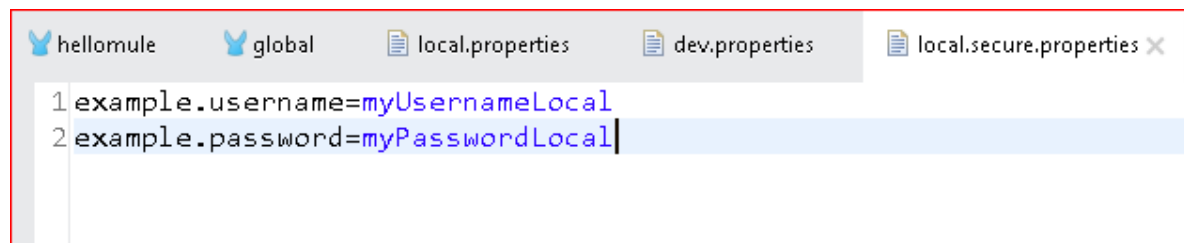
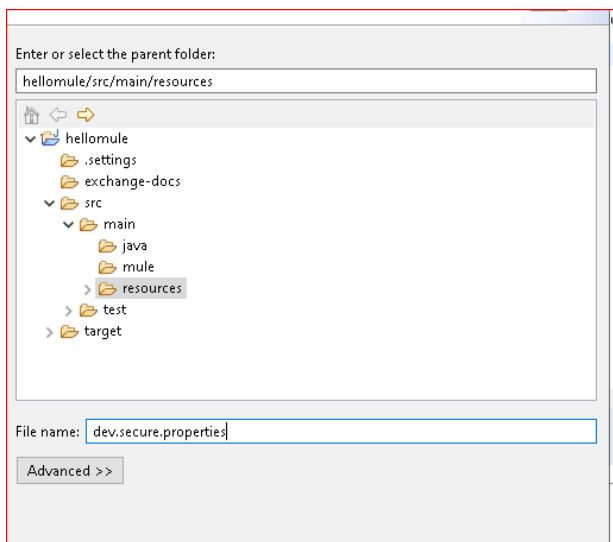
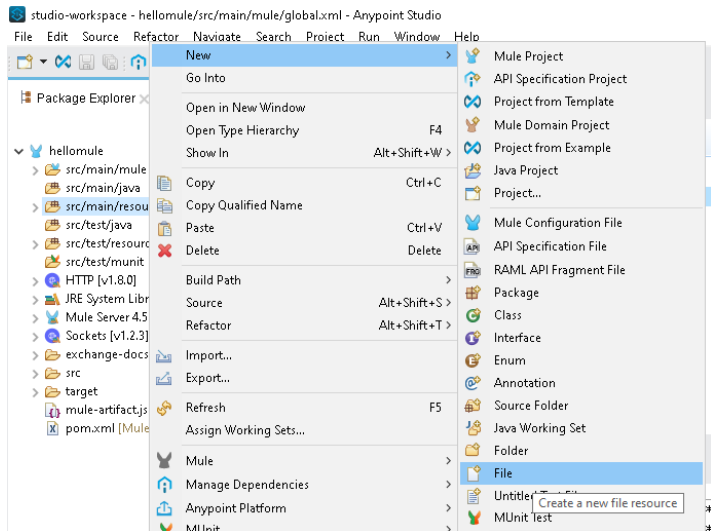
PRACTICA 3. Cómo proteger las propiedades antes de la implementación en Anypoint Studio.

Crear los archivos de propiedades seguros de sus entornos

PASO 1. Crearemos un **local.secure.properties**, para crearlo hacemos clic derecho en **src/main/resources** le damos clic en **New/File** al cual llamaremos **local.secure.properties** y le ponemos este texto

example.username=myUsernameLocal

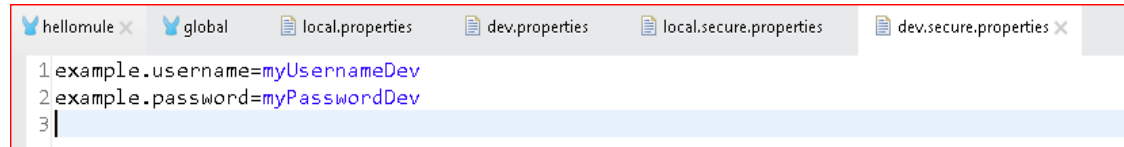
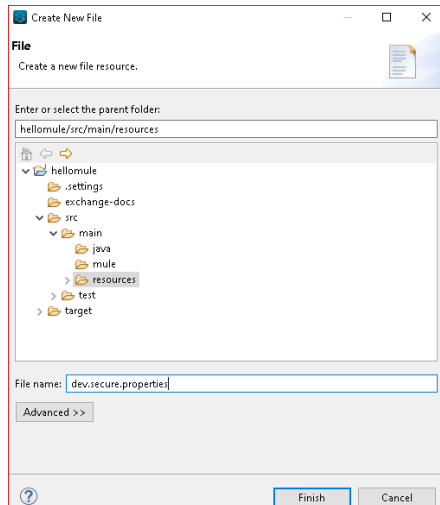
example.password=myPasswordLocal



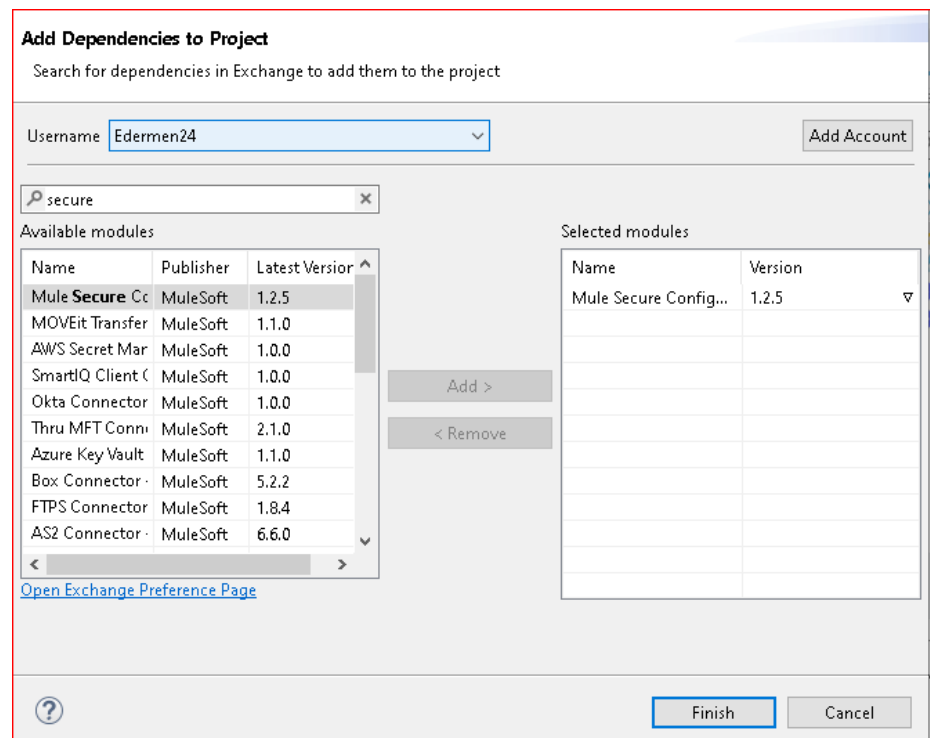
PASO 2. Ahora crearemos uno que se llame **dev.secure.properties** y ponemos este texto

