



Atelier SOA -02

Développer un service Web SOAP avec JAX-WS

Objectifs

Développer et déployer un Web Service en JAVA avec l'API JAX-WS

1. Créer le service Web

- Développer le WS (avec les annotations JAX-WS)
- Générer un fichier wsdl
- Générer les classes « skeleton »

2. Déployer ou publier le WS avec un conteneur JAX-WS

A. Créer le service WEB

1. Créer un simple projet JAVA de type **MAVEN** avec **IntelliJ IDEA**.
Choisir le modèle **maven-archetype-quickstart**

2. Spécifier les caractéristiques suivantes du projet:

- groupId : **org.soa.tp2**
- artifactId : **jax-ws-serveur**
- version : **1.0-SNAPSHOT**

Le projet est nommé **jax-ws-serveur** et enregistré sous :

D:\Dev\SOA\workspace\jax-ws-serveur

3. Supprimer les deux classes générées par **MAVEN** (App.java et AppTest.java)

4. Dans le package **org.soa.tp2** situé sous **src/main/java**, créer une classe du service Web nommée « **BanqueService** » :

```

package org.soa.tp2;

import jakarta.jws.WebMethod;
import jakarta.jws.WebParam;
import jakarta.jws.WebService;

@WebService(serviceName = "BanqueWS")
public class BanqueService
{
    @WebMethod (operationName="ConversionEuroToDT")
    public double conversion (@WebParam(name="montant")
double mt)
    {
        return mt*2.5;
    }
    @WebMethod
    public String test()
    {
        return "test";
    }
}

```

- Remarquer que le code présente des erreurs. Ajouter la dépendance de jakarata pour remédier au problème :

```

<dependency>
  <groupId>jakarta.xml.ws</groupId>
  <artifactId>jakarta.xml.ws-api</artifactId>
  <version>4.0.0</version>
</dependency>

```

5. Publier le WS à partir d'un serveur JAVA

- Créer ,dans le package **org.soa.tp2**, une classe nommée **ServeurJaxws** ayant le code suivant :

```

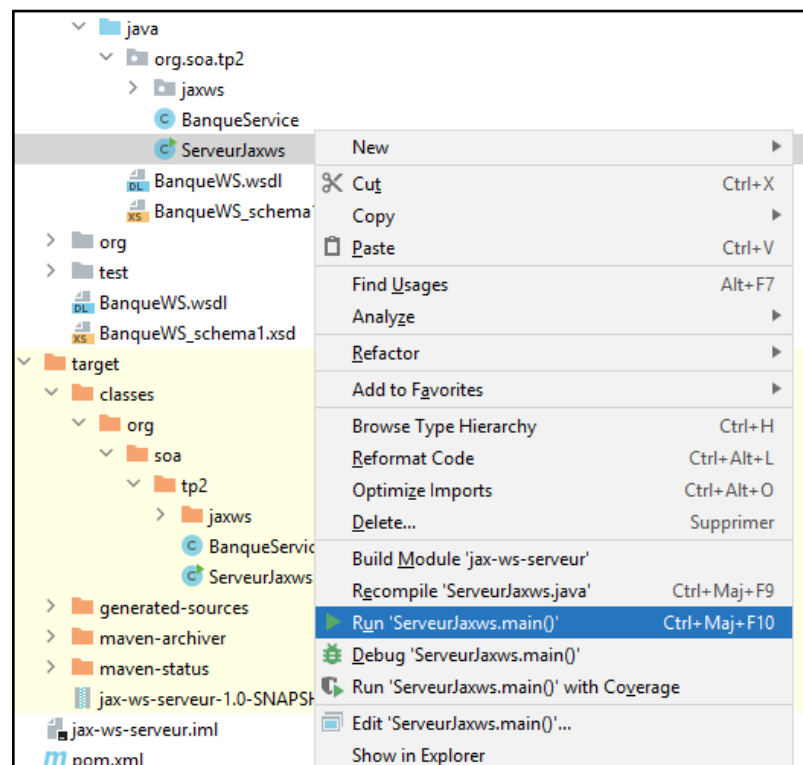
package org.soa.tp2;
import jakarta.xml.ws.Endpoint;

public class ServeurJaxws
{
    public static void main(String[] args)
    {
        String adresseIp="192.168.1.6";
        String url ="http://" + adresseIp + ":8585/";
        // publier le service web
        Endpoint.publish(url, new BanqueService());
        System.out.println(url);
    }
}

```

NB : Remplacer la valeur de variable **adresseIp** par la valeur de l'adresse IP de votre machine (utiliser la commande **ipconfig**)

- puis lancer l'exécution du serveur (As Java Application) :



- Remarquer au moment de l'exécution qu'une erreur est déclenchée :

