

Entregable 1 – Documentación de Tutoriales

Video 1 – Crea tu primera aplicación Hello Mule

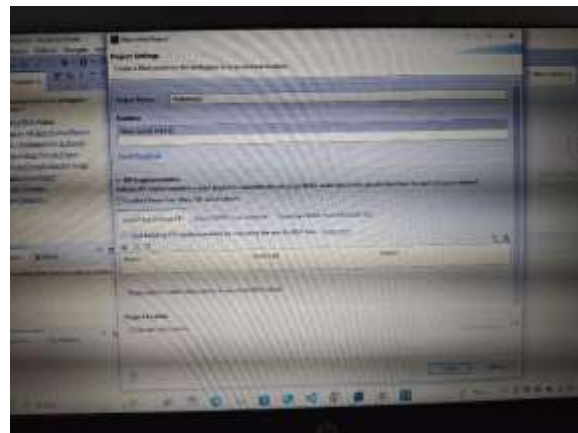
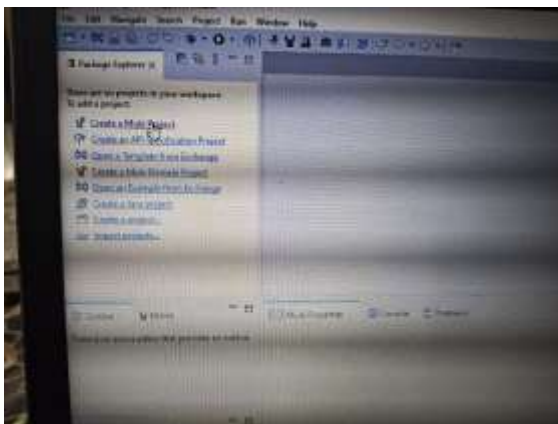
Primero Crea una cuenta en “Anypoint Platform”

Luego ingresa a “Anypoint Studio” para descargar Anypoint Studio (El IDE de MuleSoft). Para poder empezar a crear API.

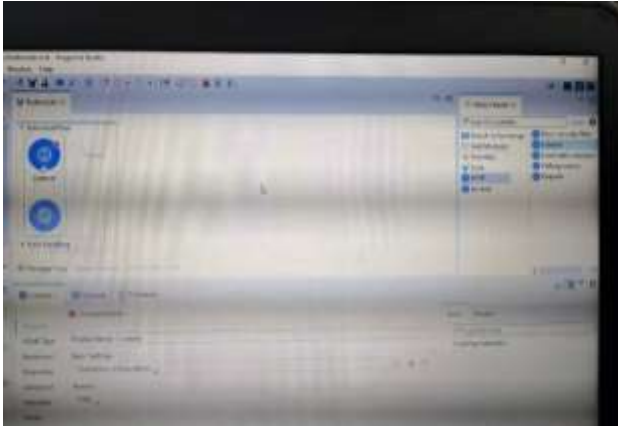
Para poder descargar Anypoint Studio te pedirá crear una cuenta para empezar tu periodo gratuito:



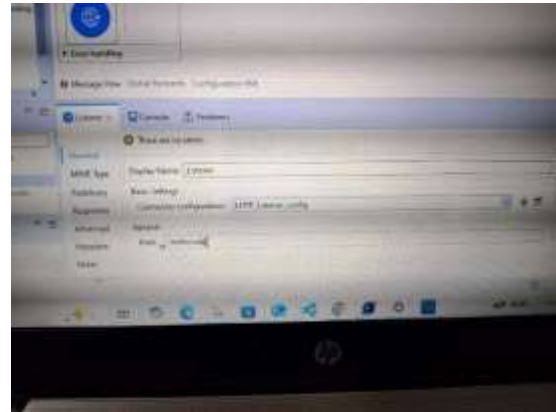
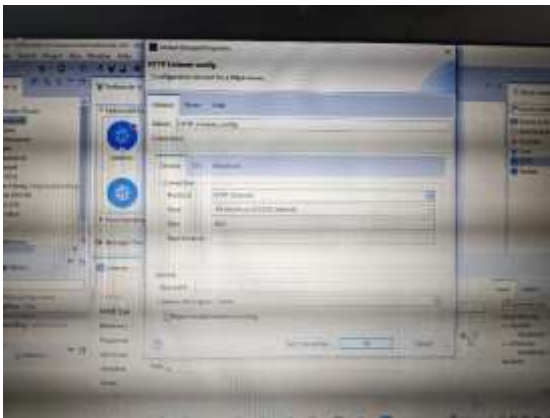
Una vez instalado y ejecutado el IDE de MuleSoft te aparecerá la siguiente ventana y da click en “Create a Mule Project” y dale el nombre de “Hello Mule”.



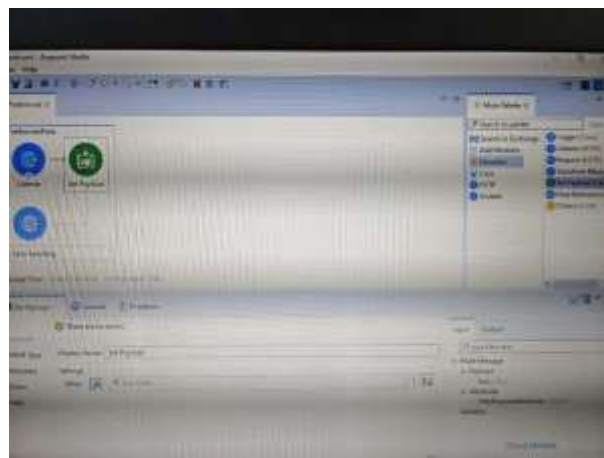
Una vez creado el proyecto, ve al lado derecho de tu pantalla en la sección de “Mule Palette” y selecciona “HTTP” → “Listener” y arrástralo hacia el espacio en blanco de la aplicación para que quede de la siguiente forma:



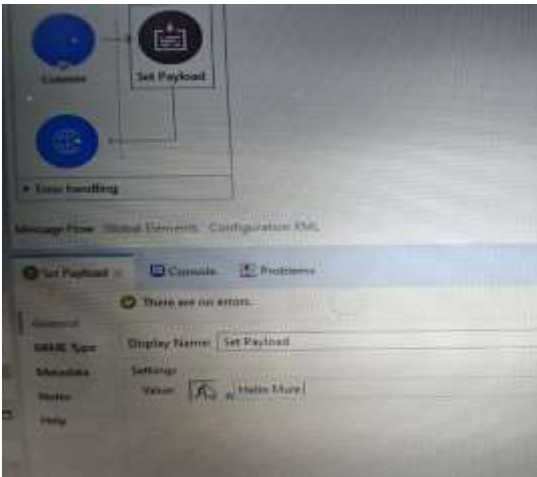
Y si das click dos veces en “Listener” podrás ver la sección de propiedades y en ella podrás ver “Connector configuration” puedes pulsar en el signo de “+” para ver “HTTP Listener config” (de momento no modificar datos) y da click en “OK” para cerrar la ventana. En la sección de “General” de Listener más abajo podrás observar “Path” ahí vas a escribir “/hellomule” y luego presiona sobre el espacio en blanco del proyecto “Hello Mule” y podrás observar que los errores desaparecieron.



