

S P S

TECH IS NOW

Desarrollador Mulesoft Jr/Mid

Práctica Técnica

Contenido

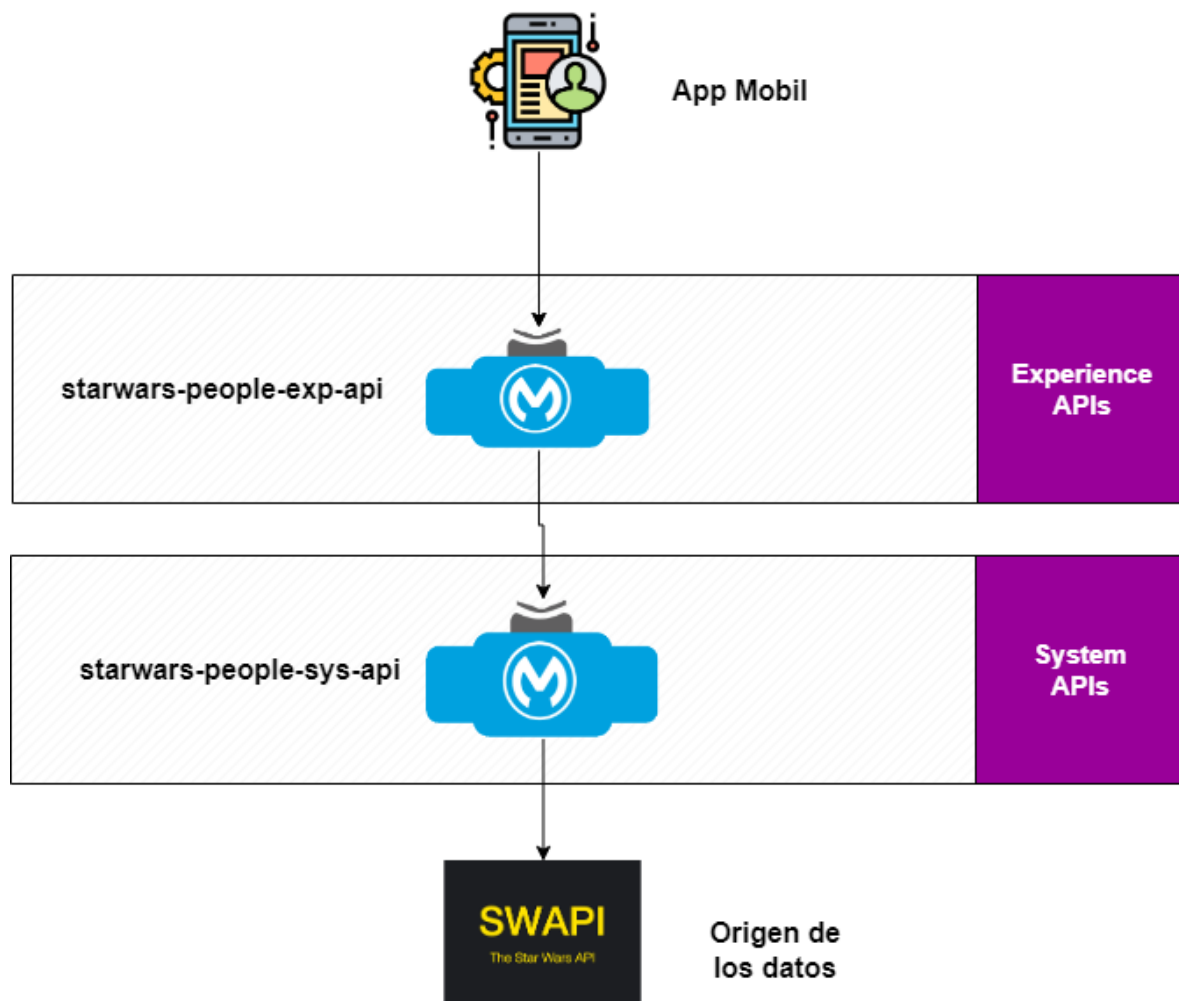
Introducción.....	1
Diagrama de Arquitectura.....	2
Diseño de la especificación de API's.....	3
Creación del proyecto Mulesoft.....	7
Repositorio.....	11

Introducción

En este documento podrán ver el paso a paso del como se fue desarrollando esta práctica, estaré seccionando los videos por partes, uno por cada tema dentro de este contenido, así como utilizare este documento para ir recabando la información e ir documentando el proceso.

