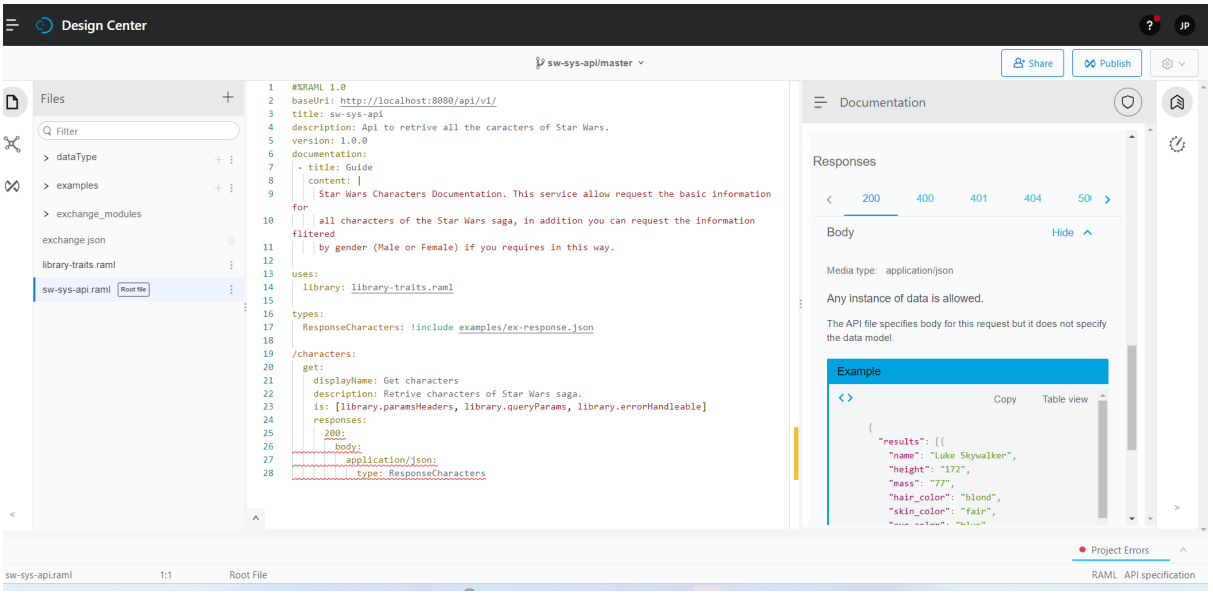
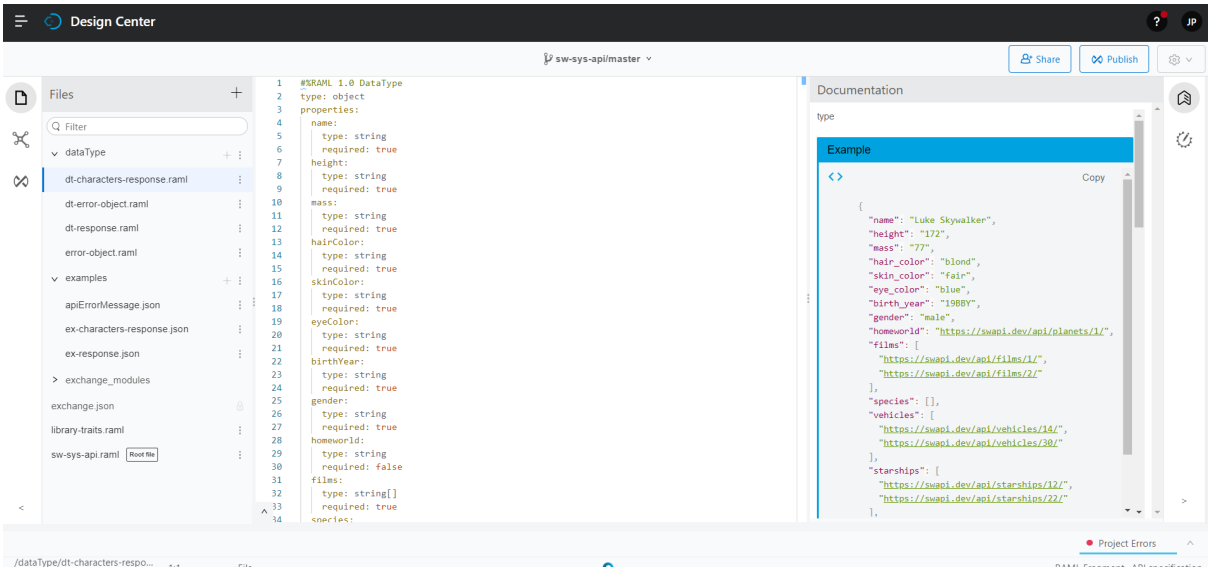
PRÁCTICA MULESOFT

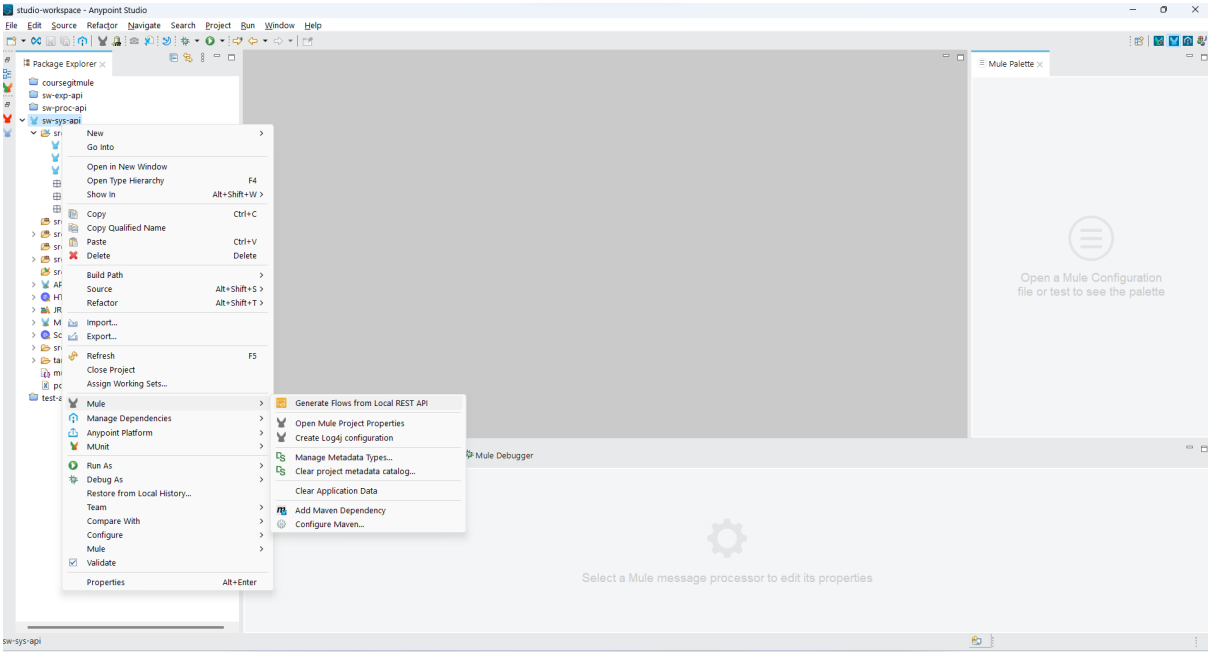
System Api.