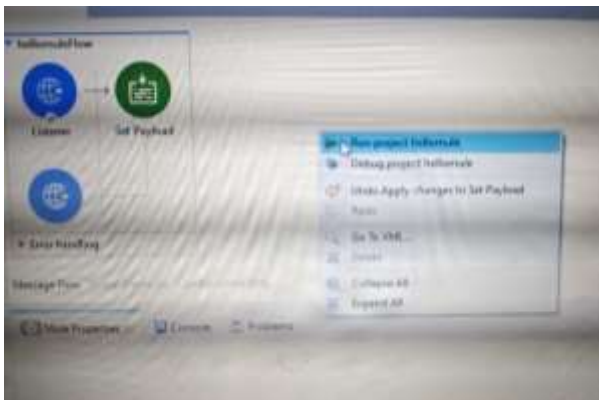
Finalizando esa parte, ahora regresa de nuevo a “Mule Palette” en la sección de “Favorites” → “Set Payload(Core)”, selecciona y arrastra al espacio en blanco al lado de “Listener” como aparece en la imagen:



Al dar click en “Set Payload” te aparecerá en la parte inferior la ventana de “General”, luego pulsa en el símbolo “fx” y escribe “Hello Mule” como aparece en la imagen (lado izquierdo) y luego pulsa nuevamente sobre el espacio en blanco para cerrar esa pestaña para proceder a guardar los cambios, para ello presiona “ctrl + S” o da click en “Save All” (como en la imagen del lado izquierdo).

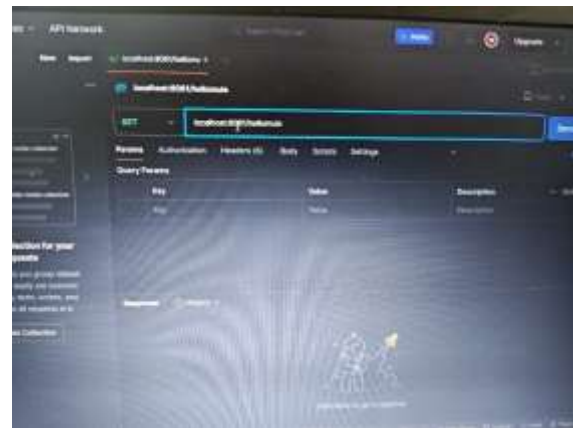
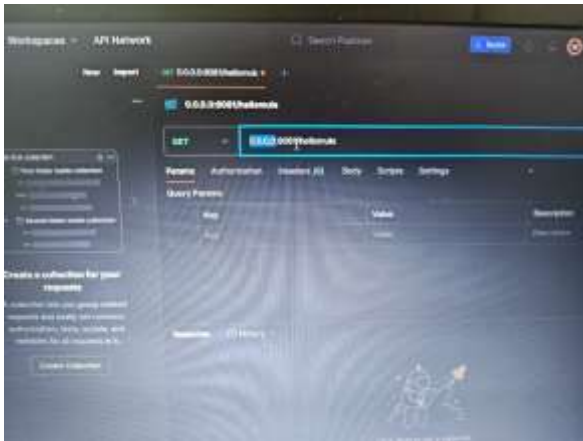


Después de guardar los cambios del proyecto pulsa el click derecho sobre el espacio en blanco y te aparecerá un pequeño menú, ahí selecciona “Run project hellomule” y espera a que cargué la en la consola la aplicación y si te aparece “Deployed” significa que ejecutó la aplicación sin errores.

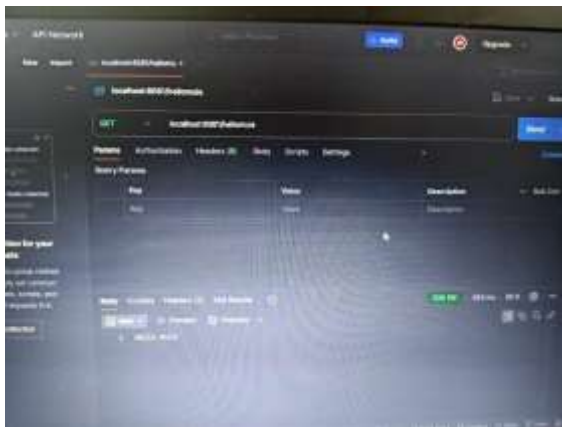


Al cargar la aplicación, se procederá a probar la aplicación, para ello puedes descargar “Advanced REST Client”, pero en este caso vamos a ocupar la aplicación “Postman”.

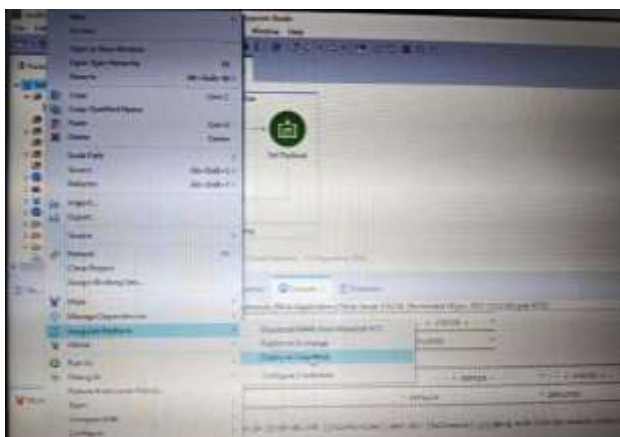
Una vez ingresando en Postman puedes click en el botón de “+” que aparece al lado de “Import” y en la sección de “GET” coloca la siguiente URL “0.0.0.0:8081/hellomule” o también puedes cambiar “0.0.0.0” por “localhost” como en las imágenes de abajo:



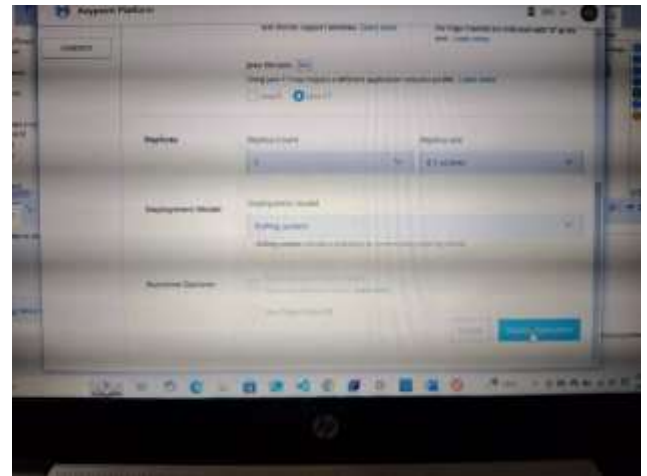
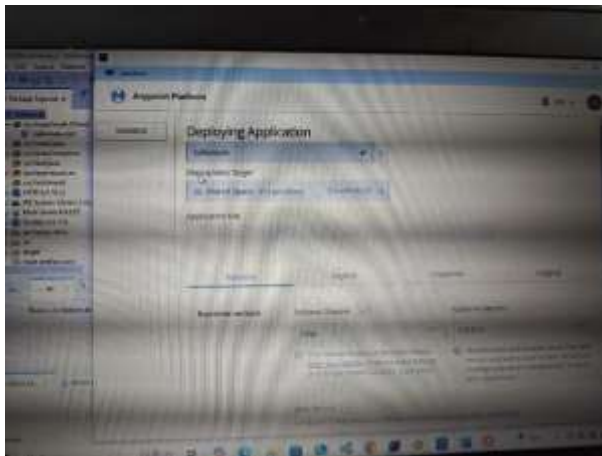
Y luego da click en el botón de “Send”, deberá aparecer en “Body” el mensaje “Hello Mule” y con el estatus “200 ok”.