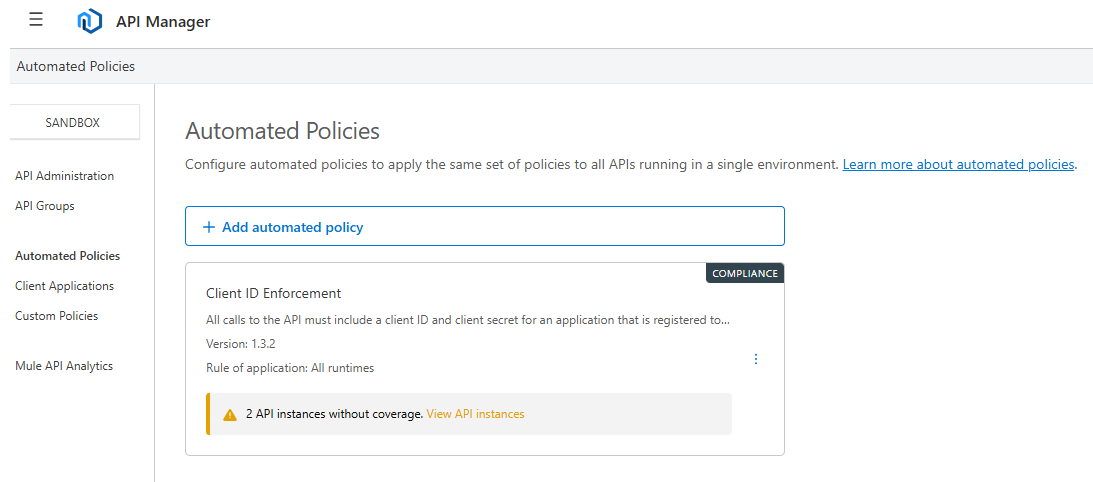
Diagrama de Arquitectura

En este diagrama fue creado en draw.io y en el se plasma la mejor solución posible a la práctica considerando las recomendaciones de mule api led, por lo que se visualiza desde el front, denominado aplicación móvil, de igual forma se considera una api en la capa de experiencia que será consumida por el front, llamada **starwars-people-exp-api**, también se considera una api en la capa de sistema, llamada **starwars-people-sys-api**, por último se visualiza la api de swapi la cual se toma como el origen de los datos.