Iniciar con los parámetros de entrada y salida con que contamos, haciendo pruebas al endpoint para validar la data y poder empezar con el Raml file, data type y ejemplos.

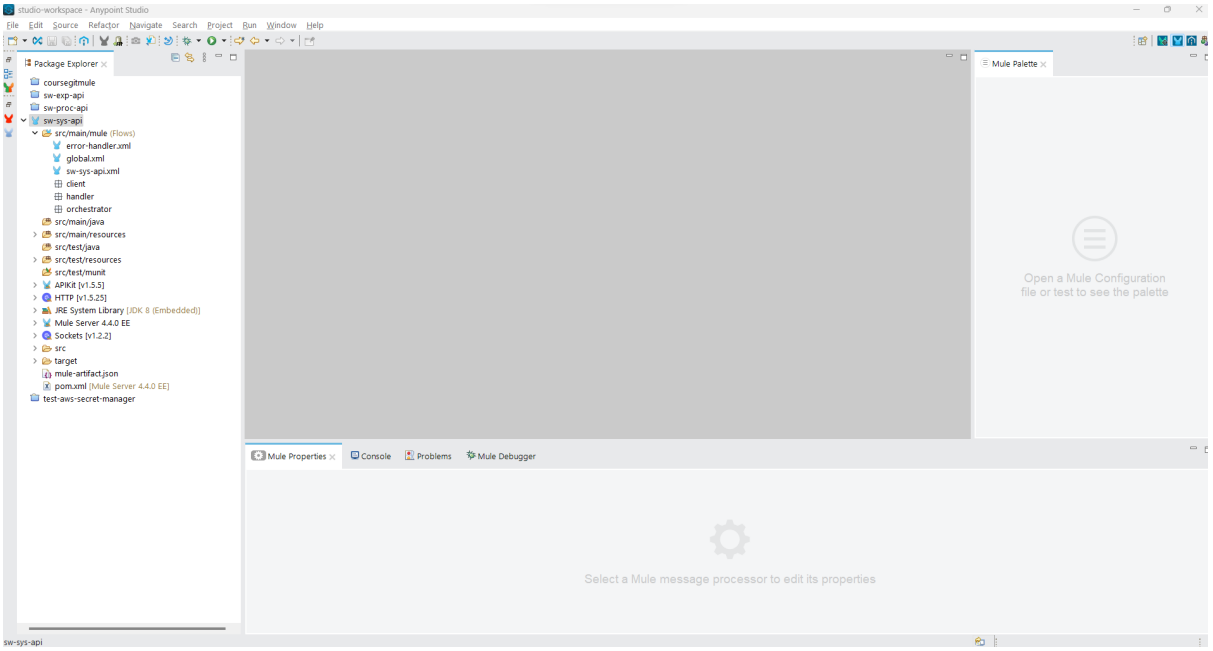




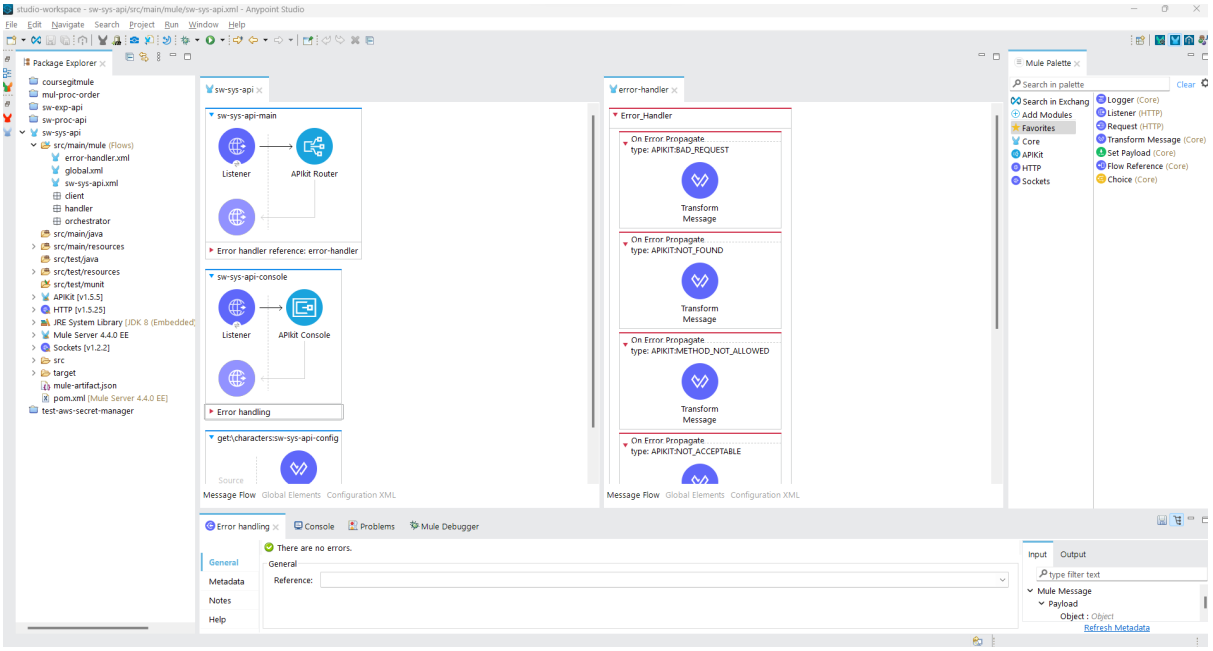
Generar los flujos iniciales a partir del Raml que contiene el proyecto en forma local.

