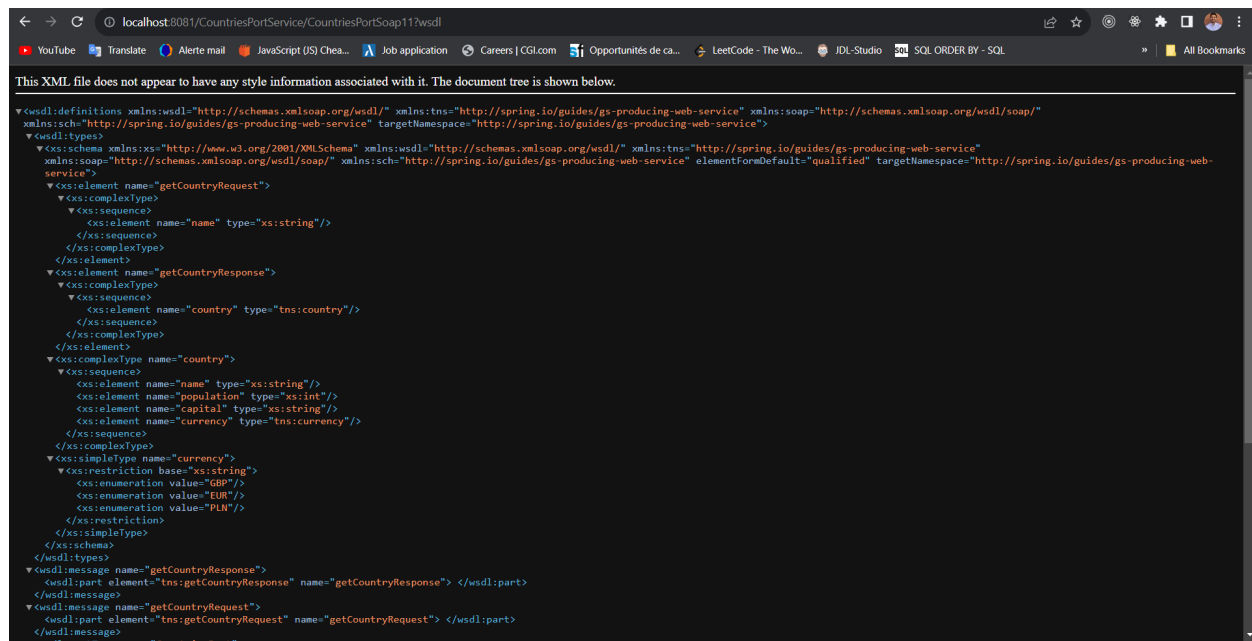


# Compte rendu du TP1 en urbanisation des SI

## Exposition d'un service web SOAP à partir d'un fichier WSDL existant

Après avoir suivi les étapes de création du projet, On peut voir que le wsdl est bel et bien exposé