Ahora vamos a hacer el “Deploy to CloudHub” para ello haz click derecho en “hellomule” y ve a la sección de “Anypoint Platform” luego en “Deploy to CloudHub” al seleccionar ahí , te va a dirigir a iniciar sesión en tu cuenta de “MuleSoft”



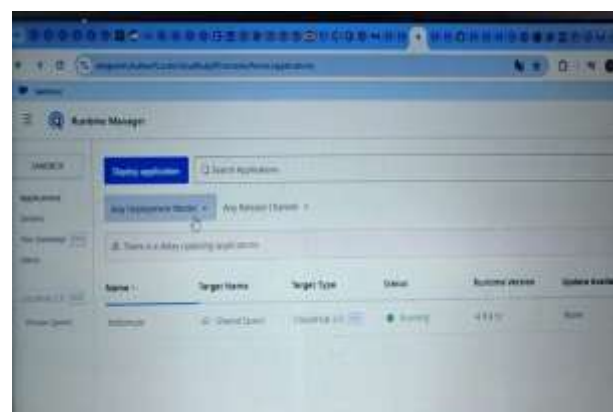
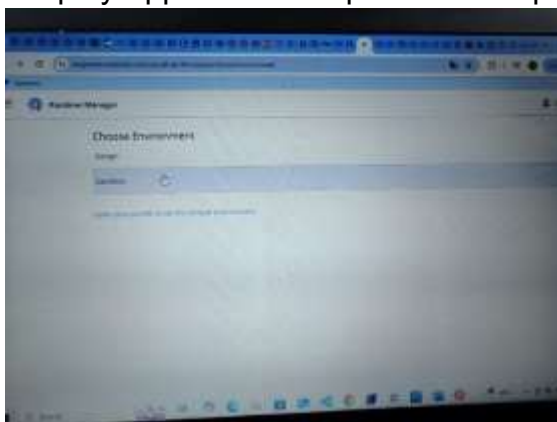
Al iniciar sesión te aparecerá la siguiente pestaña y da click en el botón de “Deploy Application”



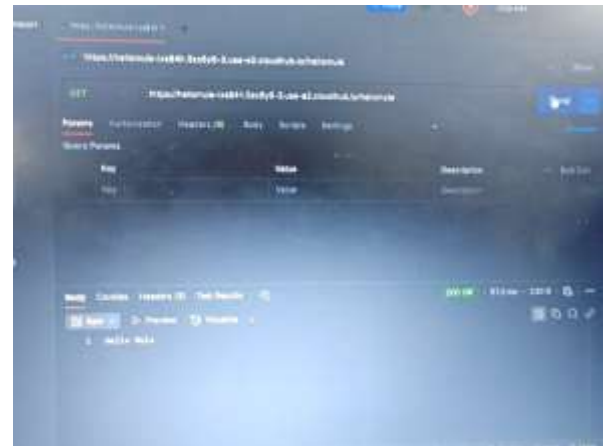
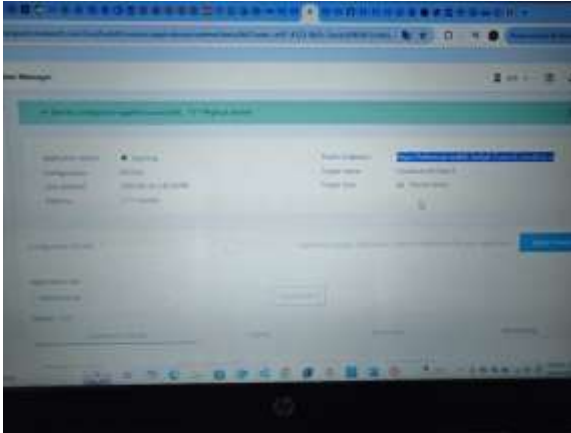
Mientras se realiza el deploy en CloudHub te aparecerá la siguiente pestaña mientras cargar, puedes pulsar en “Close Windows” si lo deseas y en la consola de tu proyecto en el IDE aparecerá, cuando finalice el deploy, “Build Success”, como aparece en las imágenes:



Luego ingresa a anypoint.mulesoft.com/cloudhub y te aparecerá la pestaña de “Runtime Manager” y selecciona “Sandbox”, al acceder te aparecerá la pestaña de “Deploy Application” te aparecerá la aplicación “hellomule”.



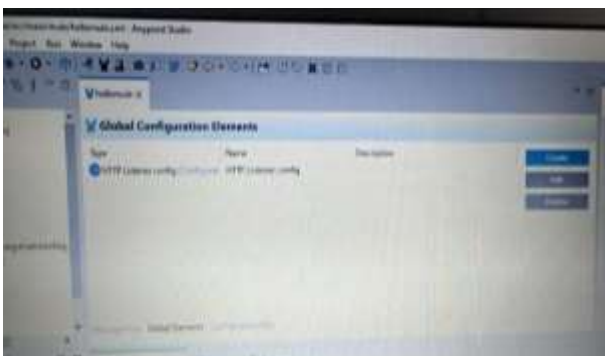
Si presionas sobre “hellomule” podrás ver el apartado “Public Endpoint” copia la dirección que te aparece ahí y pegala en Postman en la sección de url , y reemplazala con la que estaba escrita inicialmente hasta la parte de “/hellomule” quedará igual que en la siguiente imagen, y si le das click en “Send” también deberá ejecutar un “200 OK” y aparecerá el mensaje “Hello Mule”.



Video 2- Como configurar sus archivos de elementos y propiedades globales en Anypoint Studio.

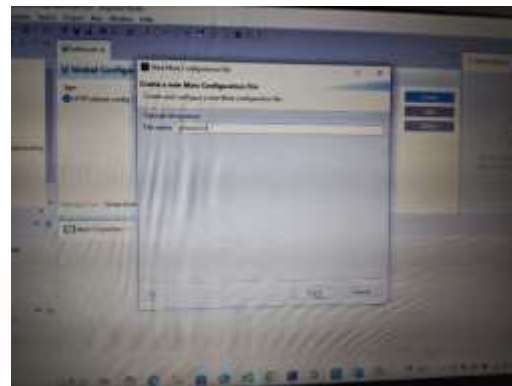
Ahora vamos a crear un archivo “global.xml” para mantener nuestros elementos globales. ¿Cuáles son los elementos globales?

Los puedes localizar dando click en “Global Elements” tal como aparece en las imágenes siguientes:

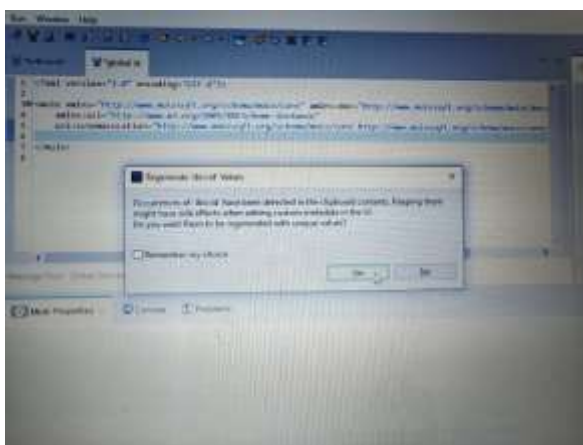


Podrás observar en la parte de “Configuración XML” el código xml que la aplicación está generando, pero cuando empezamos a crear más proyectos que tienen más archivos xml, vamos a terminar con una gran cantidad de elementos globales y es difícil realizar un seguimiento de ellos si tienen diferentes archivos xml, por ello, vamos a crear un archivo “global.xml” para mantener todas nuestras configuraciones globales.

Para ello, primero hay que crear el archivo “global.xml”, para hacerlo ve a la carpeta raíz del proyecto “hellomule” →”New “ → “Mule Configuratio File”. Y en la pantalla que te abra, dale como titulo del archivo “global.xml”



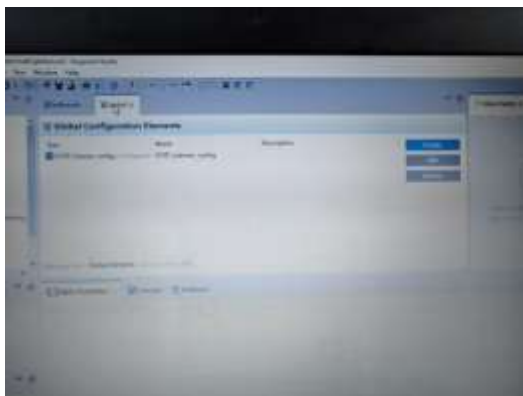
Ahora, dentro de “hellomule” en la parte de “Configuration XML”, dentro del código xml, copia y corta en esa sección la parte de “<http:listener-config>, y pegaló dentro del código “Configuration XML” en la parte de “global”,dentro de la sección de “<mule>”, tal como aparece en las imágenes. Al intentar pegar el código en “global” te aparecerá un mensaje de “Regenerated Values”, da click en “Yes”.



