جامعة مولاي إسماعيل
UNIVERSITÉ MOULAY ISMAÏL

كلية العلوم
FACULTÉ DES SCIENCES

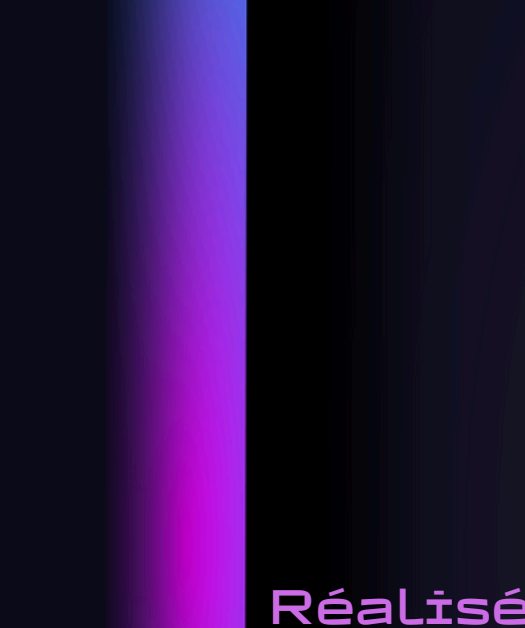
03-06-2024

RAPPORT TP5

WEB SERVICES SOAP WSDL

Réalisé par :
Bouchama Hajar

Demandé par :
Mohamed Youssfi



Master : Intelligence artificielle et analyse de données

Introduction

Ce rapport présente les travaux réalisés dans le cadre de l'Activité Pratique N°5 sur les Web services SOAP WSDL. L'objectif de cette activité est de mettre en pratique les concepts de Web services en utilisant les protocoles SOAP (Simple Object Access Protocol) et WSDL (Web Services Description Language).

En se basant sur le lien https://www.youtube.com/watch?v=WZi3s2bZNAE, nous avons pour but de créer et déployer un Web service offrant plusieurs fonctionnalités, à savoir la conversion d'un montant de l'euro en dirhams marocains (DH), la consultation d'un compte spécifique, et la consultation d'une liste de comptes.

Ce rapport détaillera les différentes étapes de la réalisation de ce projet, qui incluent :

Création du Web service : Implémentation des opérations permettant de convertir une devise, de consulter un compte, et de récupérer une liste de comptes.
Déploiement du Web service : Utilisation d'un serveur JaxWS pour héberger le service.
Analyse du WSDL : Consultation et analyse du fichier WSDL généré à l'aide d'un navigateur HTTP.

Test des opérations : Utilisation d'outils tels que SoapUI ou Oxygen pour tester les différentes opérations proposées par le Web service.
Création d'un client SOAP Java : Génération du stub à partir du WSDL et développement d'un client SOAP capable d'interagir avec le Web service.

Ces étapes permettent d'acquérir une compréhension pratique des mécanismes de création, de déploiement, et de consommation de Web services basés sur SOAP, tout en développant des compétences en manipulation des fichiers WSDL et en utilisation des outils de test de Web services. Le rapport fournira également une analyse des résultats obtenus et des défis rencontrés lors de la mise en œuvre du projet.

Enoncé !

En utilisant la démo de la dernière séance :
https://www.youtube.com/watch?v=WZi3s2bZNAE

1. Créer un Web service qui permet de :
   • Convertir un montant de l’euro en DH
   • Consulter un Compte
   • Consulter une Liste de comptes