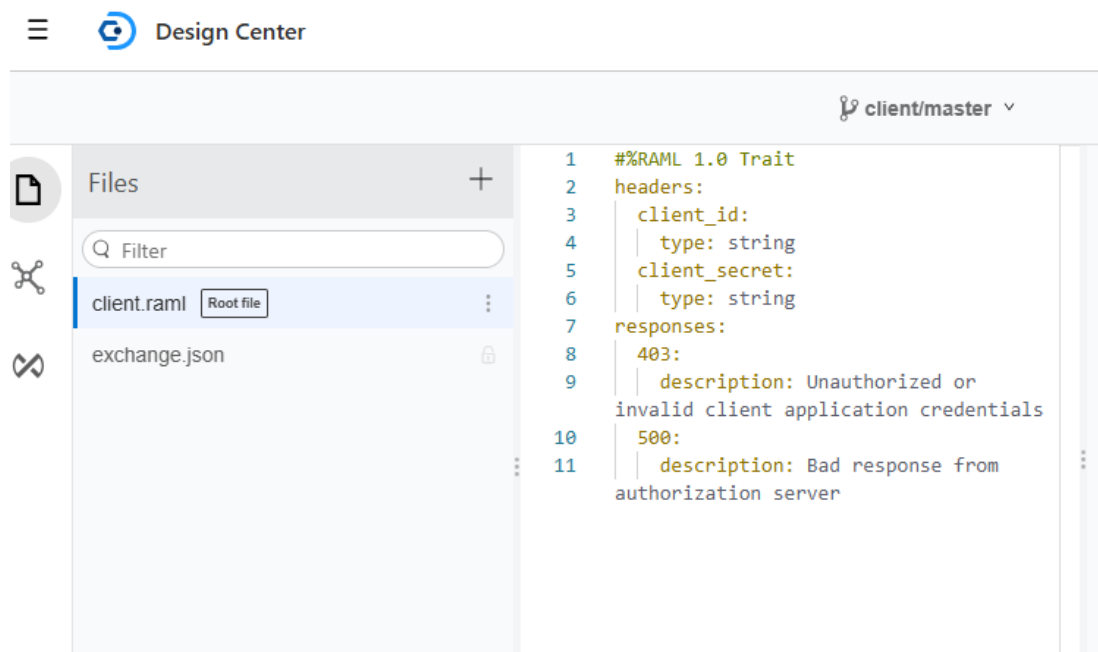


Diseño de la especificación de API's

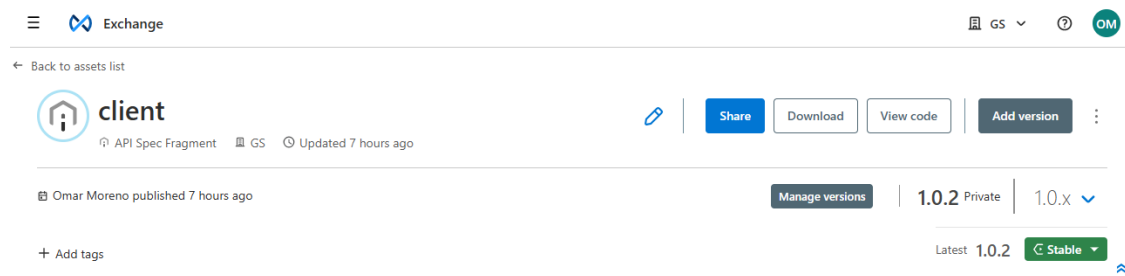
Antes de crear las especificaciones de las apis de experiencia y sistemas, se agregó la política de Client Id Enforcement dentro de API Manager:



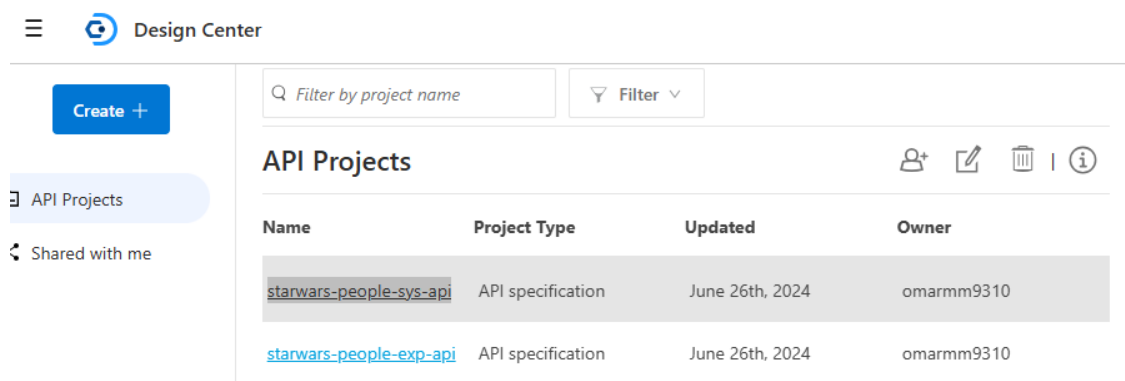
Para incluir esta política en la apis se creo un API Fragment el cual se nombre client y se configuro como se muestra en la imagen:



Y para poder incluirlo como dependencia dentro de las especificaciones de las apis se publicó en Exchange:



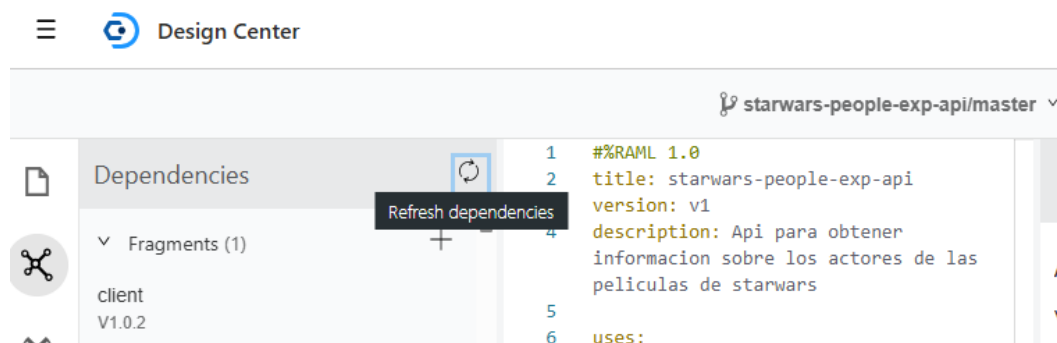
Una vez teniendo lo anterior procedimos a crear en Design center las especificaciones de estas apis starwars-people-exp-api y starwars-people-sys-api:



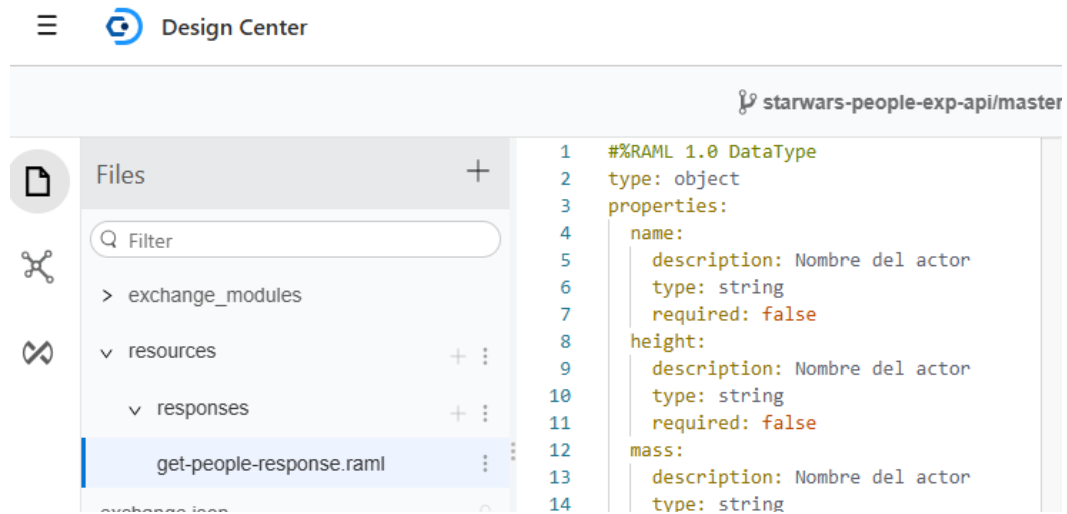
Para fines prácticos solo se agrego en el video la creación del api de experiencia.

Las especificaciones se estructuraron de la siguiente manera:

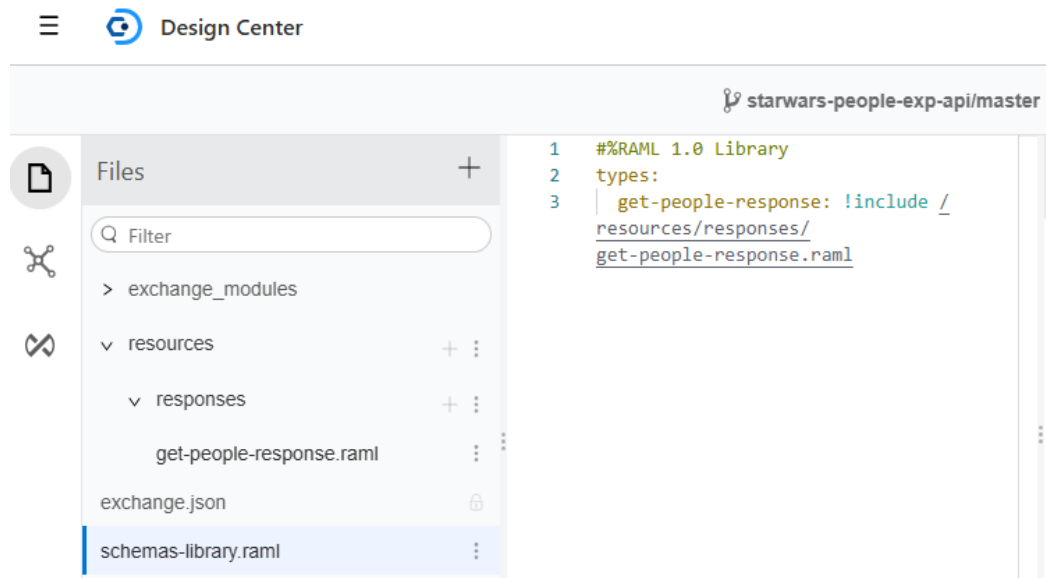
1. Se agregó la dependencia client.



