

STITUT SUPERIEUR DES ETUDES TECHNOLOGIQUES DE SFAX



Département Technologies de Informatique

Atelier SOA -03 Tester un service web avec Oxygen et SoapUI

Objectifs

Tester un Web Service développé avec l'API JAX-WS

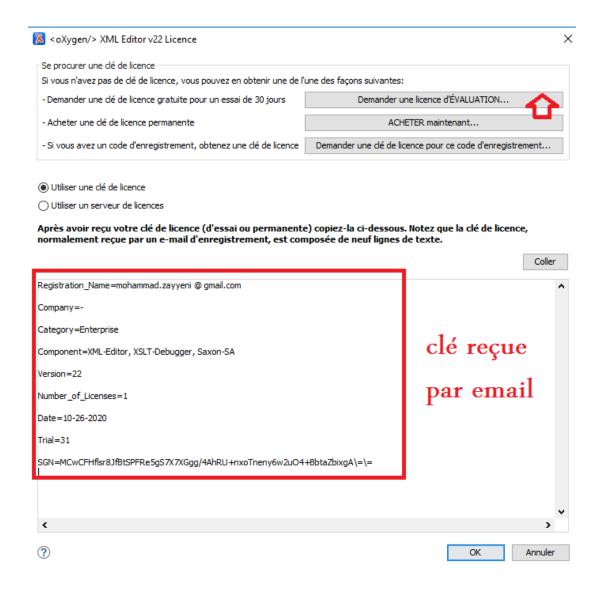
- Tester les méthodes du WS via OxygenXML et SoapUI
- Installer les outils de test
- Analyser un fichier wsdl
- Tester des opérations d'un WS

A. Tester les méthodes du SW via Oxygen XML

1. Accéder au lien suivant :

https://www.oxygenxml.com/xml_editor/download_oxygenxml_editor.html?os=Windows

- 2. Télécharger la version 64 bit
- 3. Installer l'outil « Oxygen »
- 4. Demander une clé d'évaluation. Cette clé est envoyée par émail :
 - a. Saisir un email
 - b. Saisir le pays
 - c. Recevoir un email contenant une clé composée de 9 lignes
 - d. Copier cette clé dans la zone de texte adéquate :

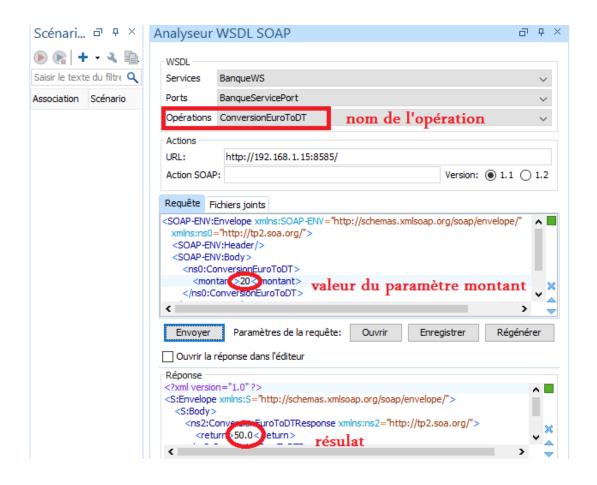


- 5. Aller au menu « Outils »
- 6. Choisir la commande « Analyseur WSDL SOAP »
- 7. Saisir l'URL du fichier wsdl généré lors du TP 02 :

http://192.168.1.6:8585/BanqueWS?wsdl

NB : Ne pas oublier d'exécuter le serveur nommé jax-wsserveur du TP_02 et vérifier l'adresse IP du serveur.

- 8. Lancer les méthodes du SW en passant les paramètres éventuelles
 - Voici un exemple d'invocation de l'opération
 « ConversionEuroToDT » en passant comme paramètre la valeur
 « 20 » et recevant comme valeur de retour « 50 » :



B. Tester les méthodes d'un service web via SOAPUI

- 9. Créer, dans le même projet «jax-ws-serveur » sous « src/main/java » un package « domaine ».
- 10. Créer, dans le package « domaine » une classe nommée « Produit » :

```
private int quantite;
private static final long serialVersionUID = 1L;
private java.util.Date dateAchat;
public java.util.Date getDateAchat() {
      return dateAchat;
public void setDateAchat(java.util.Date dateAchat) {
      this.dateAchat = dateAchat;
public Produit() {
      super();
public Long getId() {
      return this.id;
public void setId(Long id) {
      this.id = id;
public String getDesignation() {
      return this.designation;
}
public void setDesignation(String designation) {
      this.designation = designation;
public double getPrix() {
      return this.prix;
public void setPrix(double prix) {
      this.prix = prix;
}
public int getQuantite() {
      return this.quantite;
public void setQuantite(int quantite) {
      this.quantite = quantite;
}
```

```
11. Créer sous « src/main/java » un package « services ».
```