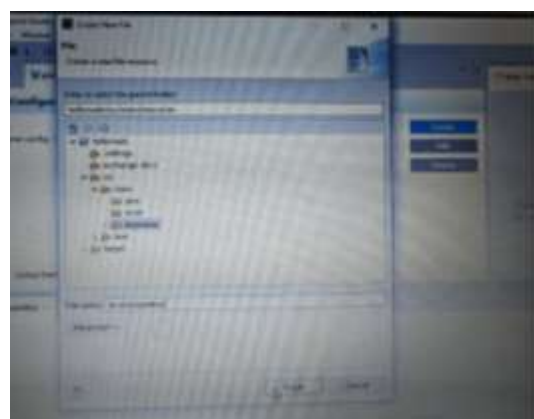
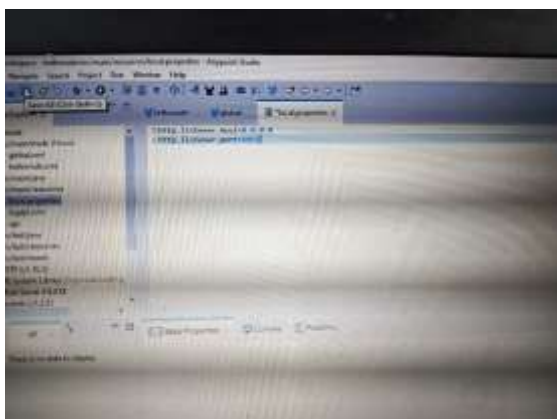
Ahora para que desaparezca el error de “nombre de archivo duplicado”, solamente da click en “Save All” para guardar los cambios y se actualice y podrás notar que el error desaparece



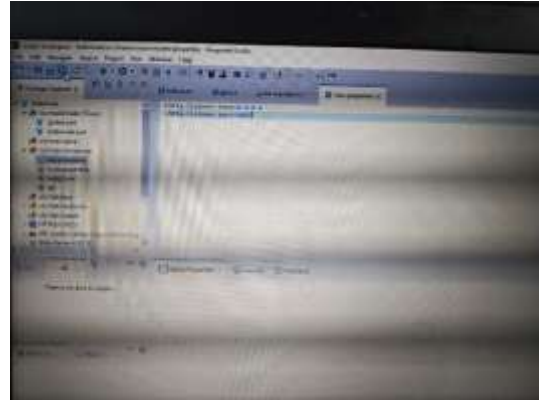
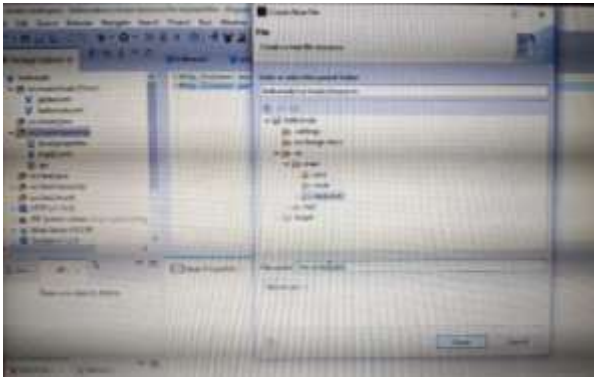
Ahora el archivo global aparecerá en nuestro archivo de “global” y ya no en “hellomule” y si queremos agregar más archivos globales, solo hay que dar click en “Create”.



Ahora, vamos a crear un archivo nuevo llamado “local.properties”, en la carpeta de “src/main/resources”. Y una vez creado el archivo, vamos a agregar las propiedades que queremos, en este caso, serán las siguientes que aparecen en la imagen como host y port y después de escribirlas, solo resta dar botón en “Save All” para guardar los cambios.



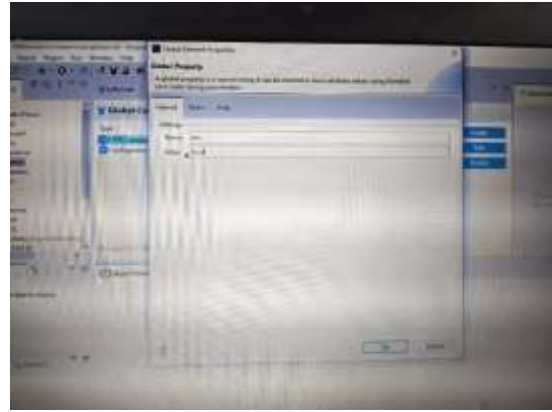
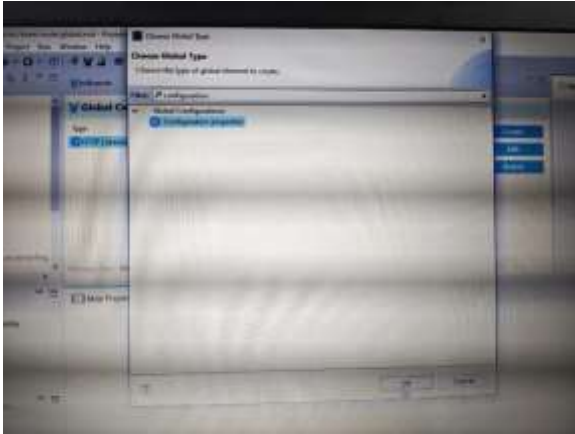
Con eso, creamos un archivo “local.properties” que se usará para nuestro entorno local pero también queremos tener uno para nuestro entorno de desarrollo que está en cloudfoundry, así que debemos, de igual forma en “src/main/resources” crear un archivo “file” nuevo “dev.properties” y vamos a agregar los mismos valores que en “local.properties”.



Ahora, vamos a reemplazar estas dos propiedades en nuestro elemento global, por lo que, primero tenemos que copiar el host y el port. Para ello da doble click en “HTTP Listener config” y luego en la ventana que abra, da doble click sobre la sección de “Host” y escribe “\${http.listener.host}”, y aplicar lo mismo con “Port” y escribir “\${http.listener.port}” y al concluir dar click en “OK”.

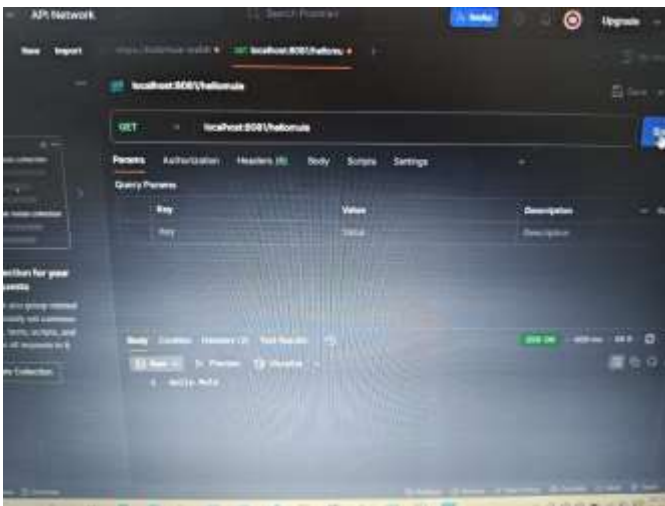


Ahora vamos a dar click en “Create” y escribe “Configuration properties” y da el nombre de “\${env}.properties”, diminutivo de “environment” ya que ahí vendrá incluido las propiedades local y global. Luego, vamos a crear otra properties, vamos a buscar nuevamente dando click en “Create” → “Global Property”, y vamos a agregar los valores de “Name: env” y en “Value: local”, así cada vez que estemos en una máquina local y ejecutemos la aplicación el runtime podrá saber que estamos corriendo en un entorno local. Después de click en “OK” y luego en “Save All”.