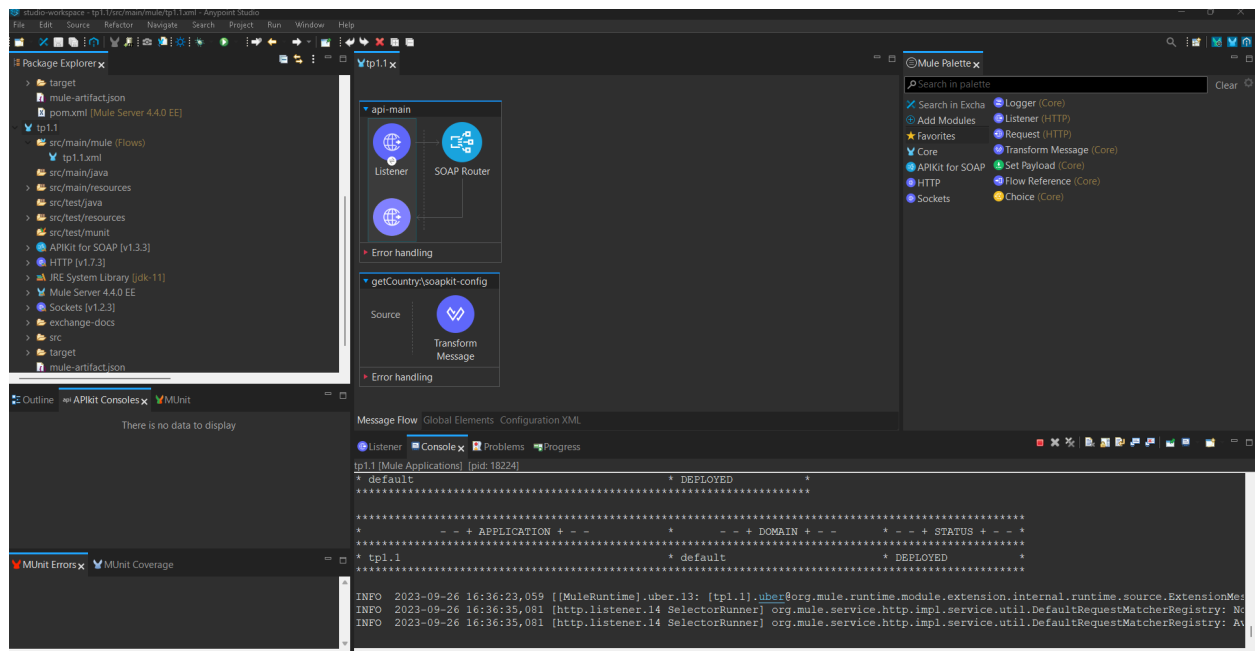


The screenshot shows a web browser window with the address bar displaying 'localhost:8081/CountriesPortService/CountriesPortSoap11?wsdl'. The page content shows the XML structure of the WSDL file. A message at the top states: 'This XML file does not appear to have any style information associated with it. The document tree is shown below.' The XML content is as follows:

```
<?xml version='1.0' encoding='UTF-8'>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://spring.io/guides/gs-producing-web-service" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:sch="http://spring.io/guides/gs-producing-web-service" targetNamespace="http://spring.io/guides/gs-producing-web-service">
  <wsdl:types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://spring.io/guides/gs-producing-web-service"
      xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:sch="http://spring.io/guides/gs-producing-web-service" elementFormDefault="qualified" targetNamespace="http://spring.io/guides/gs-producing-web-service">
      <xsd:element name="getCountryRequest">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="name" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="getCountryResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="country" type="tns:country"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:complexType name="country">
        <xsd:sequence>
          <xsd:element name="name" type="xsd:string"/>
          <xsd:element name="population" type="xsd:int"/>
          <xsd:element name="capital" type="xsd:string"/>
          <xsd:element name="currency" type="tns:currency"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:simpleType name="currency">
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="GBP"/>
          <xsd:enumeration value="EUR"/>
          <xsd:enumeration value="PLN"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="getCountryResponse">
    <wsdl:part element="tns:getCountryResponse" name="getCountryResponse"> </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getCountryRequest">
    <wsdl:part element="tns:getCountryRequest" name="getCountryRequest"> </wsdl:part>
  </wsdl:message>
</wsdl:definitions>
```