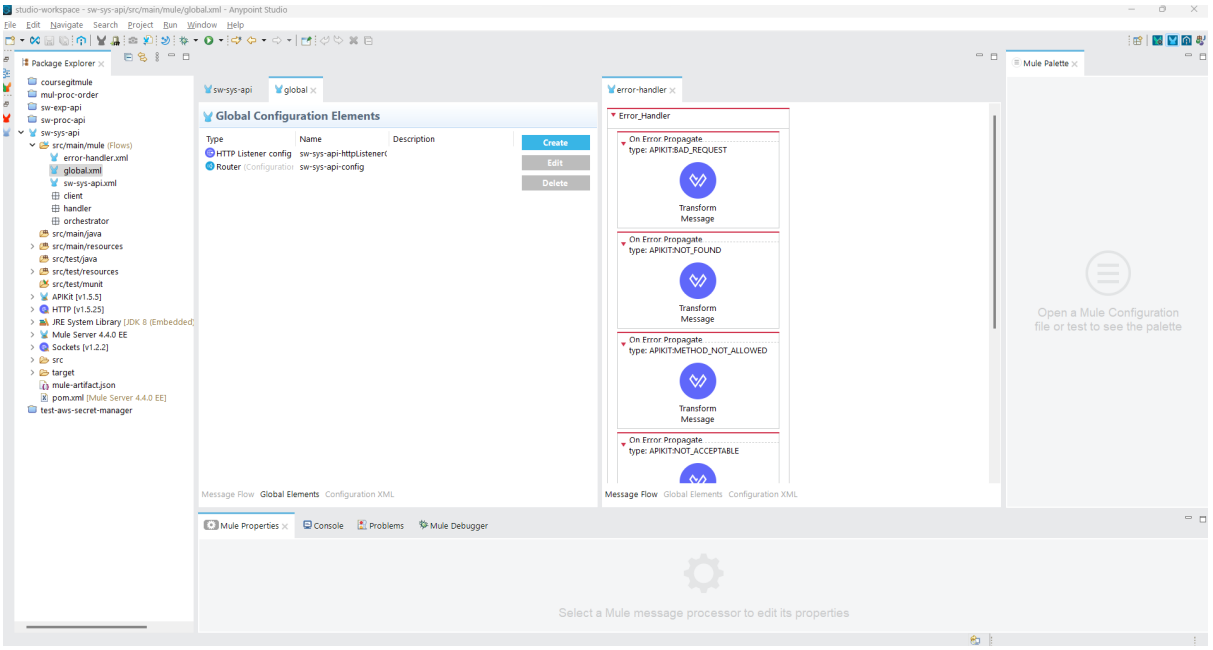


Crear la estructura base del proyecto mulesoft

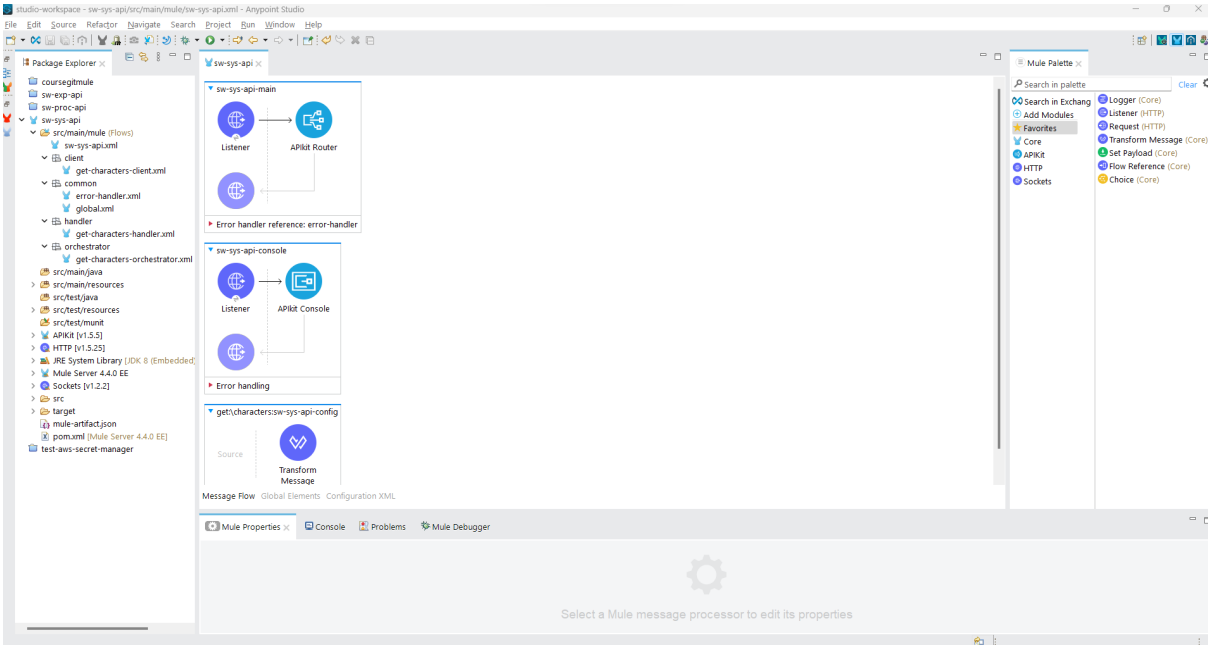


Separar el manejo de errores a un archivo .xml independiente, de igual forma la configuración del Listener y Apikit Router para una mejor organización..

