TP1 - Services Web SOAP avec Mule ESB

Télécharger PDF

Objectifs du TP

Création et consommation de web services SOAP en utilisant l'outil Anypoint Studio.

Outils et Versions

- Anypoint Studio 7 and Mule ESB 4
- MySQL Version *latest*
- SOAPUI (Open Source) Version 5.7.0
- VSCode Version *latest*

Présentation des outils

Anypoint Platform

Anypoint est une plateforme développée par l'entreprise Mulesoft qui offre les outils nécessaires pour la gestion d'APIs et l'intégration de services. Grâce à Anypoint, Mulesoft est classée par Gartner dans son Magic Quadrant dans la rubrique "Enterprise Integration Platform as a Service" de Décembre 2022 parmi les leaders du marché.

Figure 1: Magic Quadrant for Enterprise Integration Platform as a Service

Source: Gartner (January 2023)

Mule ESB

Mule, le runtime engine d'Anypoint Platform, est un ESB (enterprise service bus) léger basé sur Java ainsi qu'une plateforme d'intégration qui permet aux développeurs de connecter des applications rapidement et facilement afin qu'elles puissent échanger des données. Cela facilite l'intégration des systèmes existants, quelles que soient les technologies utilisées par les applications, notamment JMS, Web Services, JDBC, HTTP, etc. Cet ESB, déployable n'importe où, qui intègre et orchestre les événements en temps réel ou par lots, dispose d'une connectivité universelle.

Mule ESB propose les services suivants:

- Création et hébergement de services
- Médiation de services
- Routage des messages

- Transformation des données

Anypoint Studio

Anypoint Studio fournit une interface graphique de développement pour implémenter, compiler, tester et publier des services Web, des services de données et des routes de messages.

Création d'un service web SOAP avec Anypoint Studio et Mule

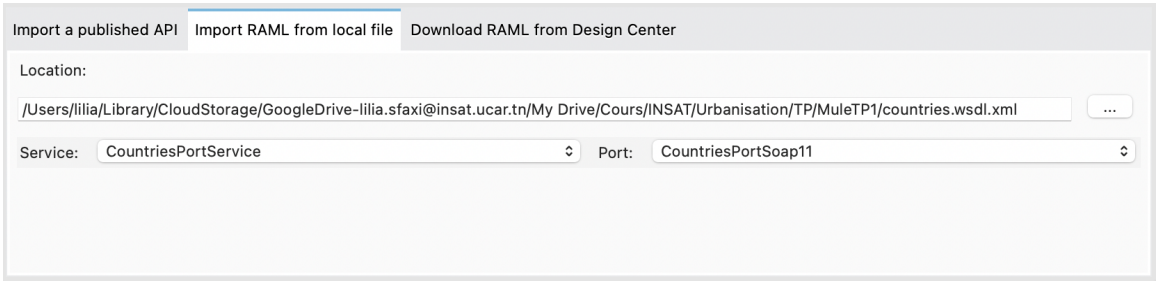
Il est possible de créer un service web SOAP avec Anypoint en utilisant les connecteurs fournis, et très peu de lignes de code. Nous allons simuler le comportement du service web décrit dans le tutoriel suivant: `https://spring.io/guides/gs/producing-web-service`.