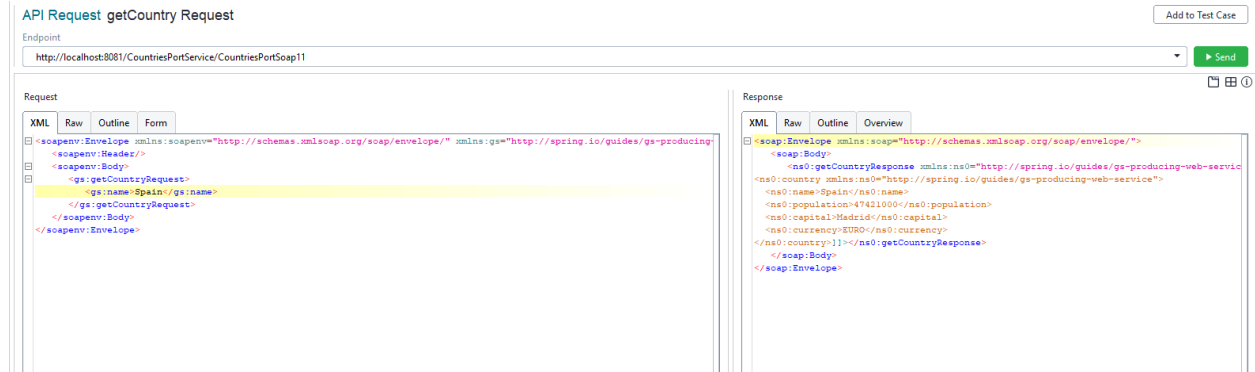
## Test du service web avec SOAP UI

Initialisation du projet



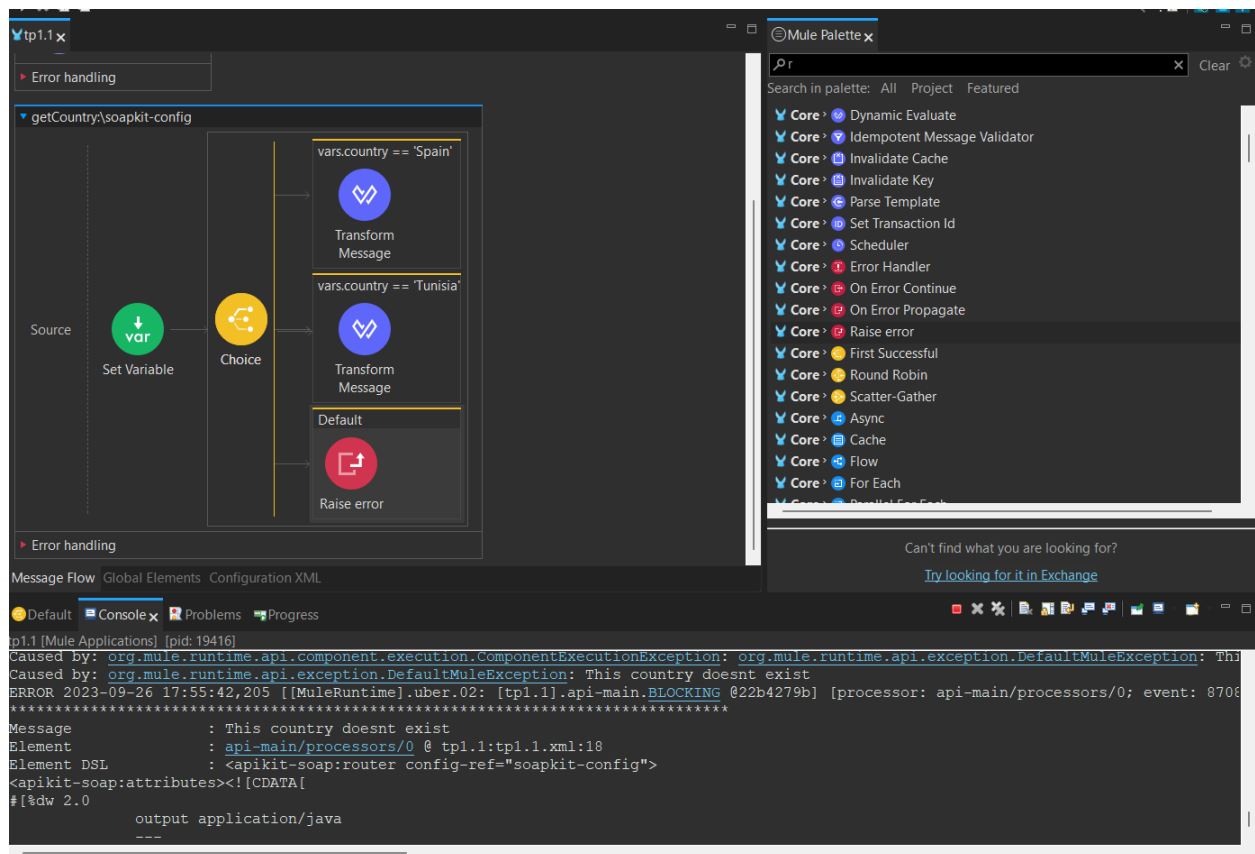
## Implémentation du service

La première requête en utilisant ReadyAPI :