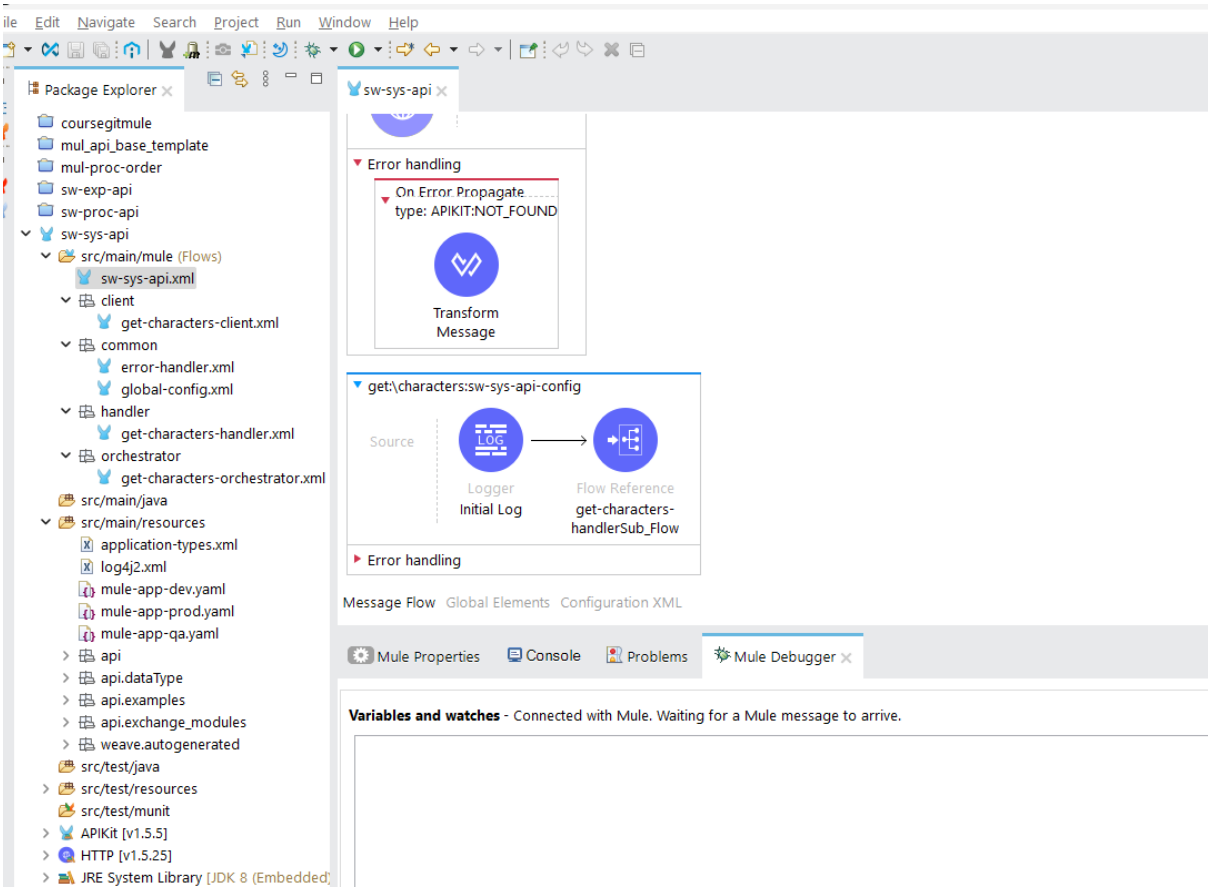


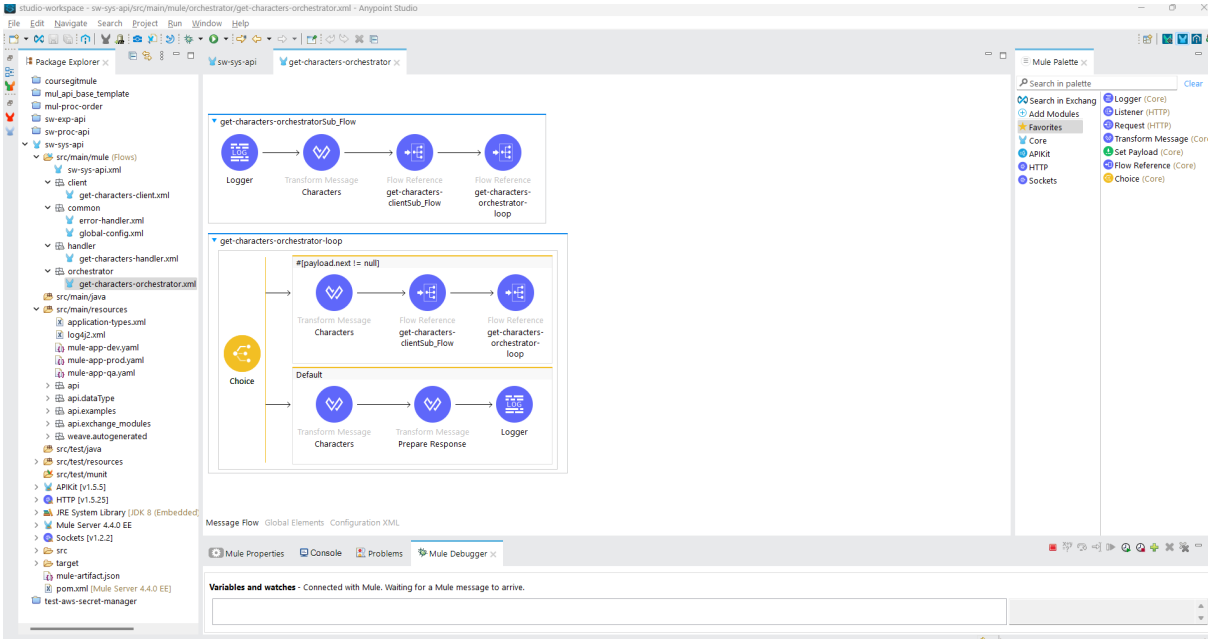


Crear archivos .xml para el handler del endpoint, un orquestador para la preparación o manipulación que la data requiera, un cliente para la llamada que se realizará, nombrandolas en base a la operación que realiza, y colocar cada uno en su respectiva carpeta.