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

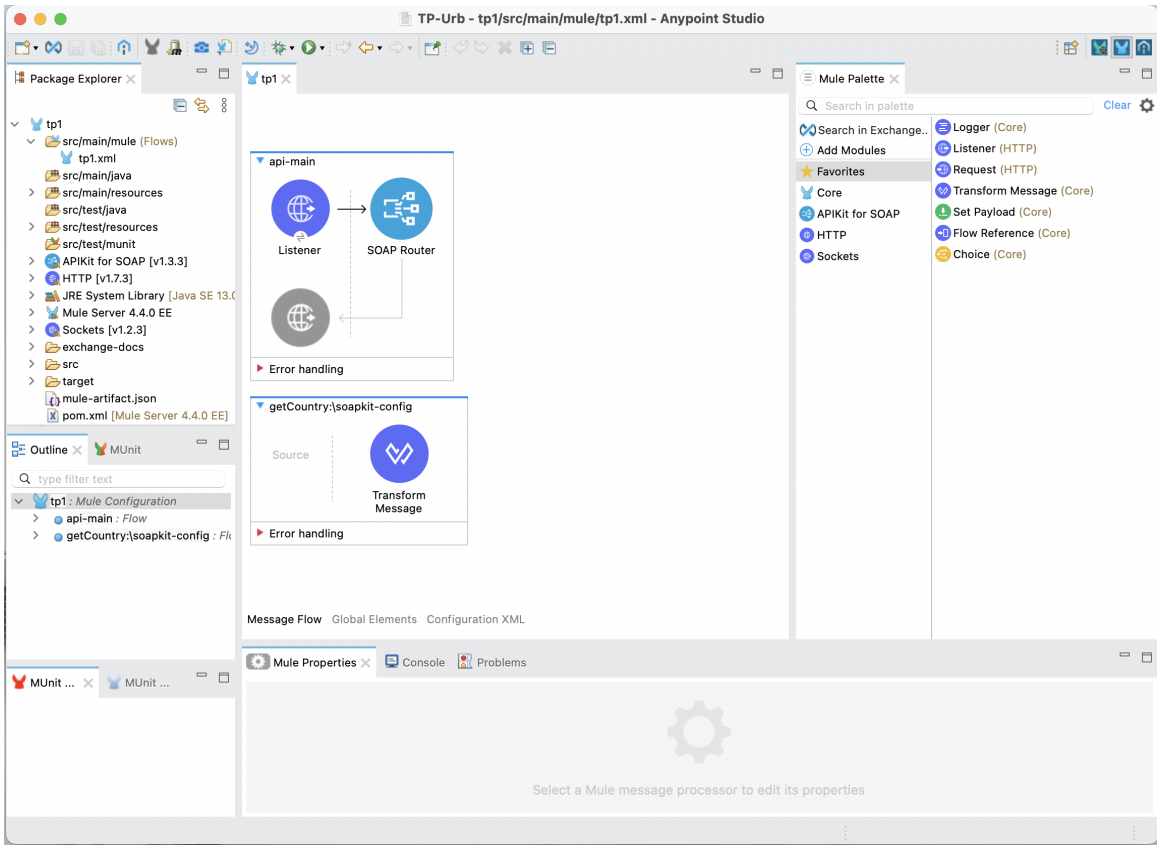
Nous récupérons d'abord le fichier WSDL du service. Il est possible de générer un fichier WSDL avec des outils tel que Eclipse, mais nous allons nous contenter d'en utiliser un qui est prêt. Vous trouverez le fichier wsdl à télécharger ici : .

Le web service SOAP que nous allons créer prend en entrée un pays (nous accepterons pour le moment *Spain* ou *Poland*), et nous donne sa population, sa capitale et sa monnaie. Nous allons suivre les étapes suivantes pour créer ce service avec Anypoint Studio.