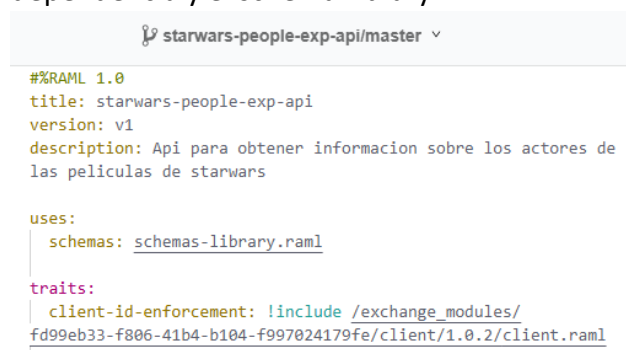
2. Se creó el folder resources, en donde se incluyó un folder llamado responses, con la respuesta esperada del método get:



3. Adicional se creó un esquema de tipos o dataTypes, en el cual se incluyó el response mencionado en el paso anterior.



4. En el root file del api de experiencia se incluyó el api fragment que agregamos como dependencia y el schema-library.

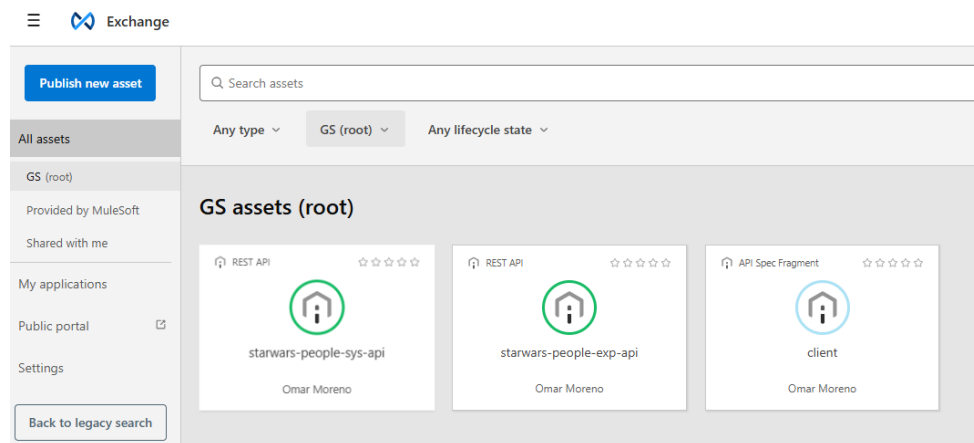


5. Definimos la estructura del método GET que utilizaremos, agregándole la política y el datatype mencionados.

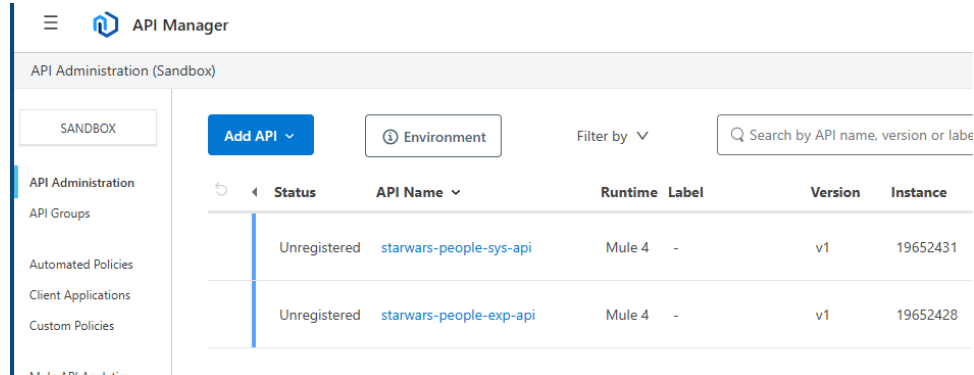
```
/people:
  get:
    is:
      - client-id-enforcement
    description: Recuperar información de todos los personajes
    de las películas de Starwars
    queryParameters:
      gender:
        type: string
        description: Filtrar por genero
        required: false
    responses:
      200:
        description: Successful response
        body:
          application/csv:
            schemas.get-people-response
```