example.username=myUsernameDev

example.password=myPasswordDev



PASO 3. Nos dirigimos al archivo de global.xml y en la **Mule Palette** buscamos la opción de **Search in Exchange** a la cual le daremos clic derecho, se nos abrirá una ventana en el cual buscaremos el modulo **secure** y lo agregamos.

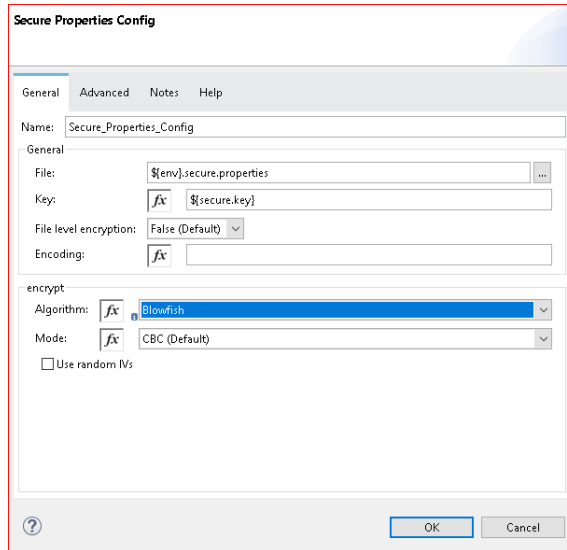


PASO 4. Nos dirigimos a la parte de **Global Elements** y le damos en crear, ahí agregaremos una **Secure Properties Config**, en donde configuraremos lo siguiente, en la parte de

File: `${env}.secure.properties`

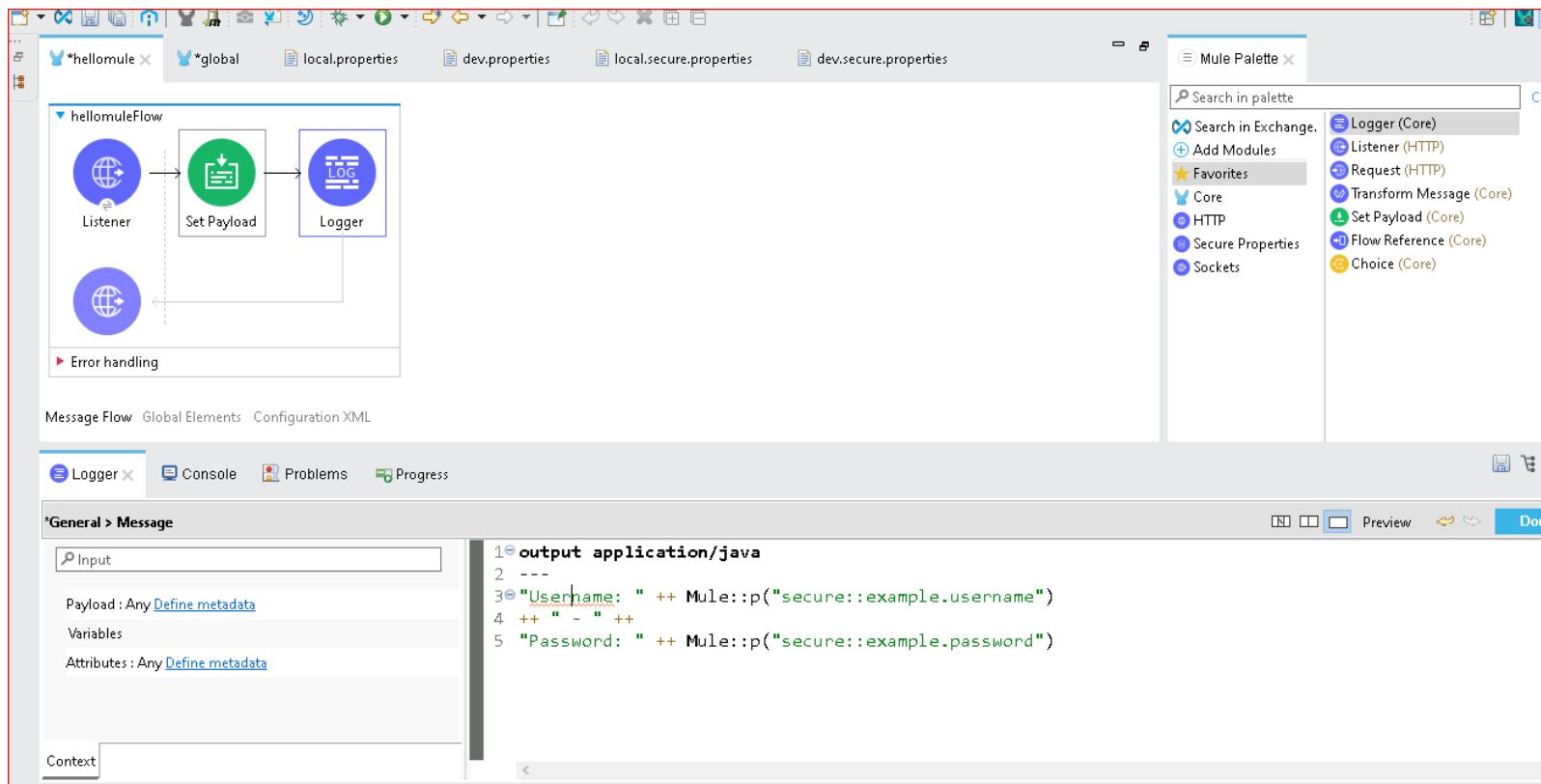
Key: `${secure.key}`

Algorithm: Blowfish



Configure un registrador para leer sus credenciales seguras

PASO 1. Nos dirigimos al archivo `hellomule.xml` donde agregaremos desde la **Mule Palette** un componente **Logger**. Luego le daremos en el botón de **fx** y en la opción de **Show Graphical View**, y nos vamos a la opción de **Source Only** para poder pegar este siguiente código.