Ahora vamos a verificar que el proyecto sigue funcionando, primero da click en “hellomule” y en el espacio presiona click derecho y da click en “Run project hellomule” y una vez que indique que esta funcionando correctamente vamos de nuevo a Postman para la prueba.

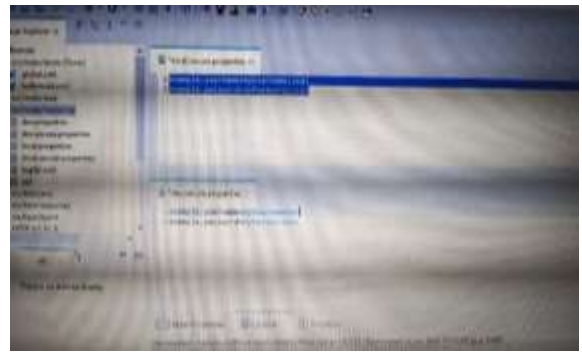
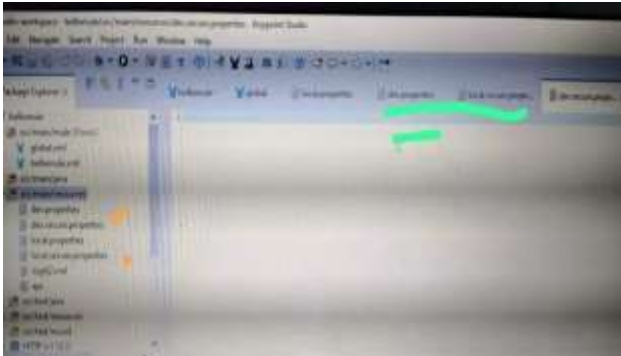
Como quedo guardado el enlace de cloud, da click en el signo de “+” para abrir una nueva pestaña y coloca el url de localhost, luego da click en “Send” y tiene que recibir un estatus “200 OK” y el mensaje “Hello Mule”.



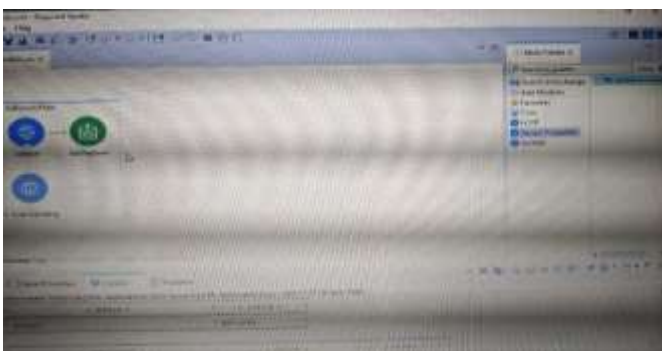
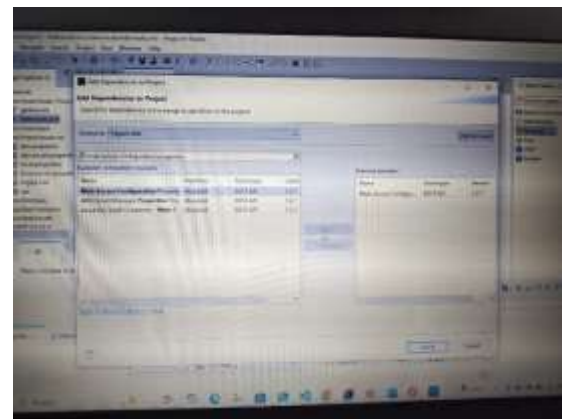
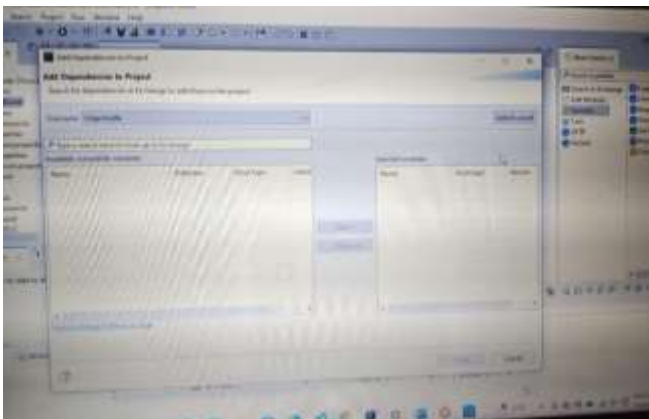
Video 3- Cómo proteger las propiedades antes de la implementación en Anypoint Studio.

Ahora vamos a crear propiedades seguras, así que para esto ve a la sección de “src/main/resources” y da click en “New” → “File” y coloca el nombre de “local.secure.properties” y después hacer lo mismo para crear el archivo “dev.secure.properties”.

Después, puedes arrastrar una de las ventanas, como en la imagen del lado derecho, para que se pueda trabajar más comodo y puedas copiar y pegar más rápido el código de “example.username”, pero ten cuidado, ya que en la ventana de “dev”, el cambia el valor de “Local” a “Dev”:



Después, puedes cerrar ambas ventanas, y ve de nuevo a “hellomule” y en la sección donde aparece “Mule Palette”, da doble click sobre “Search in Exchange” y te abrirá la siguiente pestaña: Es para buscar cualquier modulo que quieras en tu espacio de trabajo, en este caso, vamos a buscar “mule secure configuration properties” y presiona “Enter”, cuando te aparezca, seleccionalo y presiona en “Add” y luego “Finish”.



Ahora, podrás notar que no tenemos conectores u operaciones que podamos usar en este módulo, es porque de momento, no tenemos un conector para poder arrastrar y soltar en el lienzo, pero los usaremos para algunas configuraciones globales.

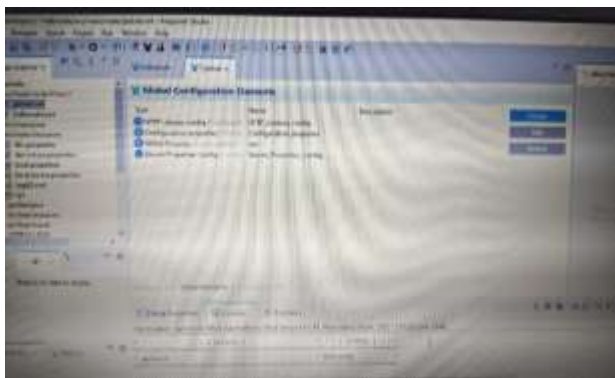
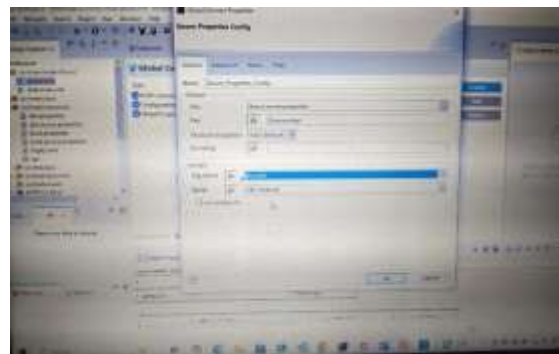
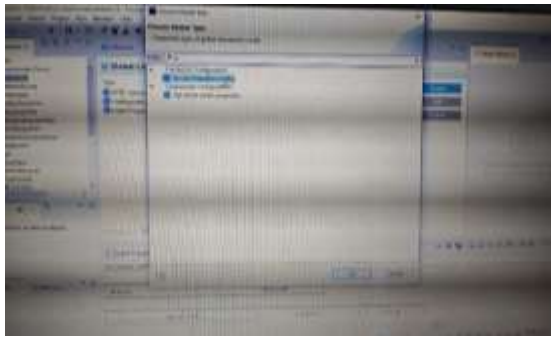
Por ahora, abre la pestaña de “global” y luego, dentro de global, cambia a la pestaña de “Global Elements”, luego pulsa en “Create” y busca “Secure Properties Config”, y cuando te aparezca, puedes pulsar en “Ok” o dando sobre click sobre ella, para que después te abra la pestaña de “Secure Properties Config” e ingresa los valores siguientes, tal cual en la imagen y luego presionar en “Ok”:

“File: \${env}.secure.properties”

“Key: \${secure.key}”

“Algorithm; Blowfish”

Luego, guarda el archivo en “Save All” y regresa al archivo “hellomule”



Ahora, vamos a leer nuestras credenciales y mostrarlas en un registrador, en este ejemplo, así que vamos a “Favorites” y vamos a seleccionar el componente “Logger” y arrastrarlo al lado de “Set Payload”, tal cual aparece en la imagen:

Luego, en esa sección, “Message”, pulsa en el botón “fx” y en esa sección, del lado derecho, da click en el botón con símbolo, al hacerlo, te abrirá la siguiente ventana, para generar el Datawave code, Datawave es un lenguaje que se utiliza

en MuleSoft para realizar transformaciones o configuraciones básicas. En este caso si queremos agregar expresiones al Logger, podemos hacerlo usando Datawave, y ahí ingresa el siguiente código:

```
"""output application/java
```

```
---
```

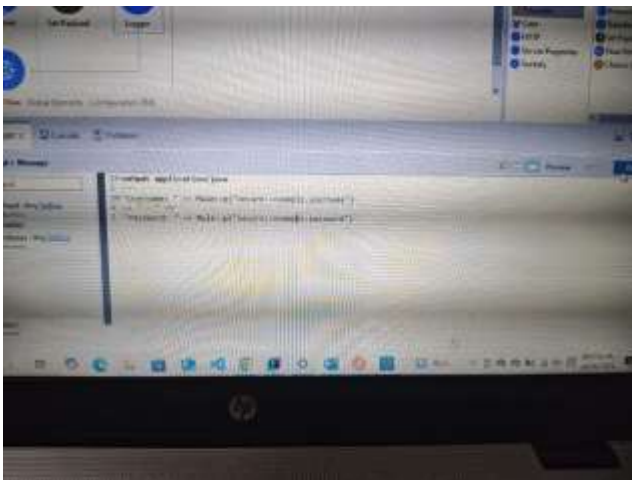
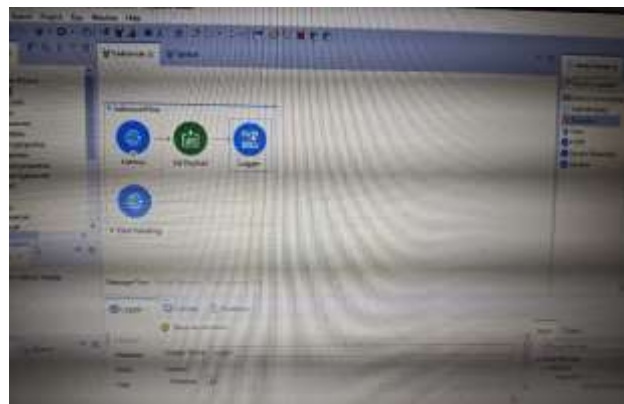
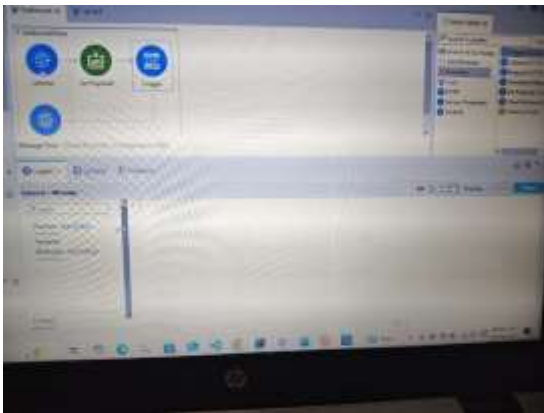
```
"Username: " ++ Mule::p("secure::example.username")
```

```
++ " - " ++
```

```
"Password: " ++ Mule::p("secure::example.password")"""
```

p() → Es para escribir la función

++ → es para concatenar



Y para cerrar esa ventana, puedes dar click en "Done" o pulsar en la parte de arriba en el espacio en blanco, y cuando se cierre la venta da click en "Save All" para guardar los cambios.

Ahora, debemos encriptar nuestras propiedades de Dev y Local, para ello, en la consola, una vez localizada la carpeta donde se descargó el archivo jar secure-properties (el cual puedes localizar en la página de MuleSoft donde viene la documentación <https://docs.mulesoft.com/mule-runtime/latest/secure-configuration-properties>)

Ingresa los siguientes código para generar la contraseña encryptada de cada propiedad creada: UsernameLocal, PasswordLocal, UsernameDev y PasswordDev.

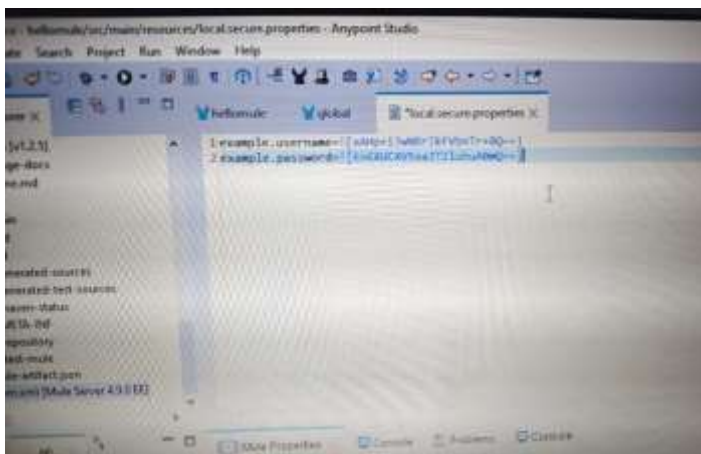
```
java -cp secure-properties-tool-j17.jar com.mulesoft.tools.SecurePropertiesTool  
string encrypt Blowfish CBC mulesoft "myUsernameLocal"
```

```
java -cp secure-properties-tool-j17.jar com.mulesoft.tools.SecurePropertiesTool  
string encrypt Blowfish CBC mulesoft "myPasswordLocal"
```

```
java -cp secure-properties-tool-j17.jar com.mulesoft.tools.SecurePropertiesTool  
string encrypt Blowfish CBC mulesoft "myUsernameDev"
```

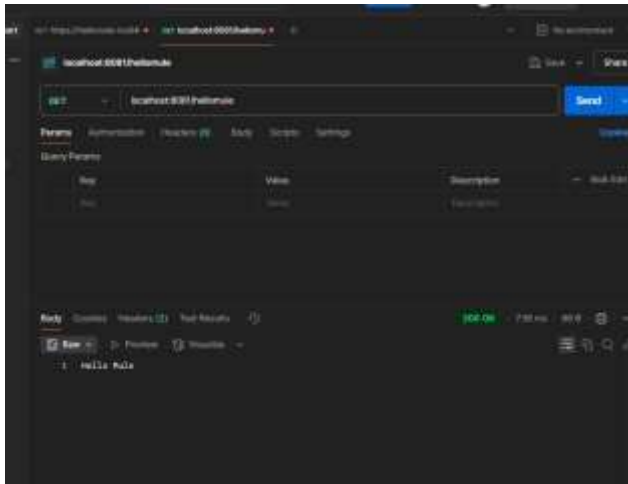
```
java -cp secure-properties-tool-j17.jar com.mulesoft.tools.SecurePropertiesTool  
string encrypt Blowfish CBC mulesoft "myPasswordDev"
```

Y agregar la contraseña encryptada que generé en la sección correspondiente escribiendola entre “{}”.