6. Publicamos la apis en Exchange y en el portal público.

Url public portal: [Anypoint Exchange \(mulesoft.com\)](https://anypoint.mulesoft.com)





















7. Por último, registramos las apis en API manager copiando sus id de instancia.

















Creación del proyecto Mulesoft

Ambos proyectos para las apis de experiencia y sistema se crearon y estructuraron tomando en cuenta un posible escalamiento o crecimiento en las apis, así mismo se utilizaron las mejores practicas en el nombrado de los archivos dentro de la estructura y los componentes utilizados, a continuación, se describe brevemente lo que se realizó para las apis.

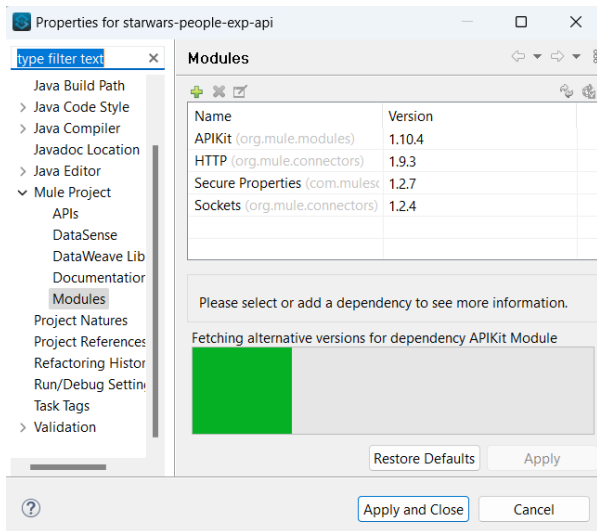
1. La estructura quedo de la siguiente manera para el api de sistema:

- ▼  starwars-people-sys-api [starwars main]
 - ▼  src/main/mule (Flows)
 -  error-handler.xml
 -  global-config.xml
 -  starwars-people-exp-api.xml
 -  starwars-swapi.xml
 -  utils.xml
 -  src/main/java
 - ▼  src/main/resources
 -  application-types.xml
 -  log4j2.xml
 -  api
 - ▼  dwl
 -  filter-attributes-people.dwl
 -  filter-attributes-people-gender.dwl
 -  set-variable-gender.dwl
 - ▼  properties
 -  local-properties.yaml

2. Para el api de experiencia quedo así:

- ▼  > starwars-people-exp-api [starwars main]
 - ▼  src/main/mule (Flows)
 -  error-handler.xml
 -  global-config.xml
 -  people-sys.xml
 -  starwars-people-exp-api.xml
 -  utils.xml
 -  src/main/java
 - ▼  > src/main/resources
 -  application-types.xml
 -  log4j2.xml
 - >  > dwl
 - ▼  properties
 -  local-properties.yaml

3. Tome en cuenta agregar el módulo de secure properties, para encriptar el header `client_secret`:



Para fines prácticos encriptamos el archivo `local-properties.yaml` utilizando el siguiente comando, dando como resultado otro archivo `local-out.yaml`.

```
java -cp C:\Mule\secure-properties-tool.jar com.mulesoft.tools.SecurePropertiesTool file encrypt AES CBC A792C3A57A024CE3479B879360FFA50B local-properties.yaml local_out.yaml
```

Y solo sustituimos en el archivo `local-properties.yaml` el valor del campo `client-secret`.

```
1@api:
2  instancia: "19652428"
3
4=http:
5  port: "8081"
6=https:
7  port: "8092"
8
9=people:
10 host: "localhost"
11 port: "8082"
12 basePath: "/api"
13 path: "/people"
14 client_id: "d07e02e8d861433cb5c0d2d05d6f3944"
15 client_secret: "![FpSSItSn5ML+rvjsgpSLEvHmF4ft3fXB07glRh7yBrrQi6m3PTikf5UNvm9LGcgE]"
```

4. Cree el archivo global-config.xml y lo estructure de la siguiente manera:

Global Configuration Elements	
Type	Name
Secure Properties Config (Configuration)	Secure_Properties_Config
Global Property (Configuration)	mule.key
Global Property (Configuration)	mule.env
HTTP Listener config (Configuration)	HTTP_Listener_config
API Autodiscovery (Configuration)	API Autodiscovery
HTTP Request configuration (Configuration)	starwars-people-sys-api

Lo más destacable en la imagen son los elementos globales, Secure Properties Config, HTTP Request configuration y API autodiscovery.