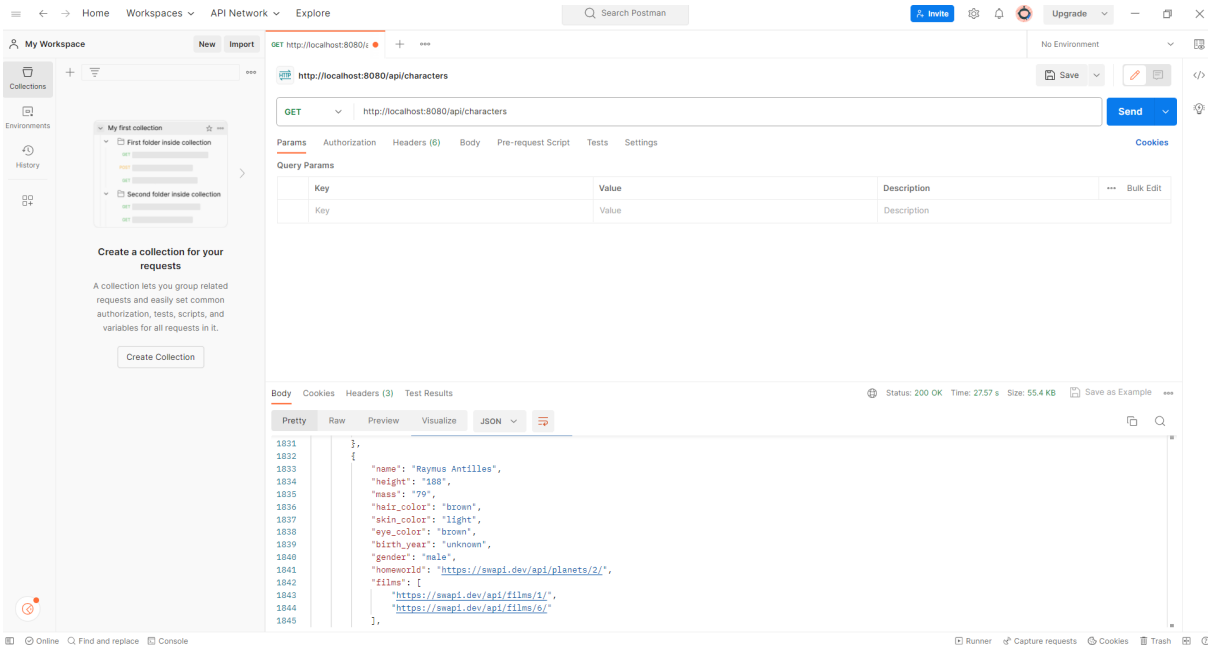


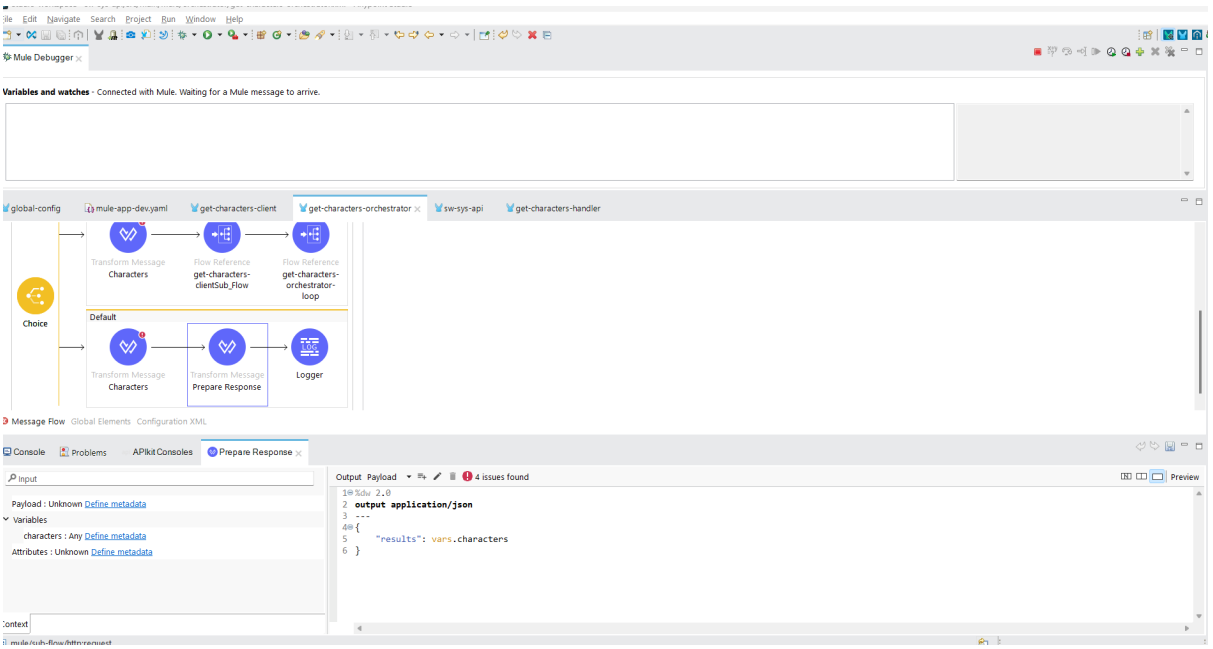
Continuamos agregando logs, flow reference y tranforms messages para consolidar el flujo final.