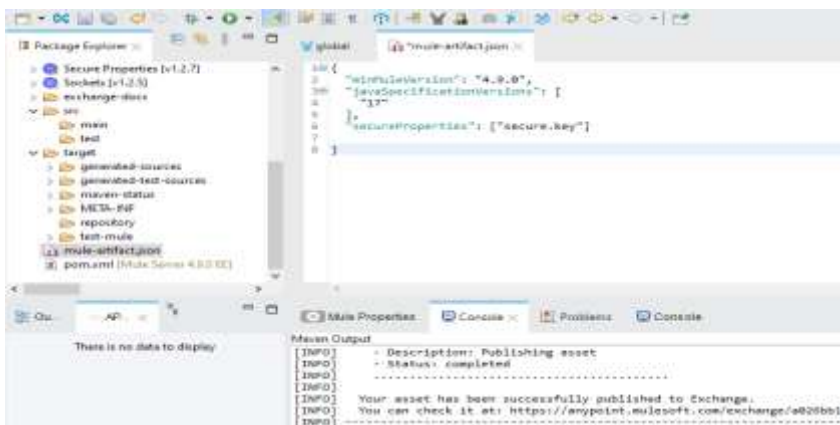


Es mejor encryptar las contraseñas, ya que luego se puede olvidar configurar las propiedades de “env” y la propiedad de secure.key no podemos hacerlo y si envías tu repositorio de github o cualquier otro lugar donde guardas todo tu código, alguien tendrá acceso a ese código, podrá ver la clave segura que estas usando y podrá descifrar todas tus propiedades. Por razones de seguridad es mejor si estaleces la secure.key, en tu “Run Configuration” en lugar en “Global Element”, tal como lo hicimos con env, properties

Luego, guarda la aplicación en “Save All” y ejecutala en “Run project hellomule” y al ejecutarse, haz nuevamente la prueba en Postman y si ejecuta “200 OK”, regresa al IDE y podrás observar en la consola que ya aparecen las credenciales.



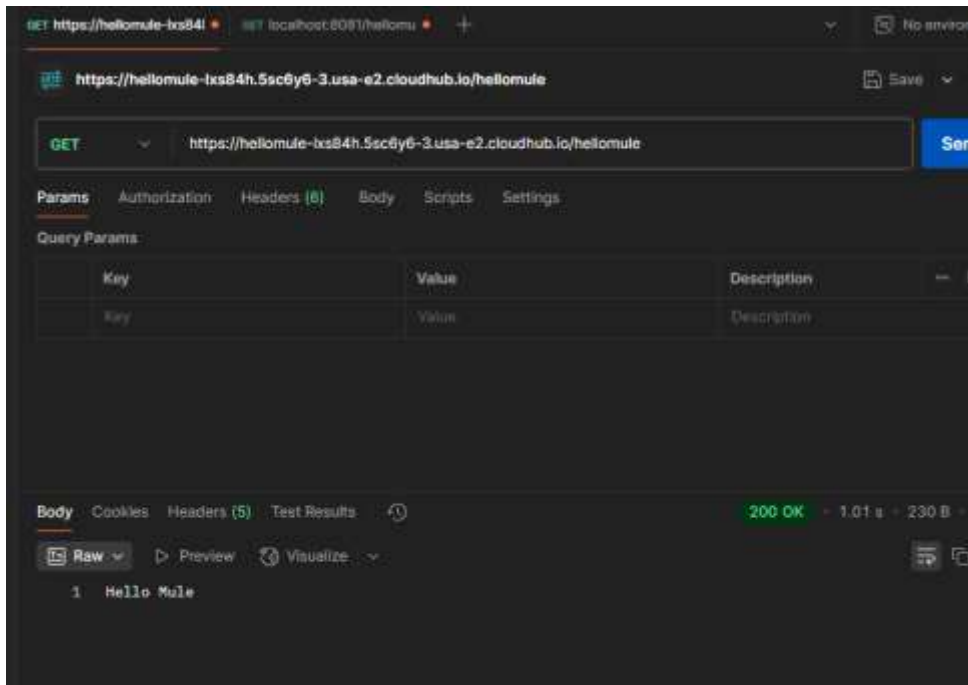
En la sección “mule-artifact.json” puedes ingresar los datos que quieres ocultar en Cloud, ahí ingresa “secure.key” tal como aparece en la imagen.



Y luego haz un deploy para Cloud y podrás observar en “Runtime Manager” que ya está oculta tu key.

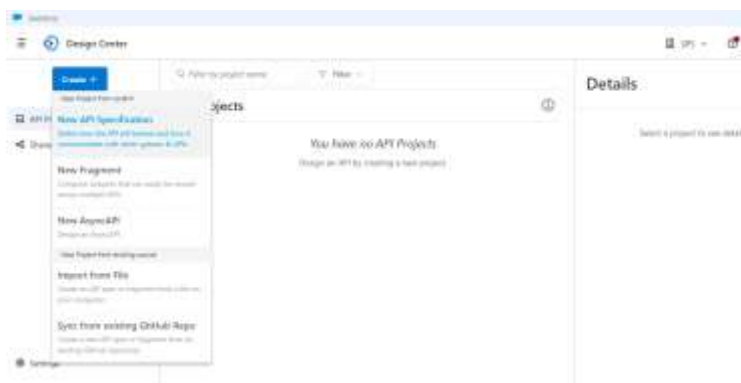
Para finalizar haz una vez más la prueba en Postman con la url generada en “Runtime Manager” y debe dar igual un “Estatus 200 OK”, y el mensaje “Hello Mule”.

Y con eso se finaliza el tutorial.