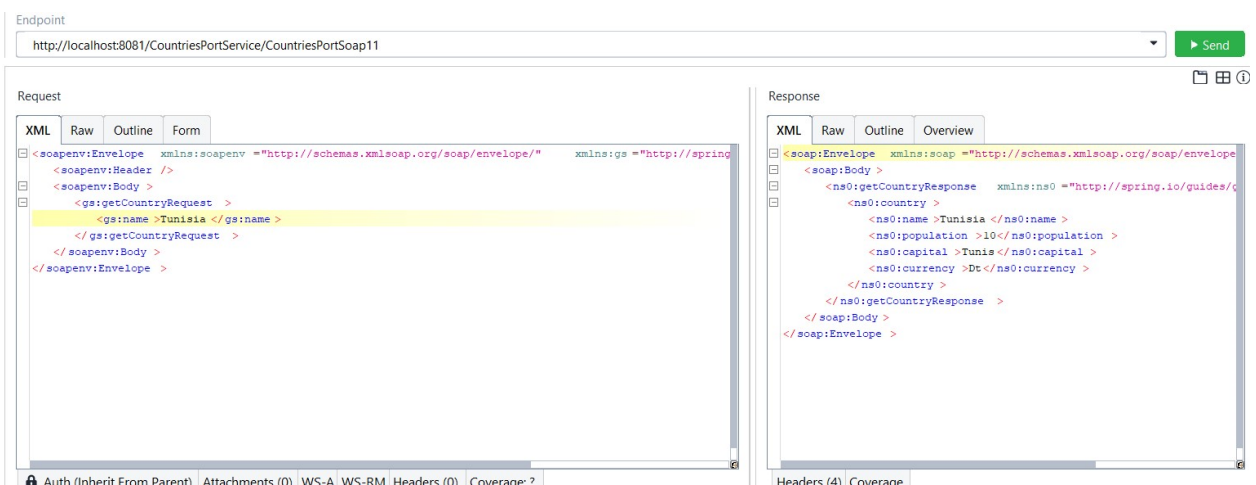


## Implémentation d'une condition

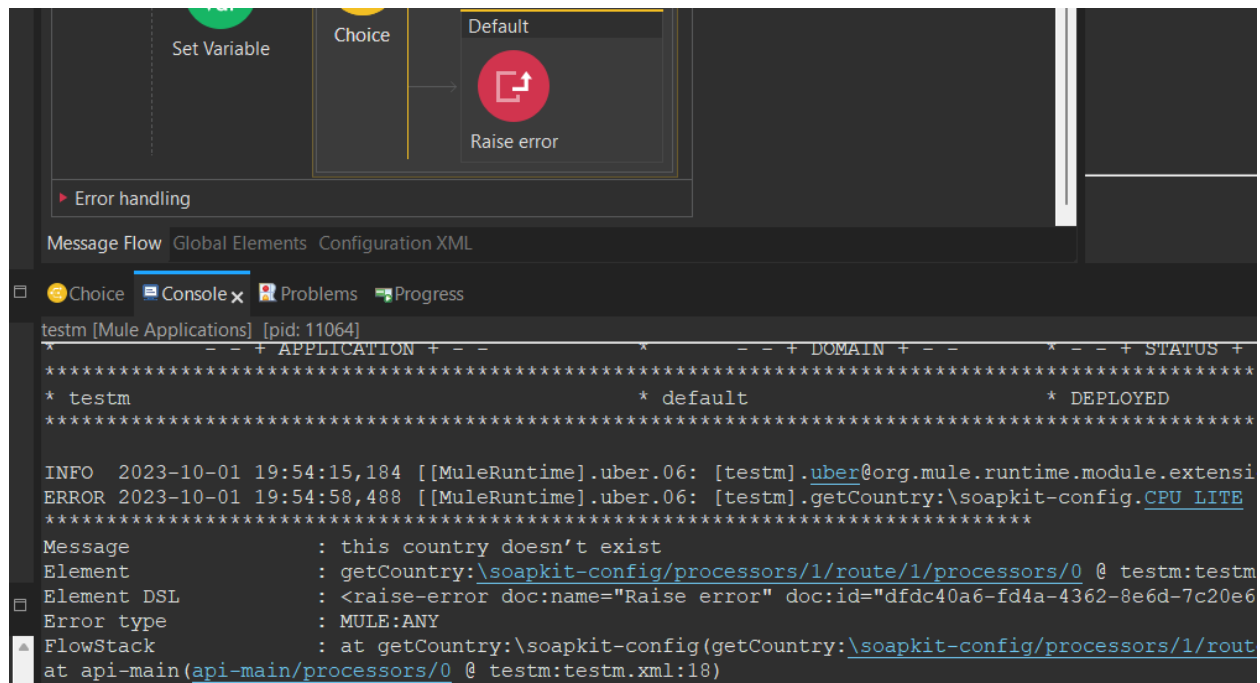
On a implémenté 2 conditions, le cas où le pays demandé est 'Spain' et l'autre où le pays demandé est 'Tunisia', sinon on va afficher le message :this country doesn't exist