Cifre los valores de sus archivos de propiedades

PASO 1. Descargaremos de este link https://docs.mulesoft.com/mule-runtime/latest/secure-configuration-properties#secure_props_tool el archivo **Secure Properties Tool Jar file**.

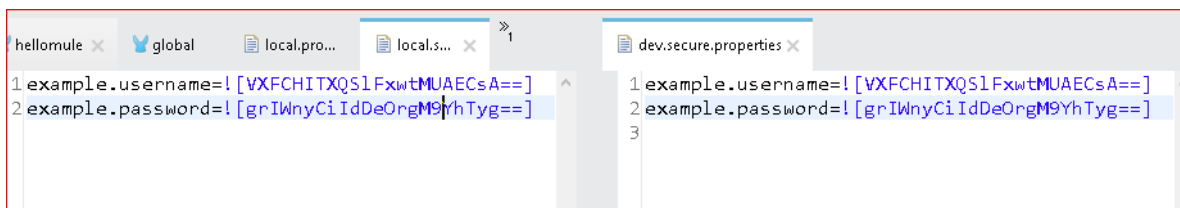
Abrimos la cmd y ponemos el siguiente código **C:\Users\Lenovo\Downloads>java -cp secure-properties-tool.jar com.mulesoft.tools.SecurePropertiesTool string encrypt Blowfish CBC mulesoftKey "myUsernameLocal"**

Para que nos de el usuario encriptado. **VXFCHITXQSIFxwtMUAECsA==**

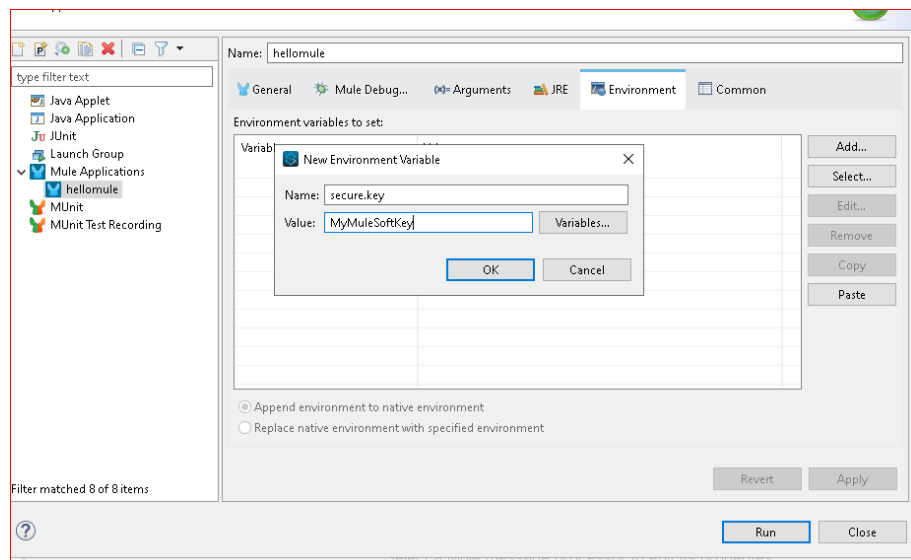
Y hacemos lo mismo para la contraseña **C:\Users\Lenovo\Downloads>java -cp secure-properties-tool.jar com.mulesoft.tools.SecurePropertiesTool string encrypt Blowfish CBC mulesoftKey "myPassLocal"**

Para que nos de la contraseña encriptada. **grIWnyCildDeOrgM9YhTyg==**

PASO 2. Agregamos lo encriptado en local.secure.properties y dev.secure.properties. De la siguiente manera es como se agregará.



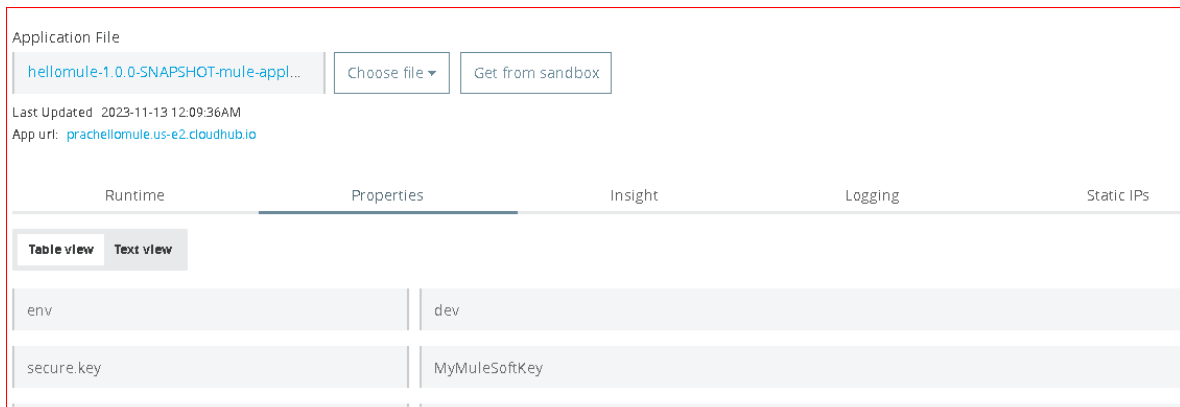
PASO 3. Nos vamos a donde se encuentra el Proyecto , nos posicionamos encima de el y le damos clic derecho, nos vamos hasta Run As , y le ponemos en run configurations. Se nos abrirá una pantalla en donde nos iremos a la parte de Environment y agregaremos en la parte de Name: secure.key y en Value: MyMuleSoftKey