Pruebas del servicio para evitar situaciones no planificadas

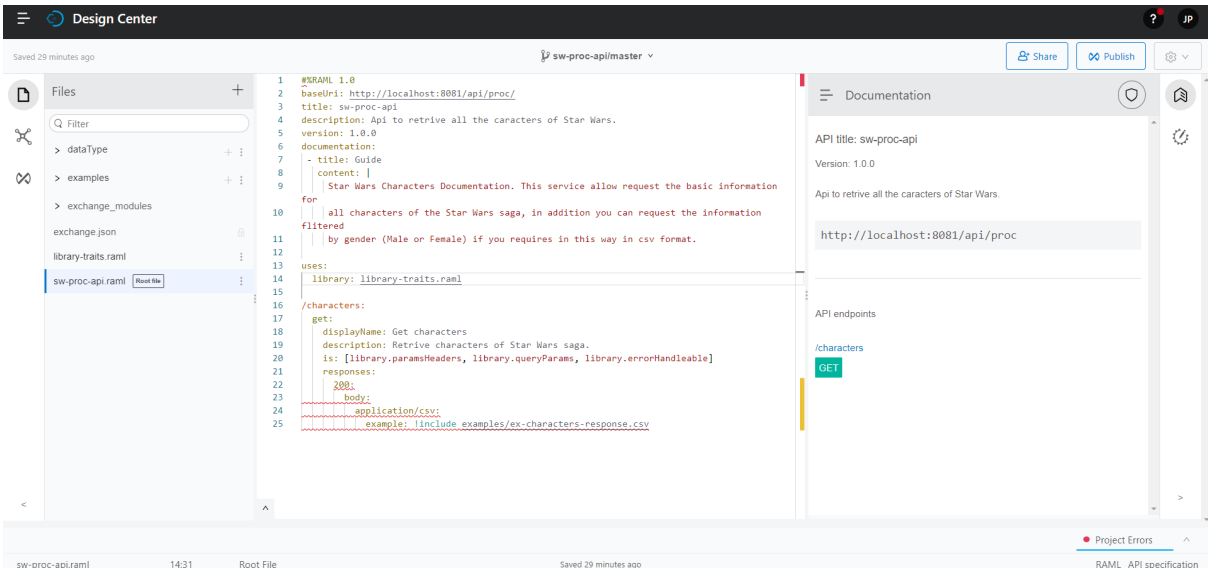


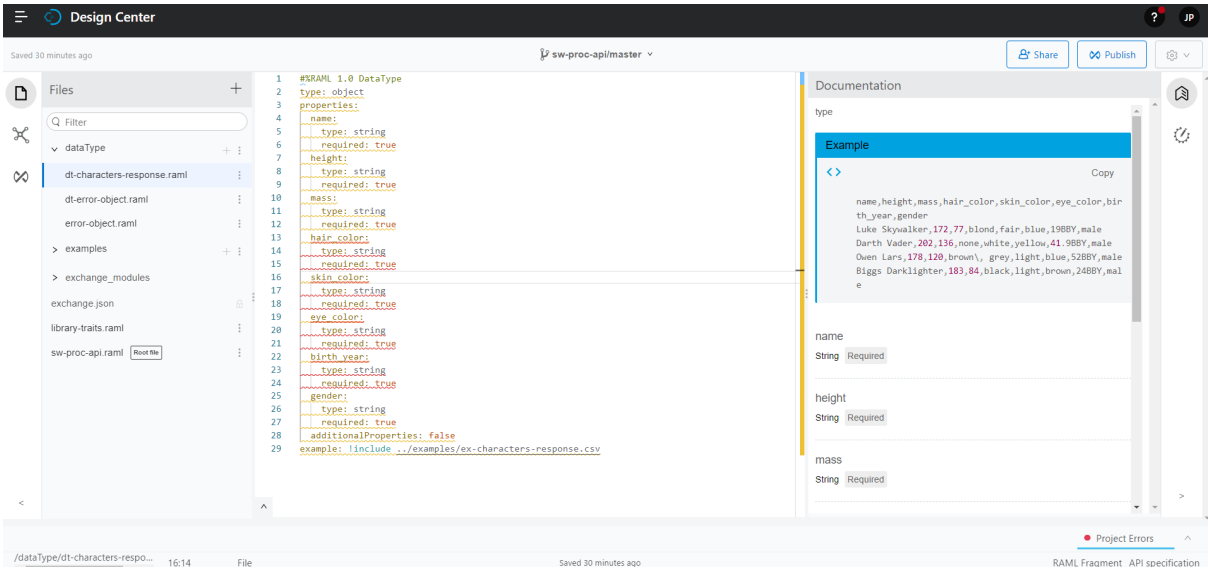

Edit Navigate Search Project Run Window Help
Mule Debugger
Variables and watches - Connected with Mule. Waiting for a Mule message to arrive.
global-config
mule-app-dev.yaml
get-characters-client
get-characters-orchestrator
sw-sys-api
get-characters-handler
Transform Message
Characters
Flow Reference
get-characters-clientSub_Flow
Flow Reference
get-characters-orchestrator-loop
Choice
Default
Transform Message
Characters
Transform Message
Prepare Response
Logger
Message Flow
Global Elements
Configuration XML
Console
Problems
APIkit Consoles
Prepare Response
Input
Payload : Unknown Define metadata
Variables
characters : Any Define metadata
Attributes : Unknown Define metadata
Context
Output Payload
4 issues found
Preview
%dw 2.0
output application/json

{
"results": vars.characters
}

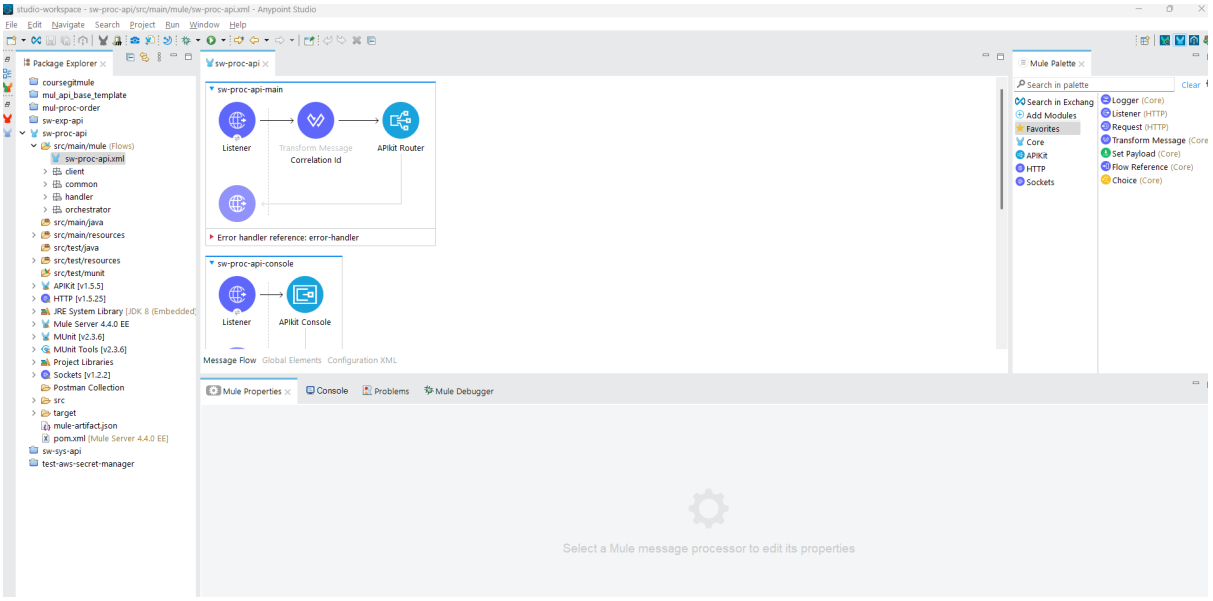
Process Api.

Para la capa de proceso tomó como base lo que se creó del api de sistema iniciando por el Raml, haciendo las adecuaciones necesarias en formatos, parámetros de entrada y salida, data types, examples.