5. Para la etapa de pruebas se consideró cubrir los requisitos de la práctica, los cuales detallo a continuación:

- Recuperación de toda la data de todos los personajes:

HTTP Practica Starwars / people

GET http://localhost:8081/api/people

Params • Authorization Headers (8) Body Scripts Settings

Query Params

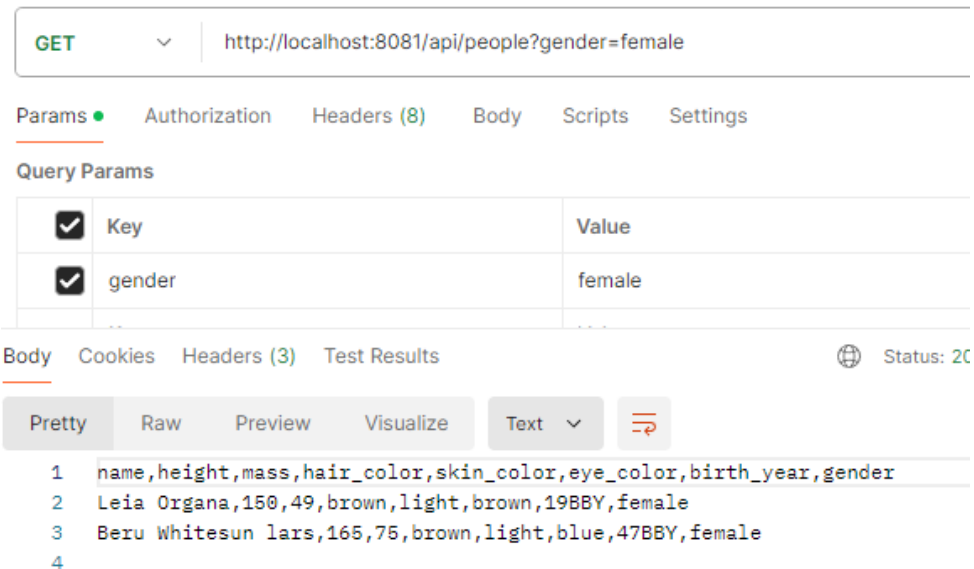
<input type="checkbox"/>	Key	Value
<input type="checkbox"/>	gender	female

Body Cookies Headers (3) Test Results Status: 200 OK

Pretty Raw Preview Visualize Text

```
1 name,height,mass,hair_color,skin_color,eye_color,birth_year,gender
2 Luke Skywalker,172,77,blond,fair,blue,19BBY,male
3 C-3PO,167,75,n/a,gold,yellow,112BBY,n/a
4 R2-D2,96,32,n/a,white\, blue,red,33BBY,n/a
5 Darth Vader,202,136,none,white,yellow,41.9BBY,male
6 Leia Organa,150,49,brown,light,brown,19BBY,female
7 Owen Lars,178,120,brown\, grey,light,blue,52BBY,male
8 Beru Whitesun lars,165,75,brown,light,blue,47BBY,female
9 R5-D4,97,32,n/a,white\, red,red,unknown,n/a
10 Biggs Darklighter,183,84,black,light,brown,24BBY,male
11 Obi-Wan Kenobi,182,77,auburn\, white,fair,blue-gray,57BBY,male
12
```

- Obtener la data a través de un Query Parameter basándose en el género del personaje gender.