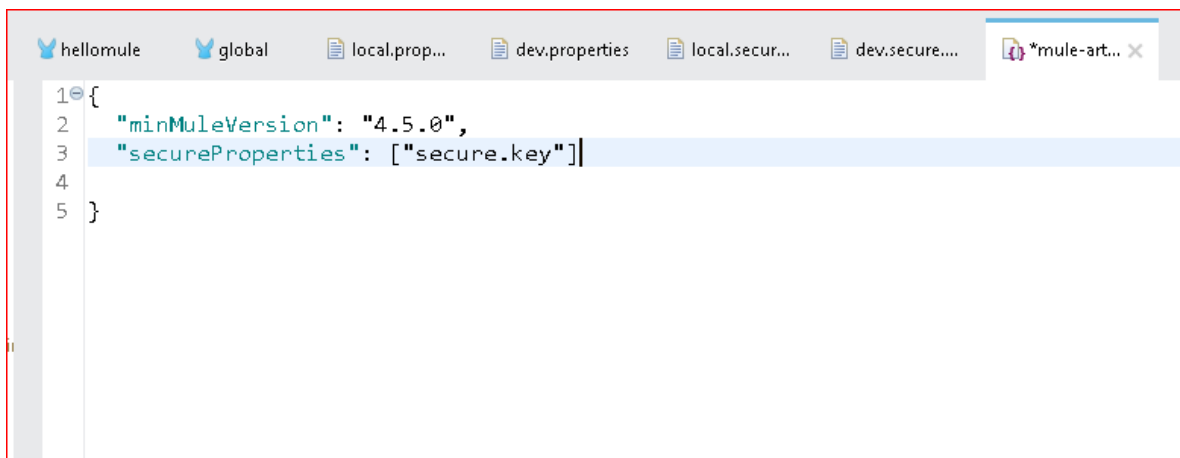
PASO 4. Después de configurar Le ponemos en Apply y en Run para probarlo.

PASO 5. Nos vamos a la plataforma AnyPoint , de ahí escogemos el apartado de RunTime Maneger, buscamos la parte de Settings , seleccionamos el properties y agregamos dos propiedades.

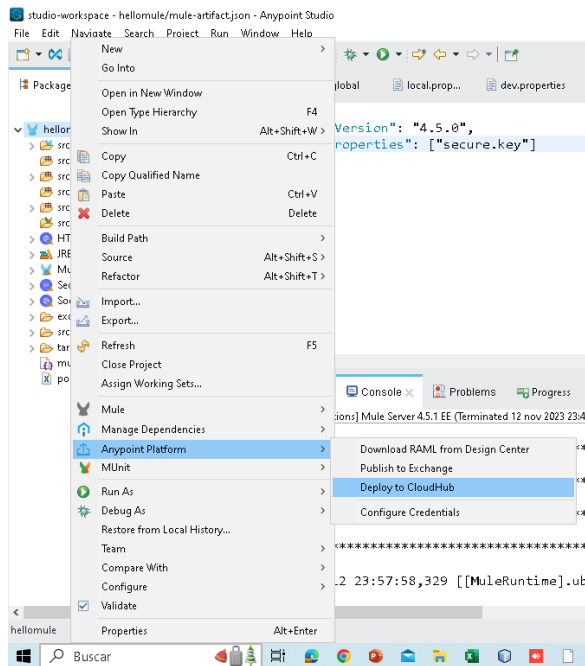
Env la estableceremos en dev y secure.key en MyMuleSoftKey.



PASO 6. Nos dirigimos al archivo **mule-artifact.json** en AnyPoint Studio donde agregaremos lo siguiente. No olvidar agregar una coma después de "4.5.0"



PASO 7. Nos dirigimos a donde se encuentra el código de hellomule.xml , le damos clic derecho , nos vamos hasta donde dice AnyPoint Platform/ Deploy in Cloudhub. Despues de esto nos brira una ventana en donde le daremos clic en el boton Deploy Application.



PASO 8. Ahora revisamos que la contraseña aparezca encriptada.

prachellomule

St

Application File

hellomule-1.0.0-SNAPSHOT-mule-appl...

Choose file

Get from sandbox

Last Updated: 2023-11-13 12:41:45AM

App url: prachellomule.us-e2.cloudhub.io

Runtime

Properties

Insight

Logging

Static IP

Table view

Text view

env	dev
secure.key	*****
key	value

And

PRACTICA 4. Cómo proteger las propiedades antes de la implementación en Anypoint Studio

Cree una nueva API en el Administrador de API

PASO 1. Nos dirigimos a **Anypoint Platform** dentro de ahí nos vamos **API Manager**, donde daremos clic en el botón azul de **Add Api/Add New Api**, se los abrirá una pantalla donde le daremos en **Mule Gateway** y le damos en el botón de Next.

The screenshot shows the 'API Manager' interface with the 'Add API' tab selected. On the left is a sidebar with navigation links: 'SANDBOX', 'API Administration', 'API Groups', 'Automated Policies', 'Client Applications', 'Custom Policies', and 'Mule API Analytics'. The main content area is titled 'Add API' and contains a 'Select runtime' section. This section has three radio button options: 'Flex Gateway' (marked 'NEW'), 'Mule Gateway' (selected), and 'Service Mesh'. Below this, the 'Proxy type' section has two radio button options: 'Connect to existing application (basic endpoint)' (selected) and 'Deploy a proxy application'. The 'Mule version' section has two radio button options: 'Mule 4 (recommended)' (selected) and 'Mule 3 or below'. At the bottom of the form are 'Cancel' and 'Next' buttons.

PASO 2. En este apartado le daremos en **Create new API** y pondremos estos datos **hellomule en Name** y escogeremos el tipo **API HTTP** y le damos en el botón de next. Y le daremos con los **datos predeterminados que nos salgan**.

This is a close-up of the 'Select runtime' section from the previous screenshot. It highlights the three radio button options: 'Flex Gateway' (marked 'NEW'), 'Mule Gateway' (selected), and 'Service Mesh'. Below this, the 'Proxy type' section shows 'Connect to existing application (basic endpoint)' selected, and the 'Mule version' section shows 'Mule 4 (recommended)' selected. The 'Cancel' and 'Next' buttons are visible at the bottom.

PASO 2.1. Se nos tiene que ver una ventana asi.

Add API

Select the API you want to manage.

☐ Select API from Exchange

☒ Create new API

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

Name

hellomule

Asset types ⓘ

HTTP API

Advanced >

Cancel

Previous

Next

inbox) Add API

Name

hellomule

Asset types ⓘ

HTTP API

Advanced ▾

Exchange generates the `GroupId`, `AssetId` and `Version` for you. If you need, you can change the `AssetId` and `Version`.