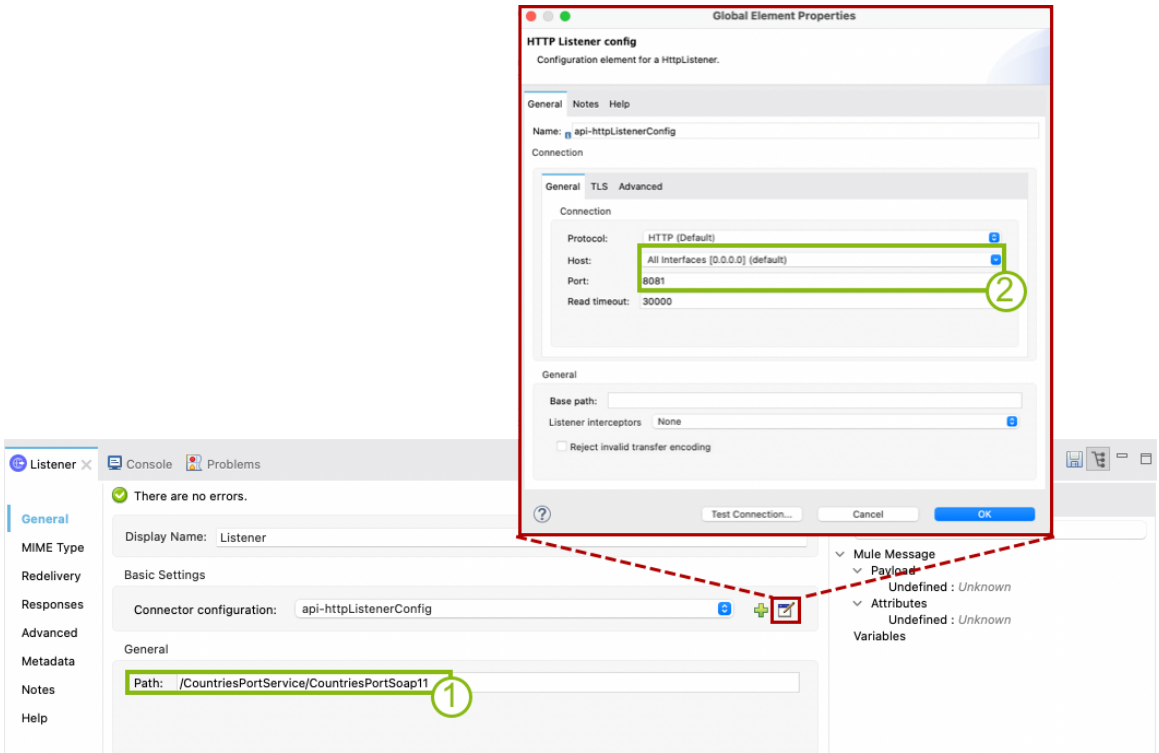
- Créer un nouveau Mule Project, qu'on appellera TP1. Dans la fenêtre de création :
  - Indiquer que le runtime est bien Mule Server 4.4.0
  - Installer le runtime supplémentaire suivant: *Anypoint APIkit SOAP Plugin*
  - Indiquer que le wsdl est le fichier que vous venez de télécharger. Pour cela, cliquer sur *"Import RAML from local file"* et choisir le fichier wsdl. Vérifier que le service et le port s'affichent bien comme suit:



Une fois le projet créé, la fenêtre suivante va apparaître:



En cliquant sur **Listener**, vous pouvez trouver (en bas de l'écran) les informations de base du service SOAP exposé, qui est décrit par votre wsdl, tel qu'indiqué dans la figure suivante:



Le path (1), domaine et port (2) permettent de représenter l'URL du service qui sera exposé. Pour tester cela, lancer le service, en faisant un clic-droit sur la fenêtre principale, et en choisissant: *Run project tp1*.

> **Remarque**
>
> Prenez soin d'utiliser une version de JDK entre 8 et 12, car ce sont celles compatibles avec le serveur Mule 4.4.