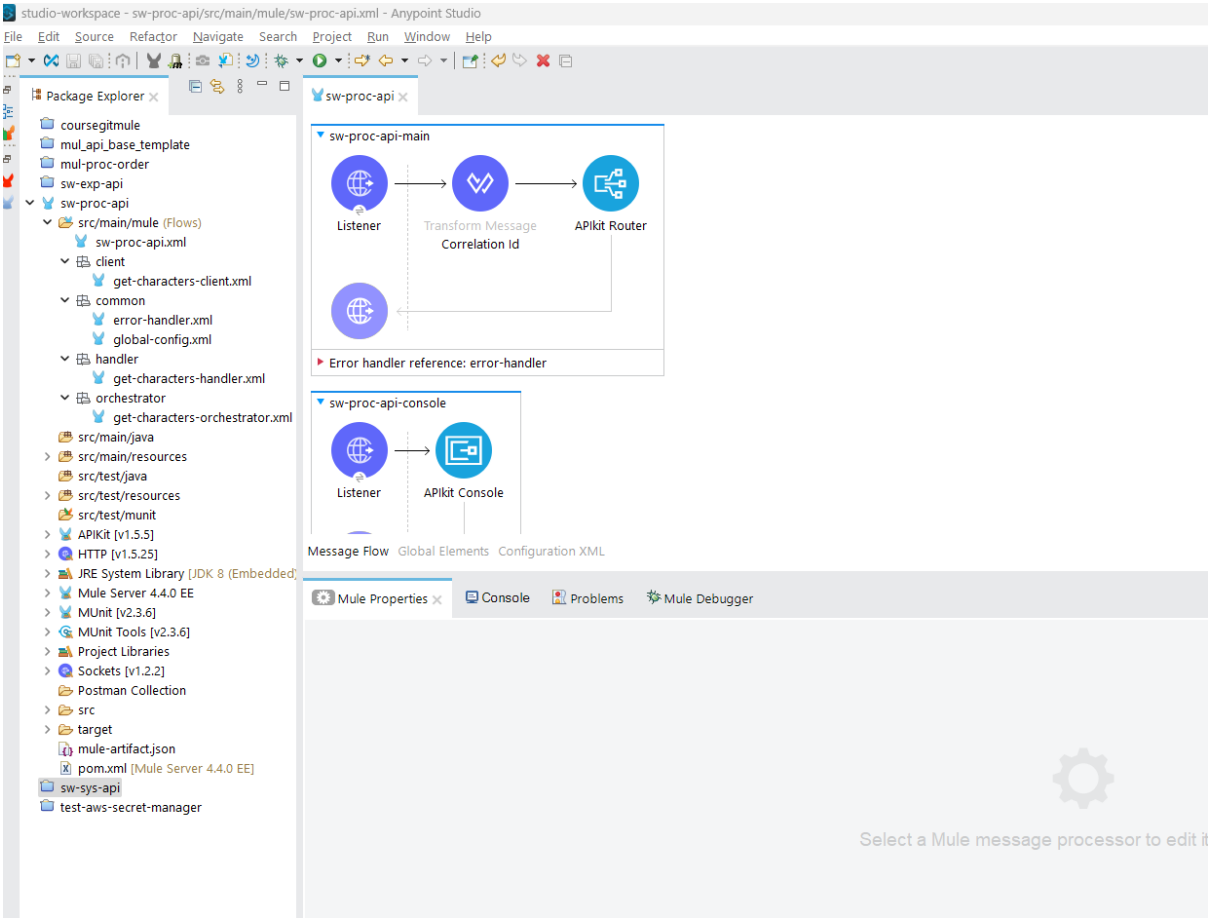




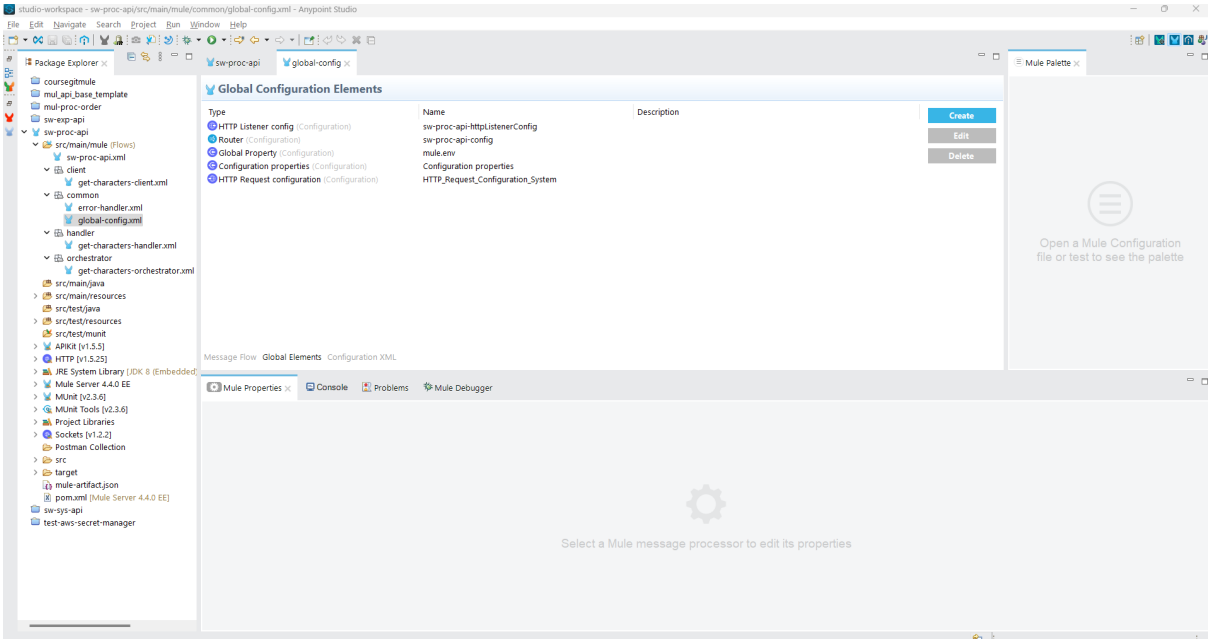
Iniciamos con el desarrollo del api en Anypoint Studio generando los flujos principales con la especificación local.



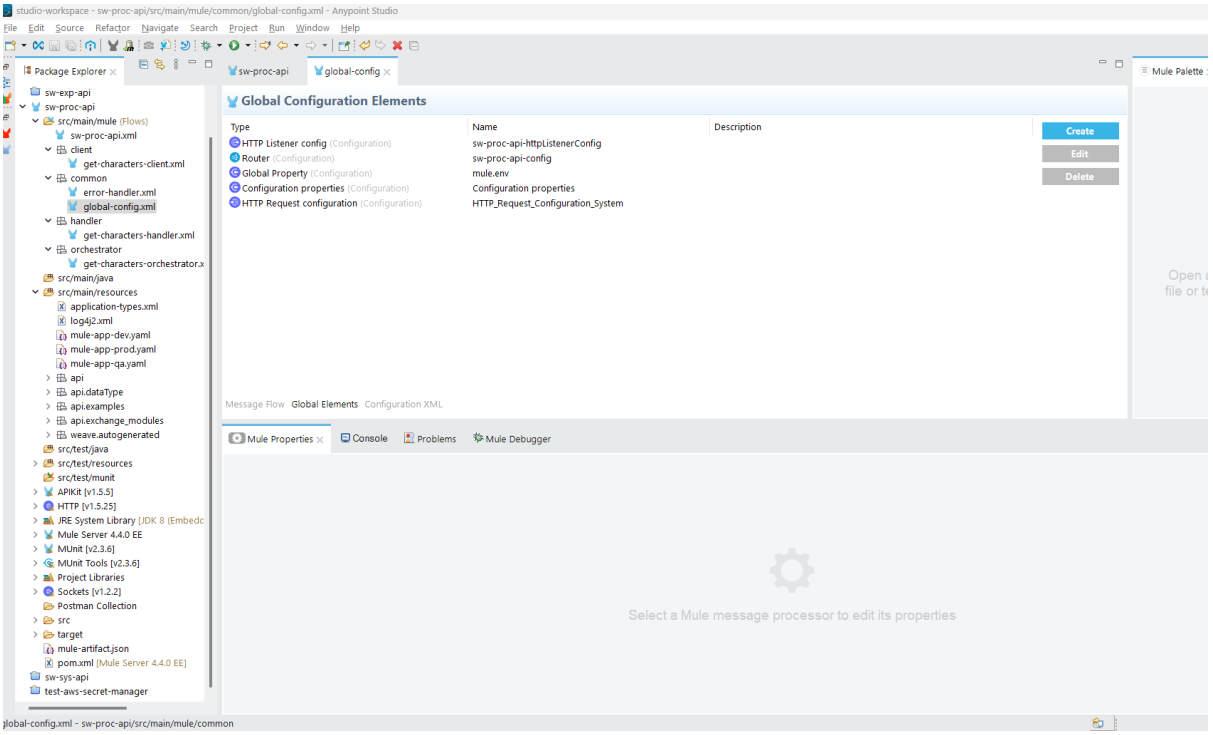
Continuar con la estructura del proyecto por carpetas.