Le cas où le pays de la requête est la Tunisie :



Dans le cas échéant , on aura une erreur 500 (qui n'est pas la meilleure façon de gérer un cas échéant) mais on a le message : this country doesn't exist



## Consultation d'une base de données :

On ouvre le serveur apache xampp, on crée la base de données nommée countries

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=countries&table=countries

YouTube Translate Alerte mail JavaScript (JS) Chea... Stage GL4 2022 202... Easy apply - INGÉN... databa

phpMyAdmin

Récentes Préférences

Nouvelle base de données

- countries
  - Nouvelle table
  - countries
- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- shop\_db
- test

Serveur : 127.0.0.1 » Base de données : countries » Table : countries

Parcourir Structure SQL Rechercher Insérer Exp

SELECT \* FROM `countries`

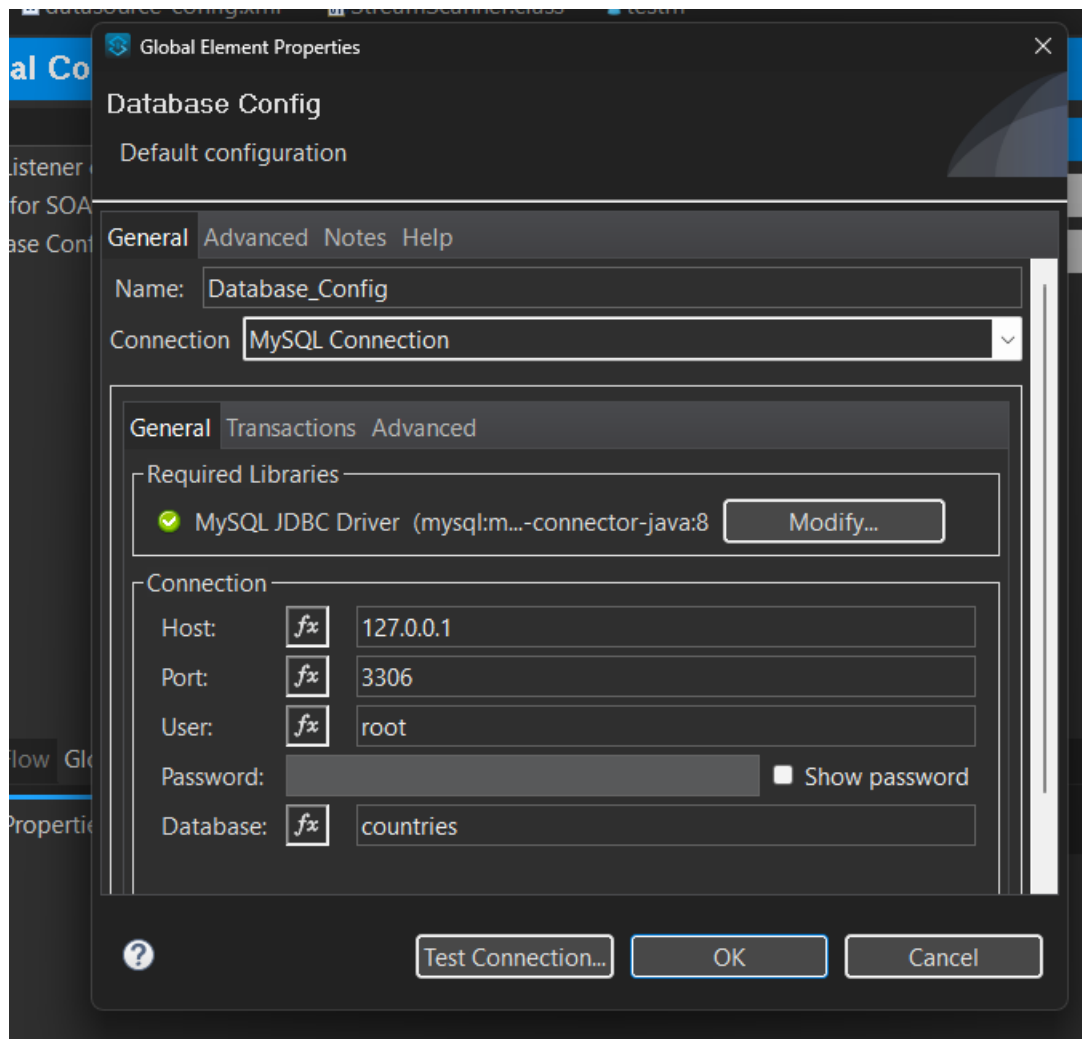
☐ Profilage [ Éditer en ligne ] [ Éditer ] [ Expliquer SQL ] [ Créer le code source PHP ] [ Actual