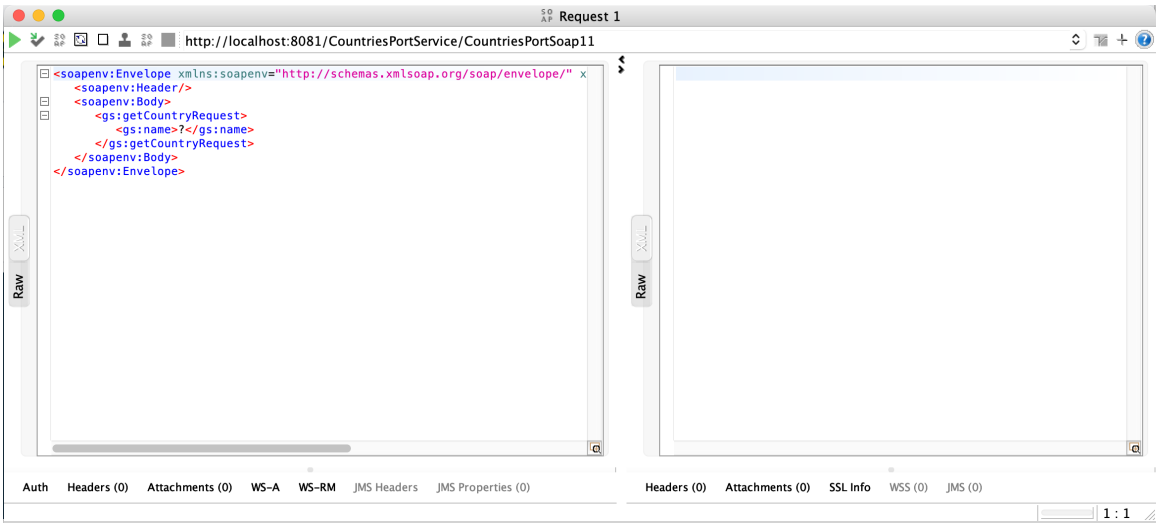
Une fois le service lancé, vérifiez bien que le wsdl est exposé, sur l'URL:
`http://localhost:8081/CountriesPortService/CountriesPortSoap11?wsdl`

2. Test du service web avec SOAP UI

Nous allons maintenant tester notre service web. Comme vous le savez certainement, pour tester un service web SOAP, il faut écrire une SOAP Request (en XML), pour obtenir un SOAP Response de la part du service. Pour cela, nous utilisons un outil de test léger, appelé SOAP UI.

Lancer SOAP UI, et créer un nouveau projet SOAP. Ensuite:

- Nommer le projet TP1 par exemple.
- Coller l'URL du WSDL dans le champs *Initial WSDL*
- Dans le projet créé, naviguer vers *TP1 -> CountriesPortSoap11 -> getCountry -> Request1* et double cliquer dessus. Une fenêtre s'ouvre, comme suit:



Pour tester le service, il suffit de remplacer le **?** dans la balise ? par le nom d'un pays, par exemple *Spain*. Lancer le service en cliquant sur la flèche verte. Que constatez-vous?

Bien entendu, rien ne va vraiment se passer, car le service n'a pas encore été implémenté. Tout ce qu'on a fait, c'est de donner une spécification vide. La réponse qui s'affiche est un fichier XML affichant une erreur semblable à la suivante: *Operation [getCountry:\soapkit-config] not implemented*.

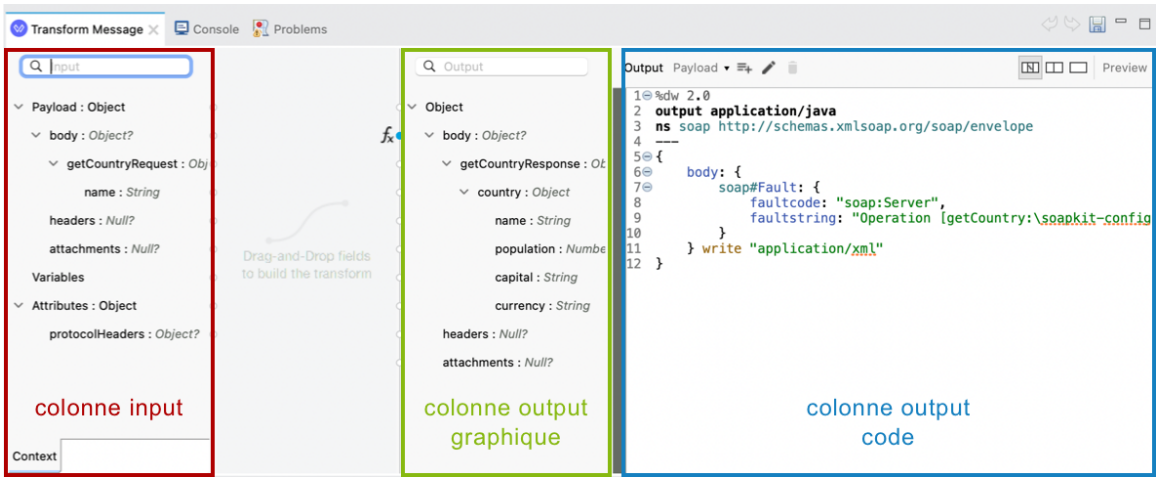
Nous allons montrer dans ce qui suit comment corriger cette erreur, pour afficher les informations d'un pays donné en paramètre.