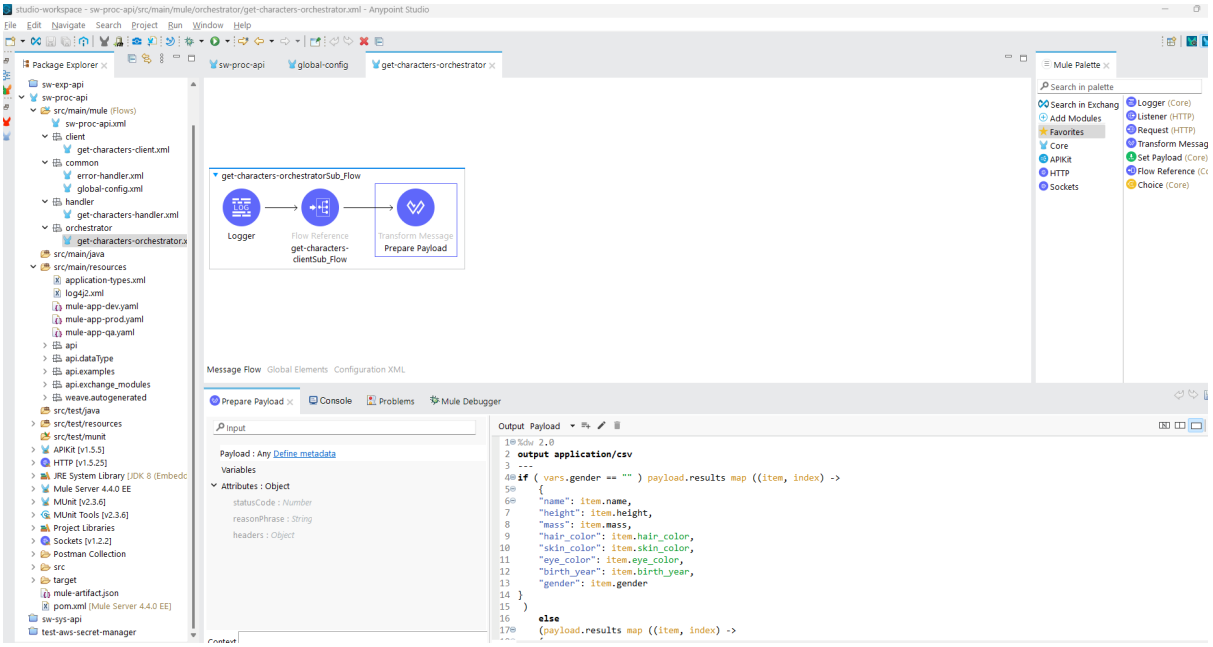
Generamos las configuraciones necesarias para la comunicación entre ambas capas.



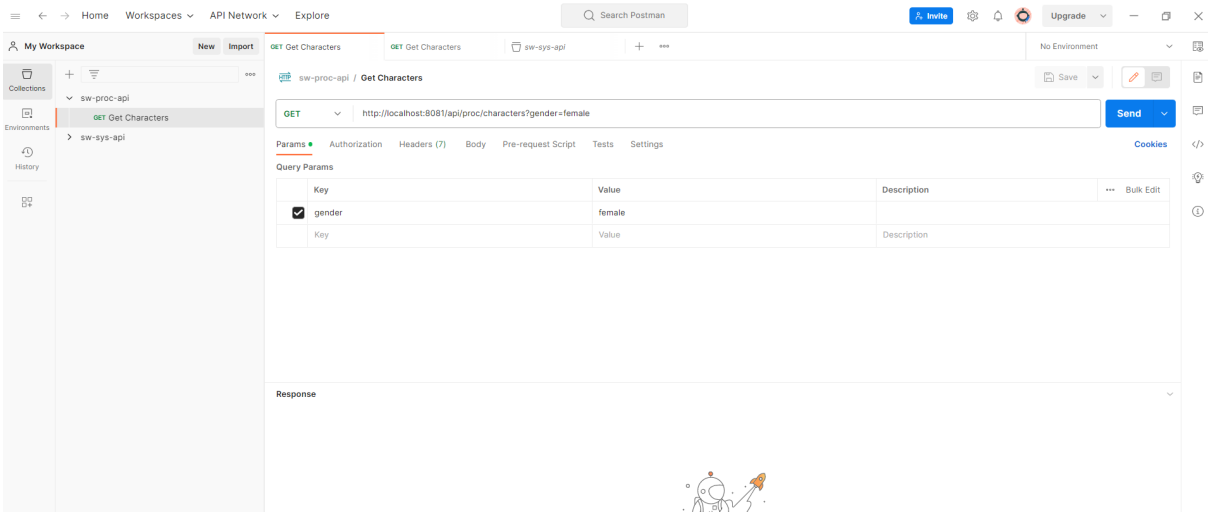
Los archivos properties por cada ambiente.



Agregamos las operaciones necesarias para devolver la respuesta deseada.



Realizamos pruebas con ayuda de Postman para validar que todo funcione como se espera.



Para ambas apis se deja cargado en el repositorio la colección Postman para la validación de su funcionamiento.

Se realizan únicamente 2 capas ya que por aprovechamiento de recursos (vCores) puede funcionar el api de sistema devolviendo toda la información en formato json para su consumos desde otra api y la capa de proceso únicamente filtrando y organizando de la forma esperada por el consumidor, haciendo que un posible cambio a futuro sea menos complicado y dando posibilidad de ser consumido por alguien más a la capa de sistema.