^{12.} Créer, dans le package « services », un service web «ProduitSWSoap ». Ce service web implémente l'interface suivante :

```
package services;
import java.util.List;
import domaine.Produit;

public interface ProduitSWSoapInterface
{
    public Produit addProduit(Produit p);
    public Produit updateProduit(Produit p);
    public void deleteProduit(Long id);
    public Produit getProduit(Long id);
    public List<Produit> getAllProduits();
}
```

Le service web « **ProduitSWSoap** » permet de gérer des produits dans un objet «**HashMap** » déclaré comme suit :

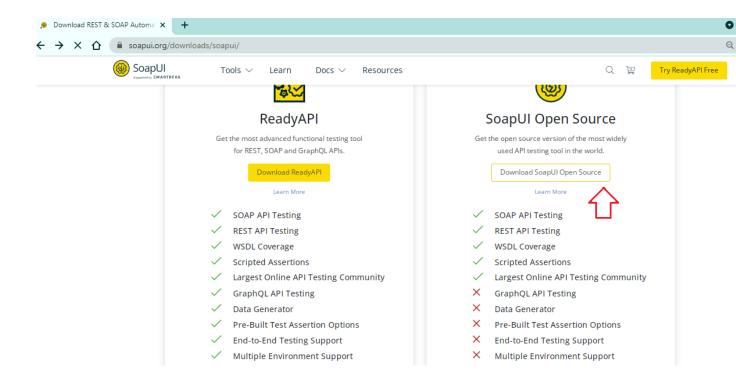
```
HashMap<Long, Produit> listeProduits =
new HashMap<Long, Produit >();
```

13. Si l'adresse IP de la machine hôte est «192.168.1.4 », publier le wsdl du service web sur l'adresse suivante :

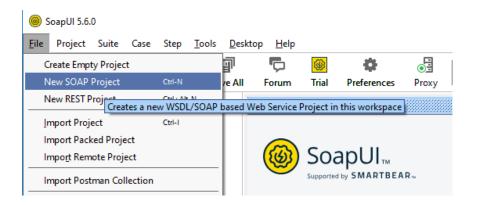
```
http://192.168.1.4:8585/GP/ProduitSWSoap?wsdl
```

14. Passer maintenant à tester le service web SOAP avec l'outil **SoapUI** accessible via le lien suivant :

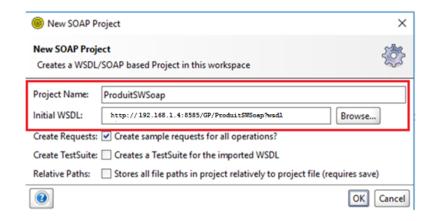
https://www.soapui.org/downloads/soapui/



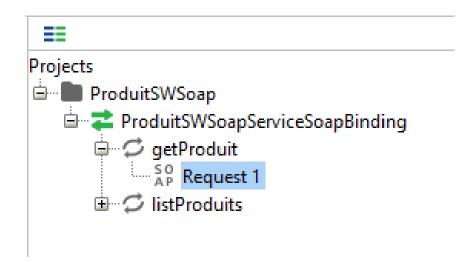
15. Lancer l'outil Soap UI et ouvrir un nouveau projet SOAP :



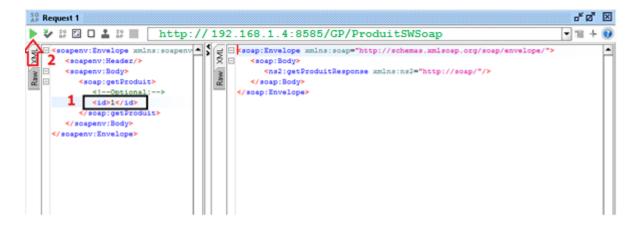
16. Spécifier l'adresse du fichier WSDL :



17. Remarquer la détection des méthodes du service web :



18. Tester la méthode « **getProduit** » pour récupérer le produit ayant l'id « 1 » :



19. Ici, le produit n'est pas existant, ajouter un produit puis relancer la requête :



20. Réaliser le test de toutes les méthodes du service web.