3. Implémentation du service

Le deuxième rectangle dans la fenêtre principale de Anypoint permet de définir le comportement du service suite à la réception d'un SOAP Request. Il contient pour le moment un seul composant : *Transform Message*. En cliquant dessus, vous retrouverez dans l'output le message d'erreur qui vous a été retourné précédemment. Ceci veut dire que le comportement du service n'a pas été implémenté.

Pour le faire, nous allons suivre les étapes suivantes:

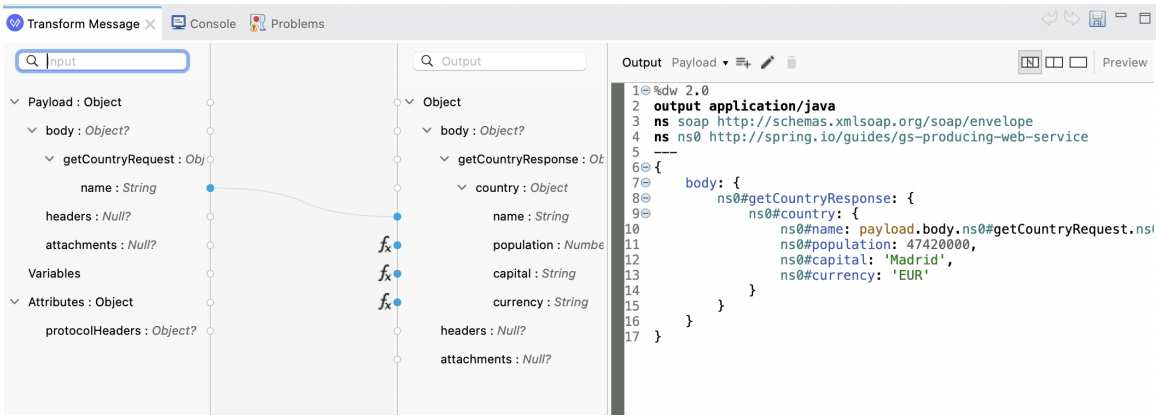
- Cliquer sur le composant *Transform Message*. Dans ses propriétés, trois colonnes sont affichées tel que représenté dans la figure suivante:




```
- Input:  le squelette de la SOAP Request
- Output (graphique): le squelette de la SOAP Response.
- Output (code): le code XML correspondant à la SOAP Response.
```

- Supprimer le bloc **Soap#Fault** du code XML.
- Glisser la variable **name** de l'Input vers la variable **name** de l'output graphique. Observez comment le code XML a également changé.
- Double cliquer sur le champs **population** dans la colonne dde l'output graphique. Le code approprié sera généré dans le XML. Remplacer la valeur **null** par la valeur de la population en Espagne (47420000 à ce jour). Faites de même pour indiquer la capitale (*Madrid*) et la monnaie (*EUR*).

Le résultat devra ressembler à ce qui suit:



Relancer le service si vous l'avez fermé, et observez le résultat sur SOAPUI. Cela a l'air de bien marcher!

Cela dit, le service donnera toujours le même résultat (population/capitale/monnaie), quelque soit l'input saisi dans pays. Nous allons essayer dans ce qui suit de le rendre un peu plus flexible, toujours sans écrire une ligne de code!