GroupId

d7b0c673-b0fa-4b69-be84-4962698037db

AssetId

hellomulee

Version

1.0.0

Version should use the [SemVer](#) format. Examples: 1.0.0 or 2.1.3

API version

v1

Cancel

localhost:8081/hellomule - My Workspace

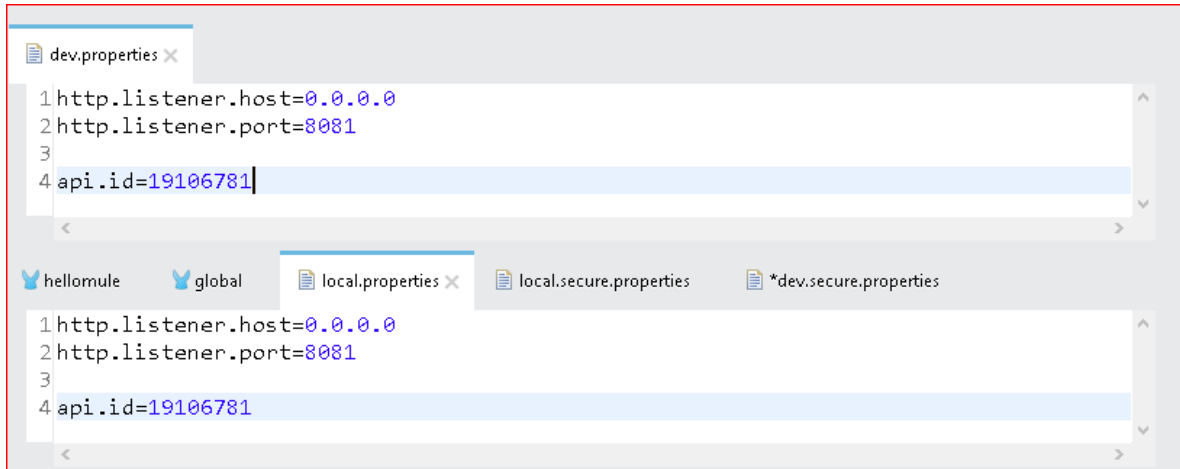
Previous

Next

Type	Asset Version	Implementation URI ⓘ	
HTTP	1.0.0 (Latest)	N/A	
API Label ⓘ	API Version	API Status	
-	v1	<div><div></div>Unregistered</div>	
Consumer Endpoint	API Instance ID ⓘ		
N/A	19106781		
Tags			
<div>ADD A TAG</div>			

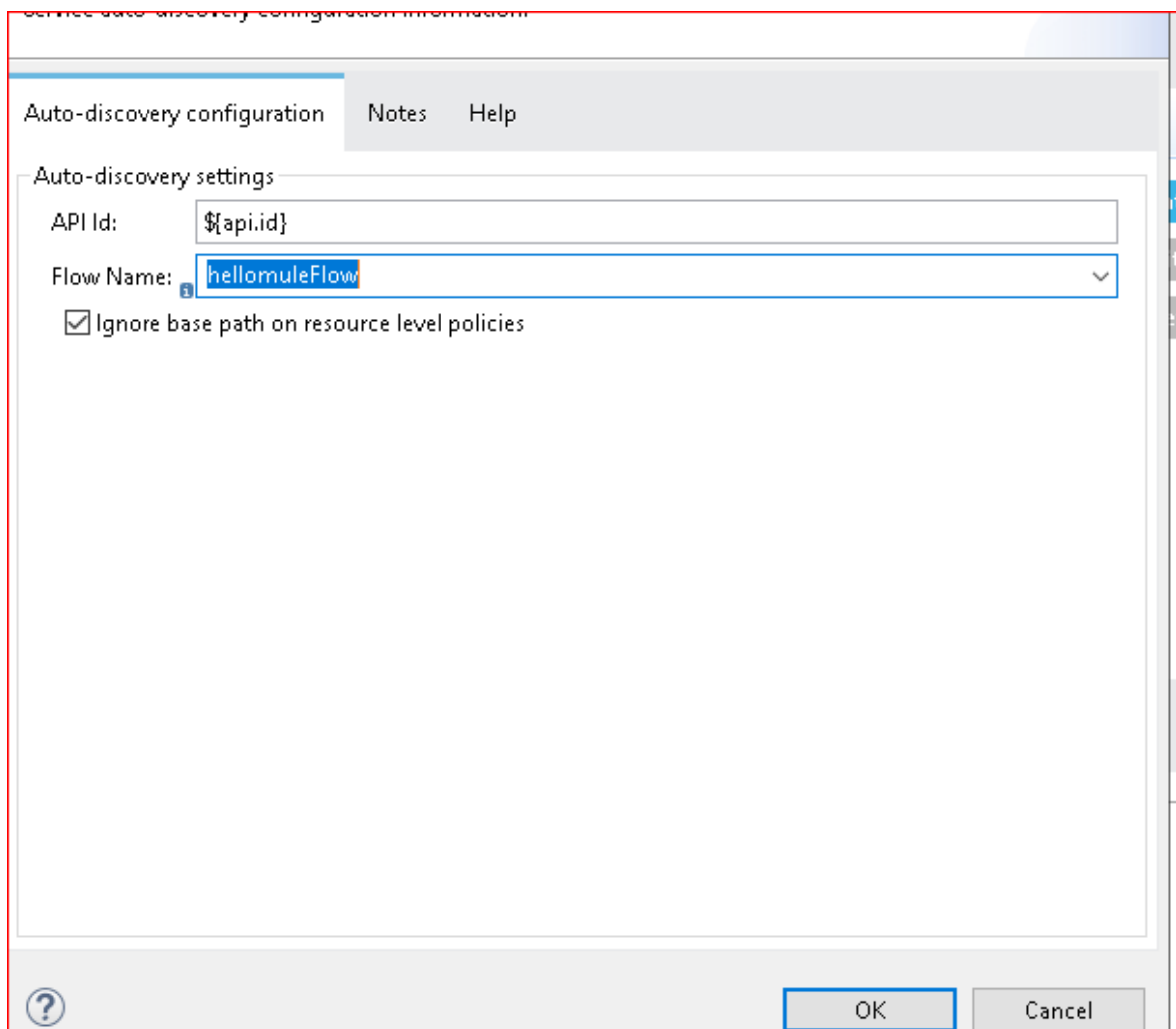
Configurar el Administrador de API

PASO 1. Nos vamos a AnyPoint Studio y ah nos dirigimos a los archivos local.properties y dev.properties para configurar el api.id de la siguiente manera.



Configurar Anypoint Studio con descubrimiento automático de API

PASO 1. Vamos al archivo global.xml en donde cambiaremos a Global Elements , donde crearemos mediante e botón de crear, en donde seleccionaremos un **API Autodiscovery** en donde **configuraremos la siguiente sintaxis \${api.id}**.



Pruebe su aplicación en su computadora local

PASO 1. Regresamos a AnyPoint Platform , nos dirigimos a Access Management , en el apartado de Business Groups, seleccionamos nuestro nombre , y copiamos el Client Id y el Client Secret.