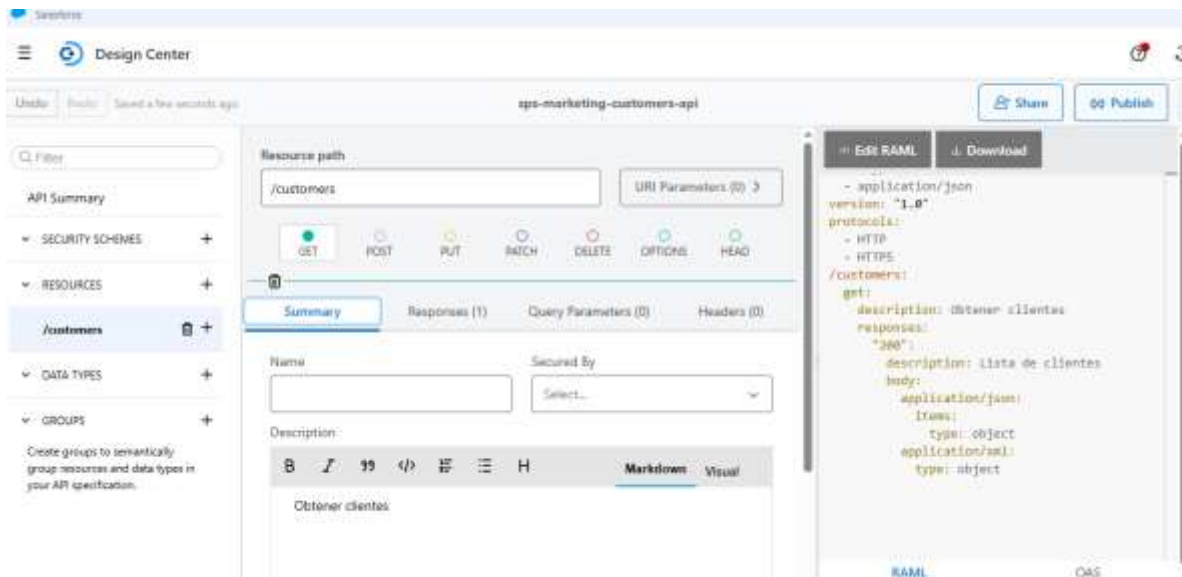


Pasos del proyecto de API

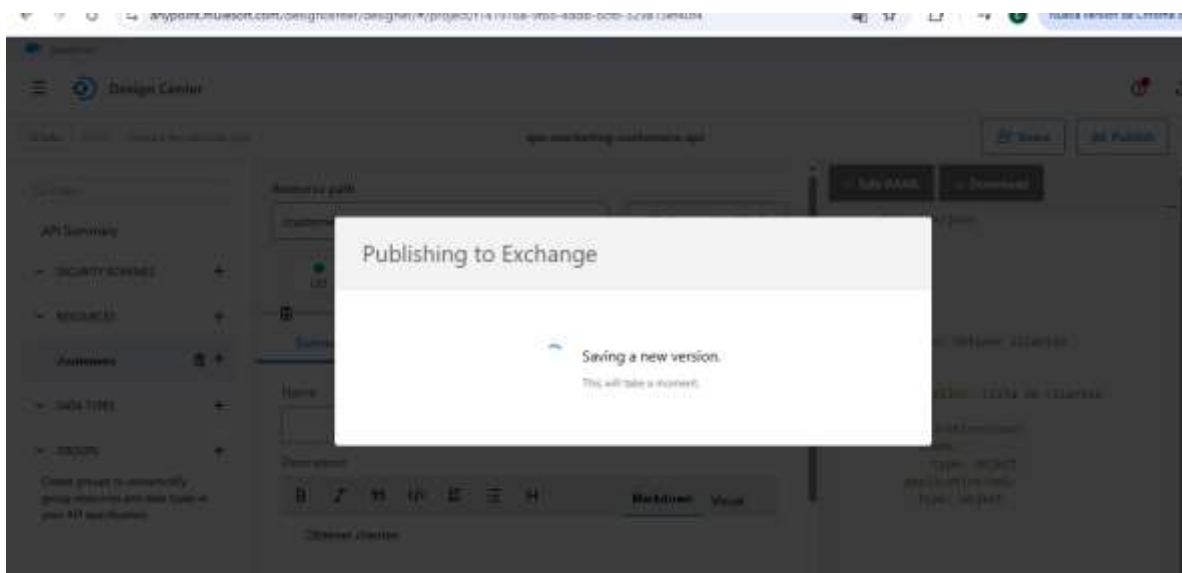
Ingresa a Design Center y selecciona “Create”



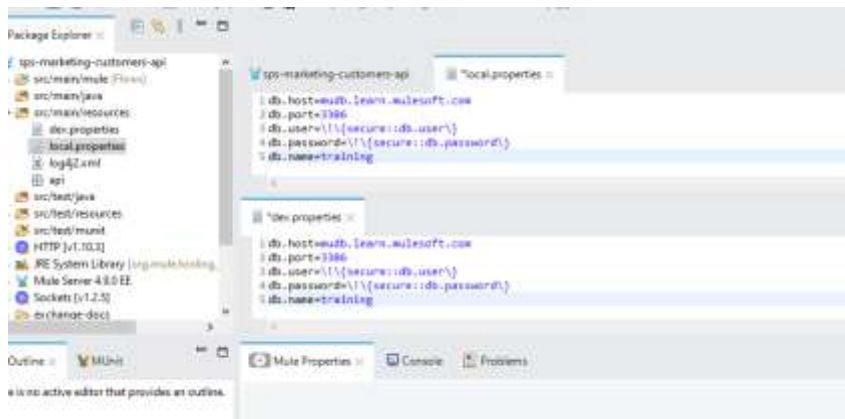
Escribe el nombre del recurso, la versión y ejemplo de uri, para después generar en “Resources” el código en RAML para las pruebas, con cuerpo JSON.



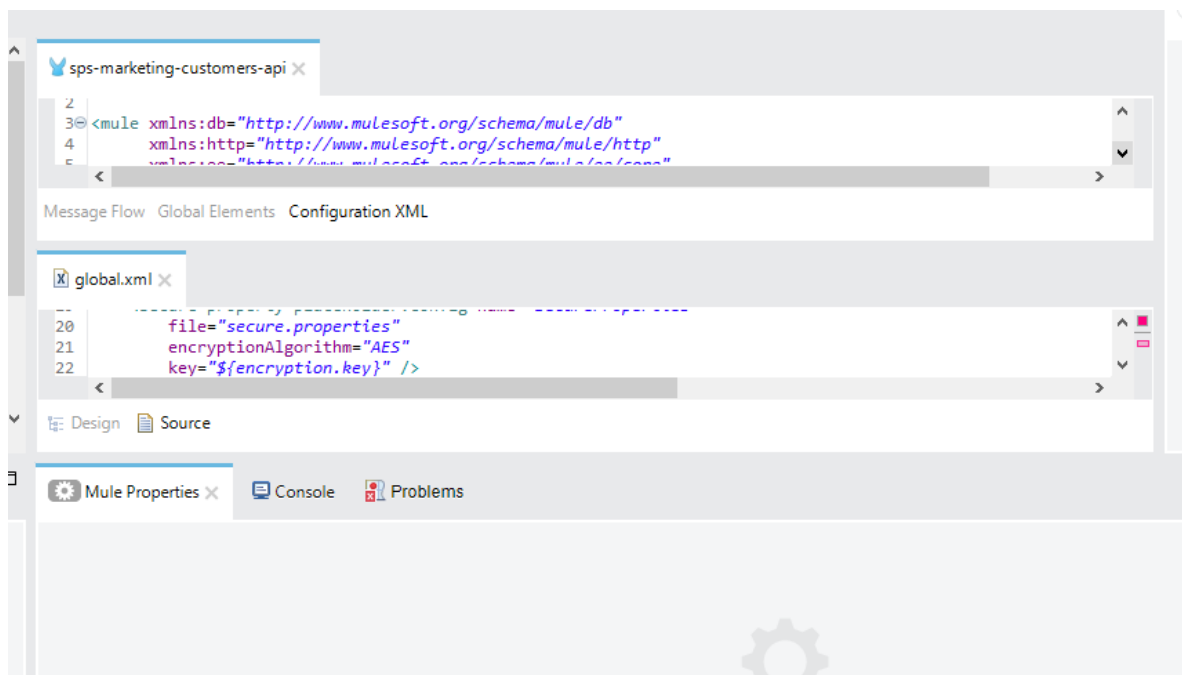
Al finaliza, publica el API, presiona en “Publish”.



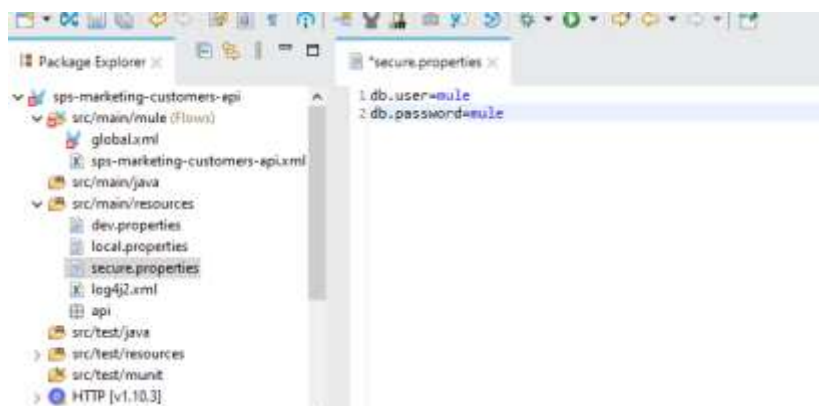
Ahora en Anypoint Studio (IDE), genera los archivos dev y local y coloca los valores para la base de datos.



Genera los archivos global y flow.



Agrega, como indica la práctica como user y password “mule” en secure.properties.



Con el siguiente comando, usalo en la consola para obtener la key encriptada.

```
java -cp secure-properties-tool-j17.jar com.mulesoft.tools.SecurePropertiesTool  
string encrypt Blowfish CBC mulesoft "myKey123"
```

Coloca la key encriptada en local o dev properties y ejecuta la aplicación.