4. Implémentation d'une condition

Pour implémenter une condition dans notre service, qui donne un output différent selon l'entrée qui lui est donnée, nous allons suivre les étapes suivantes:

- Nous allons d'abord créer une variable *country*, qu'on va alimenter à partir de l'input. Pour cela:
  - Glisser le composant *Set Variable* de la palette juste avant *Transform Message*
  - Donner le nom *country* à la variable.
  - Dans le champ *Value*, cliquer sur le bouton  .
  - Écrire quelque chose dans le champ qui s'affiche (par exemple *country*)
  - Cliquer ensuite sur le bouton  pour mapper l'entrée à la variable.
  - Dans la fenêtre qui s'affiche, glisser l'élément *name* de la requête vers la chaine String (dans l'output graphique). La valeur que vous avez saisi sera écrasée par l'expression du nom en entrée.
  - Cliquer sur *Done*
- À partir de la palette, glisser le composant *Choice* avant *Transform Message*
- Glisser ensuite *Transform Message* dans le *When*
- Cliquer sur le *When* pour définir la condition.
  - Dans le champ *Expression*, cliquer sur le bouton  , puis taper **directement** la condition: `vars.country == 'Spain'`
- Pour ajouter une autre condition (un *else if*), glisser un nouveau *Transform Message* de la palette vers l'extrémité droite du box *Choice*. Refaire ensuite les opérations précédentes pour un autre pays, tel que la Tunisie.
- Dans le box *Default*, créer un message d'erreur qui ressemble à celui qu'on avait à la création du service, qui indique que le pays donné en entrée n'est pas pris en charge.

Lancer le service, et vérifier avec SOAPUI que toutes les conditions sont bien prises en compte.

> **Attention**
>
> Prenez des imprim-écrans dès que a marche, vous en aurez besoin dans le rapport!

Consultation d'une base de données

Nous allons montrer dans ce qui suit les étapes nécessaires pour configurer et faire appel à une base de données. Nous allons utiliser MySQL dans notre exemple.

1. Création et population de la base

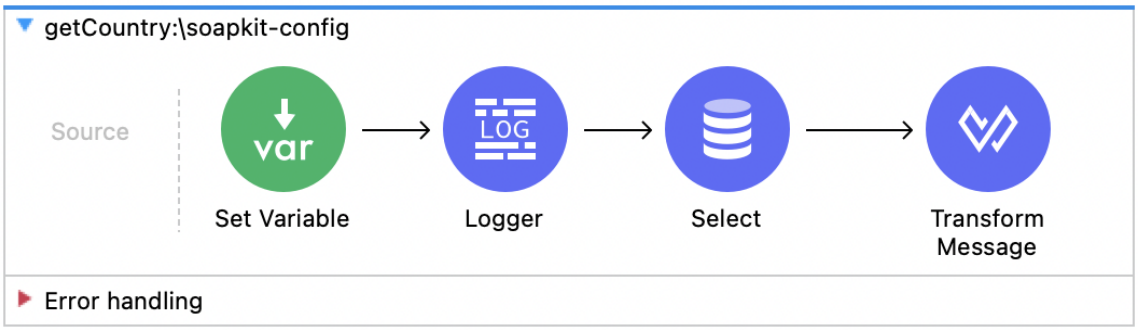
Pour cela:

- Créer une nouvelle base de données avec MySQL. On l'appellera Countries.
- Lancer le script SQL suivant pour créer la table countries populée avec plusieurs exemples de pays:

Consulter votre base pour voir son contenu et sa structure.

2. Modification du flux

Modifier le flux *getCountry* de façon à ce qu'il ressemble à l'image suivante:



- **Set Variable** permet de créer une variable *country* qui saisit l'entrée dans la SOAP Request.
- **Logger** permet d'afficher le contenu de la variable saisie sur la console, sous la forme: *"Pays: <nom_pays>"*.
- **Select** contient la requête de sélection de la base de données, filtrée par le pays donné en entrée.
- **Transform Message** retourne dans la SOAP Response, les informations du pays en entrée, extraites de la base de données.

Le résultat attendu ressemble au suivant:

Homework

Projet: Étape 1

Pour la séance de TP prochaine, vous devez réaliser l'étape 1 du projet, qui consiste à:

- Trouver le concept de votre entreprise (nom, logo, métier, et départements)
- Réaliser le tutoriel : Archisurance (voir les supports du projet)
- Commencer à réaliser les diagrammes de votre propre entreprise.

Last update: 2022-09-19