GET ▼ | http://localhost:8081/api/people?gender=female

Params ● Authorization Headers (8) Body Scripts Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	gender	female

Body Cookies Headers (3) Test Results 🌐 Status: 200 OK

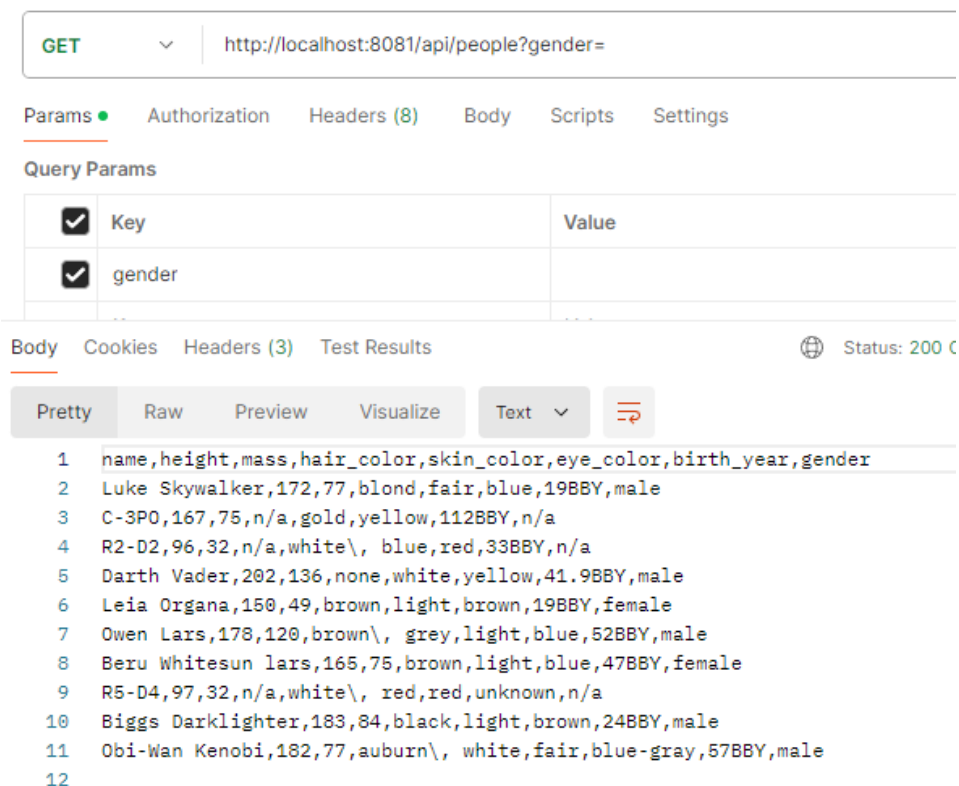
Pretty Raw Preview Visualize Text ▼ 🔗

```

1  name,height,mass,hair_color,skin_color,eye_color,birth_year,gender
2  Leia Organa,150,49,brown,light,brown,19BBY,female
3  Beru Whitesun lars,165,75,brown,light,blue,47BBY,female
4

```

- Por ultimo se agrego la funcionalidad para cuando se envié el query param vacío, las apis regresaran toda la Data, como si no se hubiera enviado el query param:



GET ▼ | http://localhost:8081/api/people?gender=

Params ● Authorization Headers (8) Body Scripts Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	gender	

Body Cookies Headers (3) Test Results 🌐 Status: 200 OK

Pretty Raw Preview Visualize Text ▼ 🔗

```

1  name,height,mass,hair_color,skin_color,eye_color,birth_year,gender
2  Luke Skywalker,172,77,blond,fair,blue,19BBY,male
3  C-3PO,167,75,n/a,gold,yellow,112BBY,n/a
4  R2-D2,96,32,n/a,white\, blue,red,33BBY,n/a
5  Darth Vader,202,136,none,white,yellow,41.9BBY,male
6  Leia Organa,150,49,brown,light,brown,19BBY,female
7  Owen Lars,178,120,brown\, grey,light,blue,52BBY,male
8  Beru Whitesun lars,165,75,brown,light,blue,47BBY,female
9  R5-D4,97,32,n/a,white\, red,red,unknown,n/a
10 Biggs Darklighter,183,84,black,light,brown,24BBY,male
11 Obi-Wan Kenobi,182,77,auburn\, white,fair,blue-gray,57BBY,male
12

```

Repositorio

Les comparto los recursos de la practica:

Url repositorio: [GitHub - omarmm9310/starwars](https://github.com/omarmm9310/starwars)

Url Exchange public portal: [Anypoint Exchange \(mulesoft.com\)](https://anypoint.mulesoft.com/)

Urls de los videos de evidencia: <https://we.tl/t-75ua2u3mKD>