☐ Tout afficher | Nombre de lignes : 25 Filtre les lignes: Chercher dans

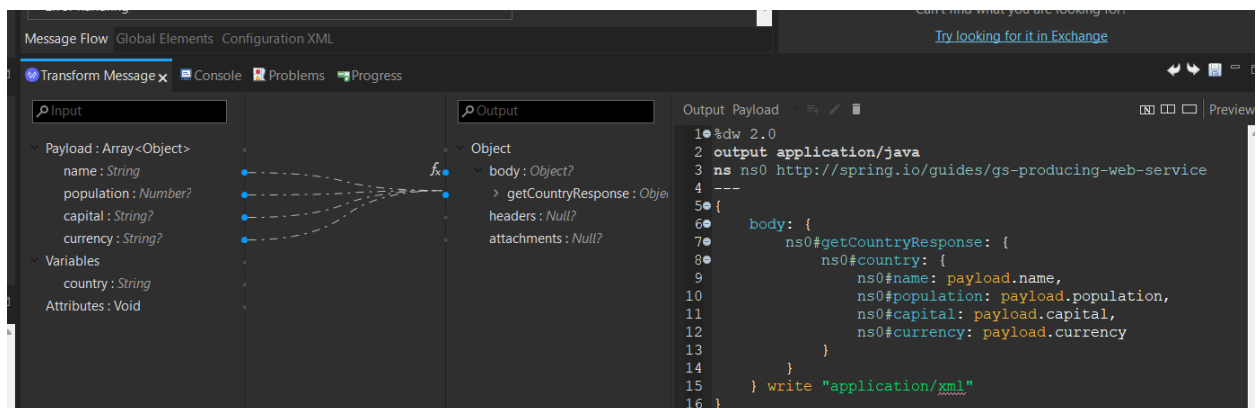
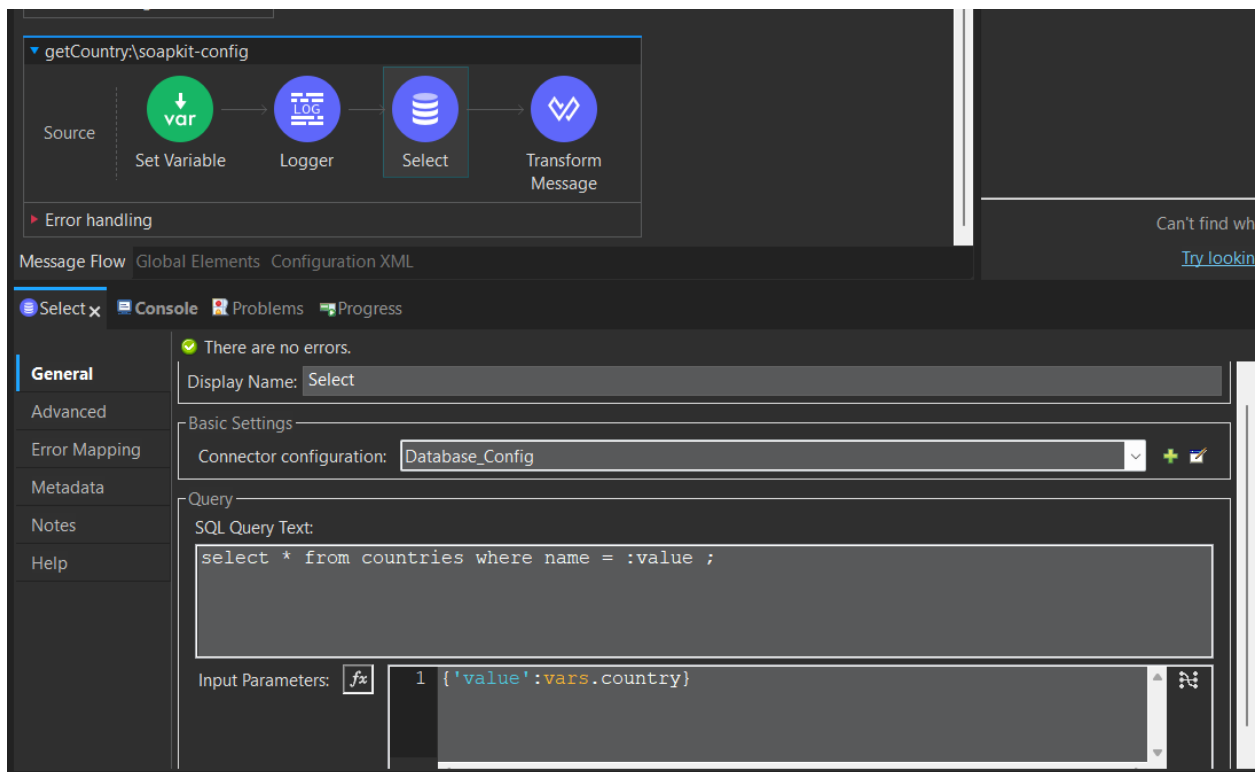
Options supplémentaires