Business Groups / **Prueba**

Settings Access Overview **New** Child Groups Environments Roles Limits

Business Group ID	d7b0c673-b0fa-4b69-be84-4962698037db
Client ID	93e4062c24d6490fb76c52eb50c65064
Client Secret Show

PASO 2. Regresar al Studio, donde nos iremos a la pestaña de Windows y ahí daremos clic en el apartado de Preferences. Dentro del submenú, estará una parte que dice AnyPoint Studio , desplegamos y nos metemos en Api Manager, en donde pegaremos el Client Id y el Client Secret que obtuvimos en el punto anterior . Después de pegarlas, validamos la información, ya por ultimo le damos en aplicar y cerrar.

Preferences

type filter text

- > General
- > Ant
- > Anypoint Studio
 - Analytics
 - Anypoint Platform
 - API Manager**
 - Authentication
 - Browser
 - DataSense
 - > DataWeave
 - Debugger
 - Editors
 - Exchange
 - Installation
 - > Maven
 - MUnit
 - Notifications
 - Perspectives
 - Runtimes
 - Source Attachments
 - Tooling
 - API Editor
- > Help
- > Install/Update

API Manager

Anypoint Platform for APIs pairing properties

URIs

API Platform URI

Analytics URI

☐ Override url configuration

Environment Credentials

Client Id

Client Secret

Validate Environment Credentials

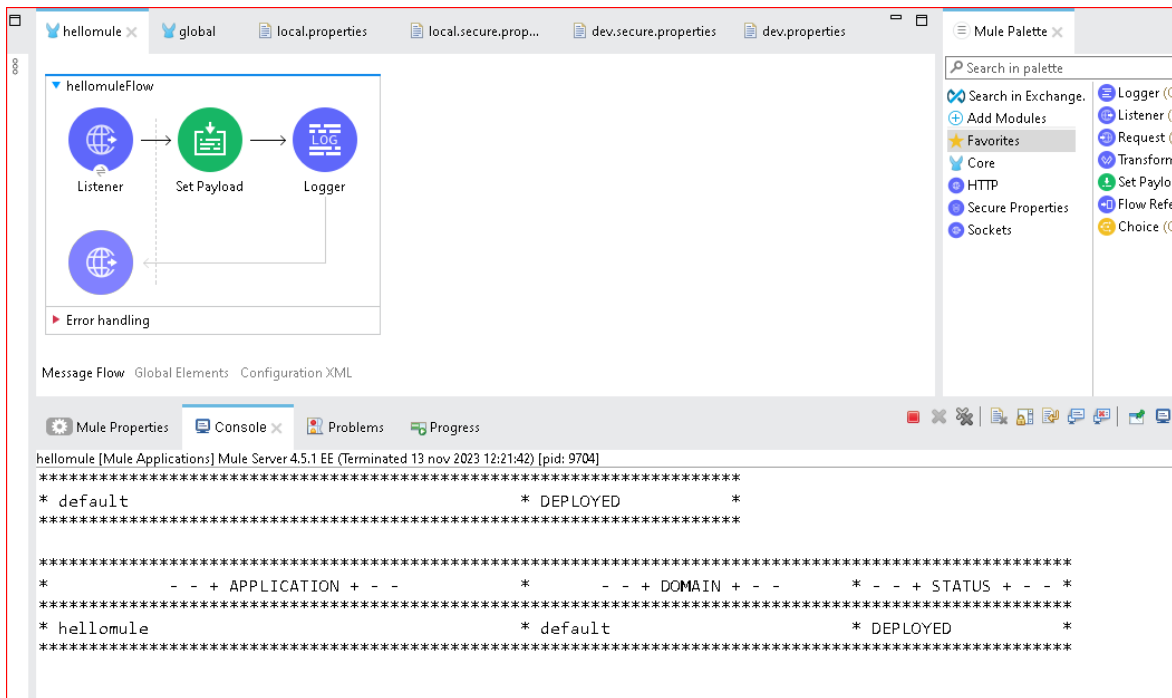
Organization Name: Prueba

Gatekeeper Settings

Configure the Gatekeeper property with a security level:

See the [Documentation](#) to configure this property

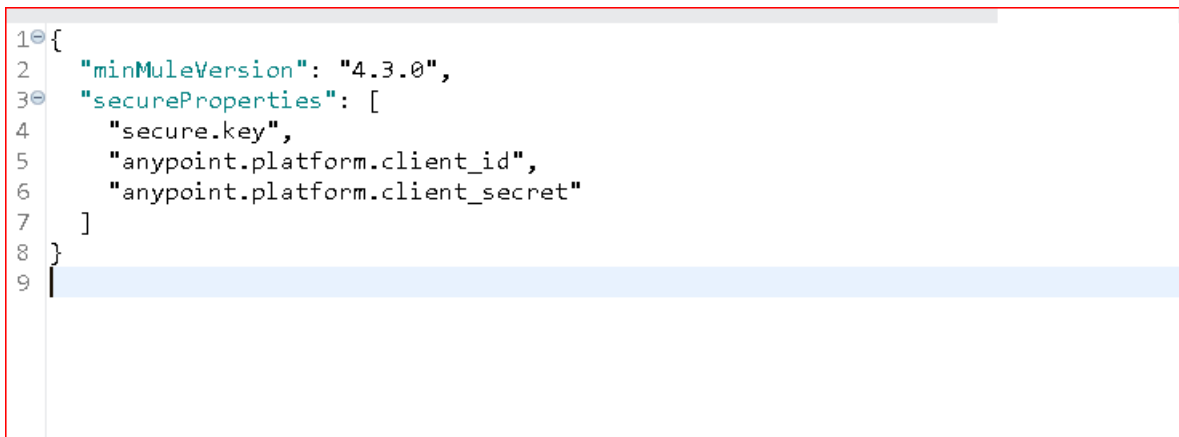
PASO 3. Corremos en la consola para verificar que todo esta correcto.



PASO 4. Nos dirigimos al PostMan y consultamos pero con el método GET.