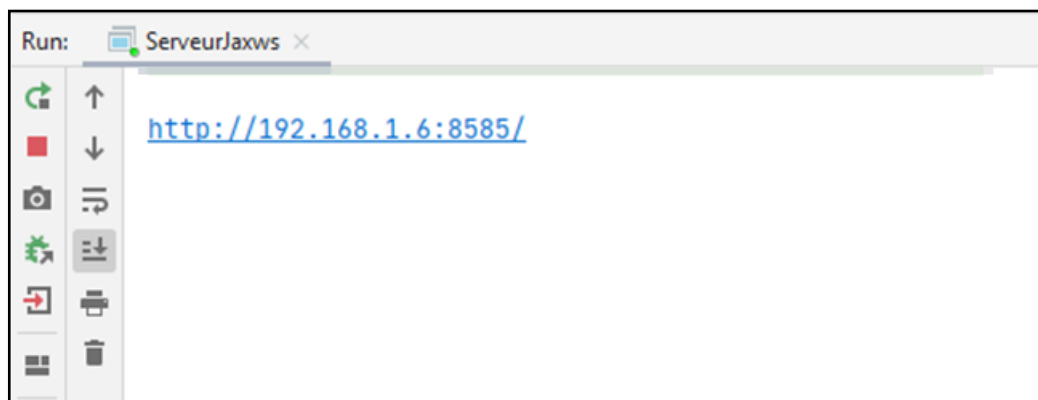
```
SEVERE: Class org.glassfish.hk2.osgiresourcelocator.ServiceLoader cannot be loaded
> java.lang.ClassNotFoundException Create breakpoint : org.glassfish.hk2.osgiresourcelocator.ServiceLoader <2 internal lines>
    at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:520)
    at java.base/java.lang.Class.forName0(Native Method)
    at java.base/java.lang.Class.forName(Class.java:375)
    at jakarta.xml.ws.spi.FactoryFinder.isOsgi(FactoryFinder.java:133)
    at jakarta.xml.ws.spi.FactoryFinder.find(FactoryFinder.java:98)
    at jakarta.xml.ws.spi.Provider.provider(Provider.java:64)
    at jakarta.xml.ws.Endpoint.publish(Endpoint.java:224)
    at org.soa.tp1.Serveur.<init>(Serveur.java:19)
    at org.soa.tp1.Serveur.main(Serveur.java:12)

Exception in thread "main" jakarta.xml.ws.WebServiceException Create breakpoint : Provider com.sun.xml.ws.spi.ProviderImpl not found
    at jakarta.xml.ws.spi.FactoryFinder$1.createException(FactoryFinder.java:35)
    at jakarta.xml.ws.spi.FactoryFinder$1.createException(FactoryFinder.java:32)
    at jakarta.xml.ws.spi.ServiceLoaderUtil.newInstance(ServiceLoaderUtil.java:76)
    at jakarta.xml.ws.spi.FactoryFinder.find(FactoryFinder.java:110)
    at jakarta.xml.ws.spi.Provider.provider(Provider.java:64)
```

- Pour **corriger** cette erreur, ajouter la dépendance suivante :

```
<dependency>
  <groupId>com.sun.xml.ws</groupId>
  <artifactId>rt</artifactId>
  <version>4.0.1</version>
</dependency>
```

- Re-exécuter le programme et vérifier , dans le volet « **run** », que le serveur est démarré et il se met en écoute dans le port **8585**:



- Interroger le serveur via un navigateur Web à travers l'URL suivante :

192.168.1.6:8585/BanqueWS?wsdl

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e94600d
<!-- Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e94600d
▼ <definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/2004/09/policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://tp2.soa.org/" name="BanqueService">
  ▼ <types>
    ▼ <xsd:schema>
      <xsd:import namespace="http://tp2.soa.org/" schemaLocation="http://192.168.1.6:8585/?xsd=1"/>
    </xsd:schema>
  </types>
  ▼ <message name="test">
    <part name="parameters" element="tns:test"/>
  </message>
  ▼ <message name="testResponse">
    <part name="parameters" element="tns:testResponse"/>
  </message>
  ▼ <message name="ConversionEuroToDT">
    <part name="parameters" element="tns:ConversionEuroToDT"/>
  </message>
  ▼ <message name="ConversionEuroToDTResponse">
    <part name="parameters" element="tns:ConversionEuroToDTResponse"/>
  </message>
  ▼ <portType name="BanqueService">
    ▼ <operation name="test">
      <input wsam:Action="http://tp2.soa.org/BanqueService/testRequest" message="tns:test"/>
      <output wsam:Action="http://tp2.soa.org/BanqueService/testResponse" message="tns:testResponse"/>
    </operation>
    ▼ <operation name="ConversionEuroToDT">
      <input wsam:Action="http://tp2.soa.org/BanqueService/ConversionEuroToDTRequest" message="tns:ConversionEuroToDT"/>
      <output wsam:Action="http://tp2.soa.org/BanqueService/ConversionEuroToDTResponse" message="tns:ConversionEuroToDTResponse"/>
    </operation>
  </portType>
  ▼ <binding name="BanqueServicePortBinding" type="tns:BanqueService">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
  </binding>
</definitions>
```