				name	population	capital	currency
<input type="checkbox"/>	Éditer	Copier	Supprimer	Armenia	3000000	Yerevan	AMD
<input type="checkbox"/>	Éditer	Copier	Supprimer	Germany	83000000	Berlin	EUR
<input type="checkbox"/>	Éditer	Copier	Supprimer	Poland	38000000	Warsaw	PLN
<input type="checkbox"/>	Éditer	Copier	Supprimer	Russia	145000000	Moscow	RUB
<input type="checkbox"/>	Éditer	Copier	Supprimer	Spain	47420000	Madrid	EUR
<input type="checkbox"/>	Éditer	Copier	Supprimer	Tunisia	12260000	Tunis	TND

On installe la dépendance maven de mysql-connector-java



On introduit la variable 'value' afin de faire la liaison entre la variable 'name' dans le corps de la requête xml et la condition de la requête sql,



Le résultat est le suivant :

The screenshot shows the Anypoint Studio interface for testing a SOAP API. The endpoint is `http://localhost:8081/CountriesPortService/CountriesPortSoap11`. The request is a SOAP envelope with a `getCountryRequest` containing the name `Russia`. The response is a SOAP envelope with a `getCountryResponse` containing details for Russia: name, population, capital, and currency.

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:gs="http://spring.io/guides/gs">
  <soap:Header/>
  <soap:Body>
    <gs:getCountryRequest>
      <gs:name>Russia</gs:name>
    </gs:getCountryRequest>
  </soap:Body>
</soap:Envelope>
```

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns0="http://spring.io/guides/gs">
  <soap:Header/>
  <soap:Body>
    <ns0:getCountryResponse xmlns:ns0="http://spring.io/guides/gs">
      <ns0:country>
        <ns0:name>Russia</ns0:name>
        <ns0:population>145000000</ns0:population>
        <ns0:capital>Moscow</ns0:capital>
        <ns0:currency>RUB</ns0:currency>
      </ns0:country>
    </ns0:getCountryResponse>
  </soap:Body>
</soap:Envelope>
```

The screenshot shows the Anypoint Studio interface for testing a SOAP API. The endpoint is `http://localhost:8081/CountriesPortService/CountriesPortSoap11`. The request is a SOAP envelope with a `getCountryRequest` containing the name `Germany`. The response is a SOAP envelope with a `getCountryResponse` containing details for Germany: name, population, capital, and currency.

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:gs="http://spring.io/guides/gs">
  <soap:Header/>
  <soap:Body>
    <gs:getCountryRequest>
      <gs:name>Germany</gs:name>
    </gs:getCountryRequest>
  </soap:Body>
</soap:Envelope>
```

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns0="http://spring.io/guides/gs">
  <soap:Header/>
  <soap:Body>
    <ns0:getCountryResponse xmlns:ns0="http://spring.io/guides/gs">
      <ns0:country>
        <ns0:name>Germany</ns0:name>
        <ns0:population>83000000</ns0:population>
        <ns0:capital>Berlin</ns0:capital>
        <ns0:currency>EUR</ns0:currency>
      </ns0:country>
    </ns0:getCountryResponse>
  </soap:Body>
</soap:Envelope>
```

Conclusion : on a pu créer une API SOAP à partir des composants de l'interface de Anypoint studio sans devoir écrire du code (seulement la requête sql) et on a testé le fonctionnement des requêtes à partir de ReadyAPI avec succès