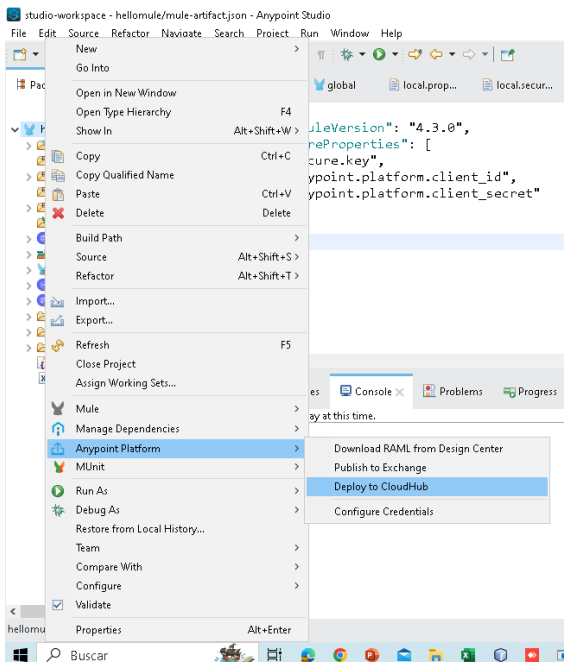
PASO 5. Nos vamos al archivo **mule-artifact.json** dentro del studio para poder agregarle este texto.

```
{
  "minMuleVersion": "4.3.0",
  "secureProperties": [
    "secure.key",
    "anypoint.platform.client_id",
    "anypoint.platform.client_secret"
  ]
}
```



```
1 {
2   "minMuleVersion": "4.3.0",
3   "secureProperties": [
4     "secure.key",
5     "anypoint.platform.client_id",
6     "anypoint.platform.client_secret"
7   ]
8 }
9
```

PASO 6. Ahora lo implementaremos en Cloudhub.



PASO 7. En la pestaña que se abra pondremos nuestro Client Id y nuestro Client Secret y le daremos en Deploy Application.

Application File

hellomule.jar

Runtime Properties Insight Logging Static IPs

Table view Text view

anypoint.platform.analytics_base_uri	https://analytics-ingest.anypoint.mulesoft.com/	✕
anypoint.platform.client_id	✕
anypoint.platform.client_secret	✕
anypoint.platform.base_uri	https://anypoint.mulesoft.com/	✕
secure.key	✕
env	dev	✕

PASO 8. Ahoa solo verificamos que la API tenga estus Activo. Para eso nos dirigimos a API Manager

API Manager

Administration (Sandbox) hellomule (v1) - API Summary

SANDBOX APIs / hellomule / API Summary

Administration

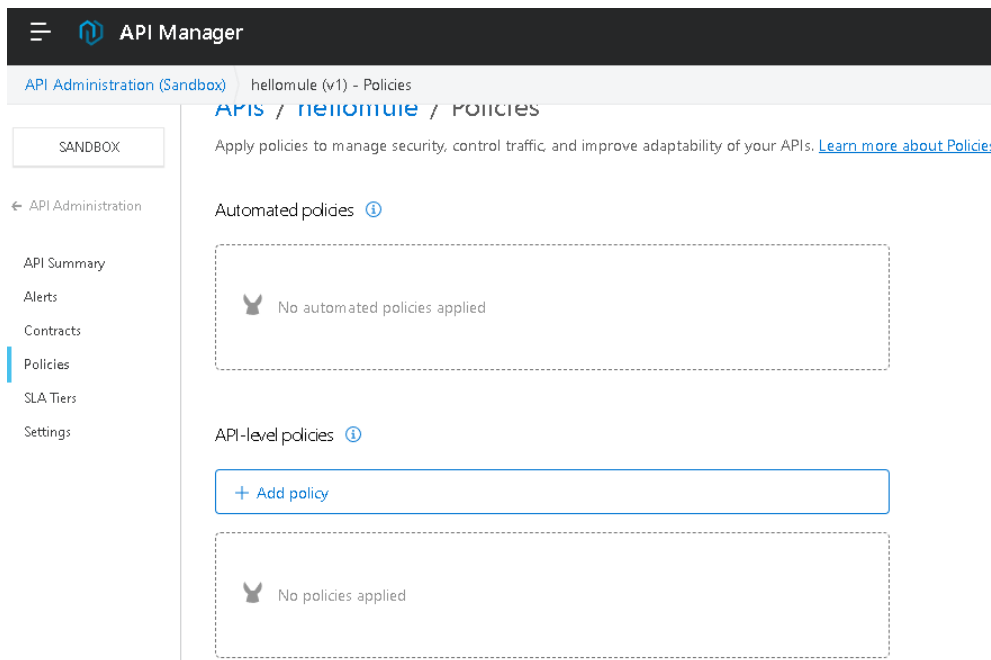
Summary

Type	Asset Version	Implementation URI ①
HTTP	1.0.0 (Latest)	N/A
API Label ①	API Version	API Status
-	v1	● Active
Consumer Endpoint	API Instance ID ①	Mule Version
N/A	19106781	4.5.1
Tags	ADD A TAG	

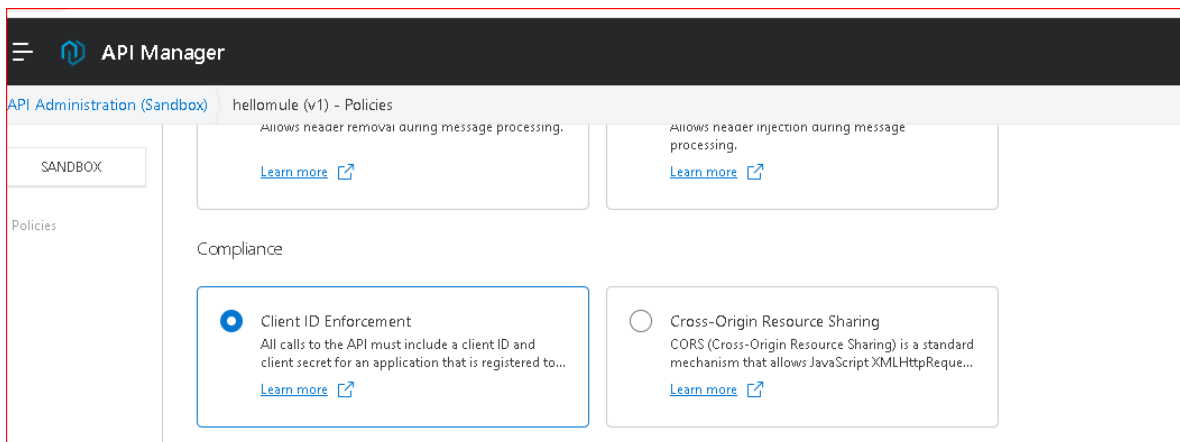
PRACTICA 5. Cómo aplicar la política de cumplimiento de ID de cliente a su aplicación Mule en API Manager

Establecer la política de cumplimiento de ID de cliente en API Manager

PASO 1. Nos dirigimos a AnyPoint Platfom , nos dirigimos a API Manager, escogemos la API con la que vamos a trabajar y seleccionamos Policies.



PASO 2. Dentro de Policies , damos en agregar una nueva , en donde escogeremos la de Id Client y le daremos en Next.



PASO 3. En la siguiente pestaña escogemos HTTP Basic y le damos Apply.

APIs / [hellomule](#) / Policies / Configure Client ID Enforcement policy

Credentials origin
Origin of the Client ID and Client Secret credentials.

☒ HTTP Basic Authentication Header


☐ Custom Expression


Advanced options >
Configure policy version, methods and resources

Previous Apply

PASO 4. Nos dirigimos a Postman donde mandaremos a llamarla con la siguiente URL.

prachellomule.us-e2.cloudhub.io/hellomule, donde no saldrá un error de este tipo. Esto quiere decir que nuestra política funciona bien.


 prachellomule.us-e2.cloudhub.io/hellomule


GET  prachellomule.us-e2.cloudhub.io/hellomule

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

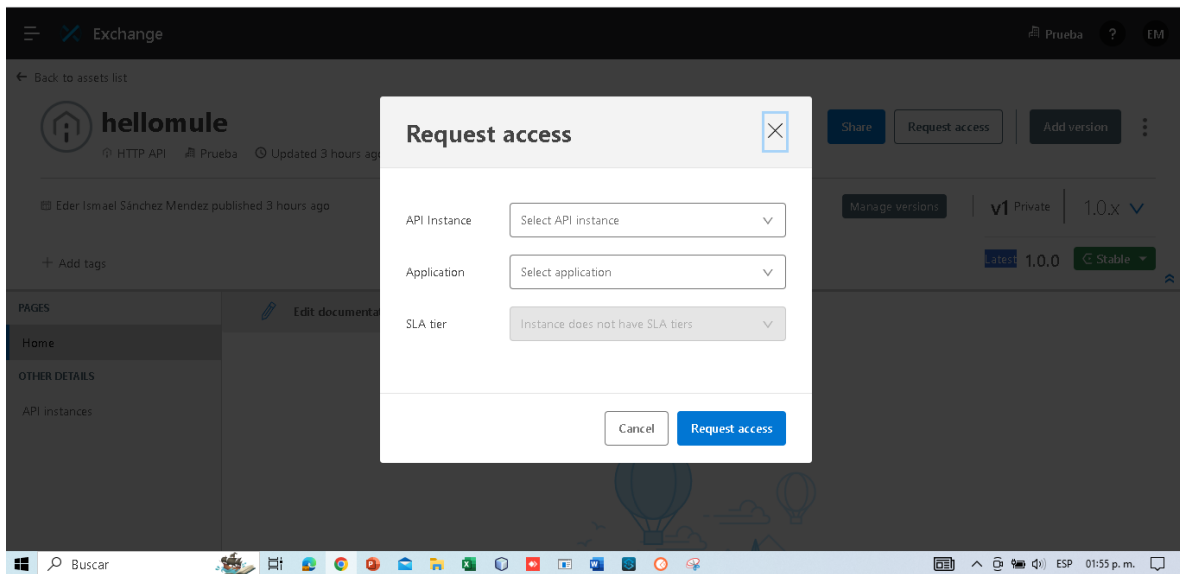
	Key	Value
	Key	Value

Body Cookies Headers (6) Test Results  Status: 401 Unauthorized

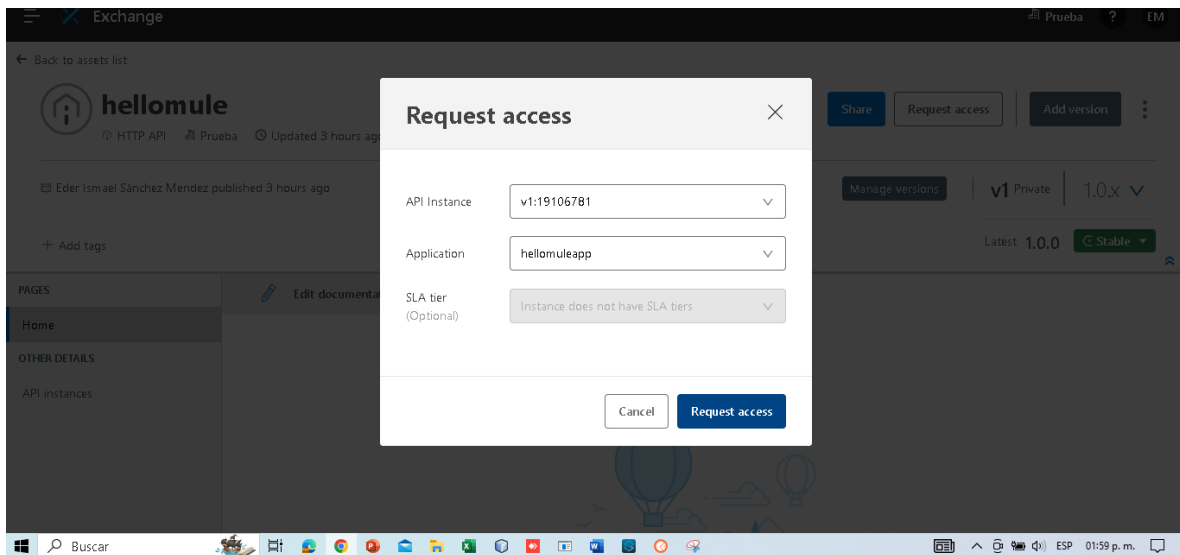
Pretty Raw Preview Visualize JSON 

```
1 {  
2   "error": "Authentication denied."  
3 }
```

PASO 5. Ahora vamos a la AnyPoint Platform , entraremos al apartado de Exchange , abrimos nuestra API que ya se encuentra creada .Le damos en el botón de Request Access.



PASO 6. En la venta que se abrió seleccionamos el primero que nos sale , donde dice application le daremos en crear una nueva.La nombramos hellomuleapp y le damos en créate.Por ultimo le damos en request Access.



PASO 7. Los accesos que nos arrojo los configuraremos en el Postman.

Request API access

Your request has been received and approved.

Use this client ID and client secret to access the requested API instance. You can always find these values at the [application page](#).

Client ID: 157f9294034446f2b3264babe33c7e6f

Client Secret: e9fBe842e39E42C2aA9440cf9a129568

PASO 8. Para eso nos vamos a Postman, damos clic donde dice Authorization, en el Type escogemos Basic Auth y ponemos los accesos.

The screenshot shows the Postman interface with a GET request to `prachellomule.us-e2.cloudhub.io/hellomule`. The **Authorization** tab is selected, and the **Type** is set to **Basic Auth**. The **Username** field contains the Client ID `157f9294034446f2b3264babe33c7e6f`, and the **Password** field contains the Client Secret `e9fBe842e39E42C2aA9440cf9a129568`. The status bar at the bottom indicates a **401 Unauthorized** response.