2. Déployer le Web service avec un simple Serveur JaxWS

3. Consulter et analyser le WSDL avec un Browser HTTP

4. Tester les opérations du web service avec un outil comme SoapUI ou Oxygen

5. Créer un Client SOAP Java
   - Générer le Stub à partir du WSDL
   - Créer un client SOAP pour le web service

1. Créer un Web service qui permet de :

Après avoir crééer le package WS dans ce dernier on créé la class Compte qui porte plusier attributs et methodes comme suite :

```java
package ws;
import java.util.Date;
6 usages
public class Compte {
    3 usages
    private int code;
    3 usages
    private double solde;
    3 usages
    private Date dateCreation;
    no usages
    public Compte() {
    }
    4 usages
    public Compte(int code, double solde, Date dateCreation) {
        this.code = code;
        this.solde = solde;
        this.dateCreation = dateCreation;
```

```java
    public int getCode() {
        return code;
    }
    no usages
    public void setCode(int code) {
        this.code = code;
    }
    no usages
    public double getSolde() {
        return solde;
    }

    public void setSolde(double solde) {
        this.solde = solde;
    }
    public void setDateCreation(Date dateCreation) {
        this.dateCreation = dateCreation;
    }
}
    public Date getDateCreation() {
        return dateCreation;
    }
```

Dans le fichier pom.xml on ajoutent les dependences suivantes :

```xml
<dependencies>
    <dependency>
        <groupId>com.sun.xml.ws</groupId>
        <artifactId>jaxws-ri</artifactId>
        <version>4.0.2</version>
        <type>pom</type>
    </dependency>
</dependencies>
```

- **Convertir un montant de l'euro en DH**
- **Consulter un Compte**
- **Consulter une Liste de comptes**

Dans le package WS on créé le service BanqueService :

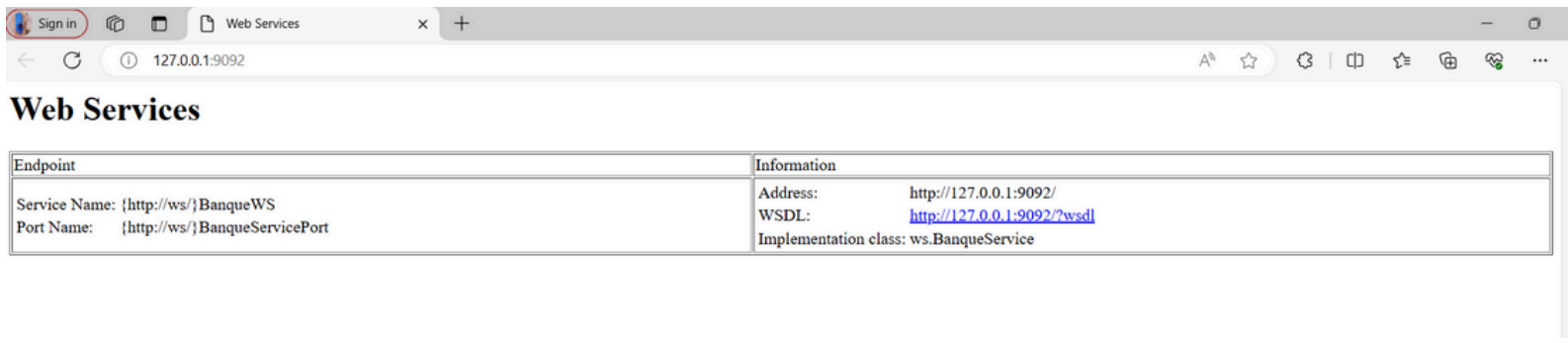
```java
@WebService(serviceName = "BanqueWS")
public class BanqueService {
    no usages
    @WebMethod(operationName = "conversionEutoToDH")
    public double conversion(@WebParam(name = "montant") double mt){
        return mt*11.3;
    }
    no usages
    @WebMethod
    public Compte getCompte(@WebParam(name = "code") int code){
        return new Compte(code, solde: Math.random()*80000, new Date());
    }
    no usages
    @WebMethod
    public List<Compte> listComptes(){
        return List.of(
                new Compte( code: 1, solde: Math.random()*80000, new Date()),
                new Compte( code: 2, solde: Math.random()*80000, new Date()),
                new Compte( code: 3, solde: Math.random()*80000, new Date())
        );
    }
}
```

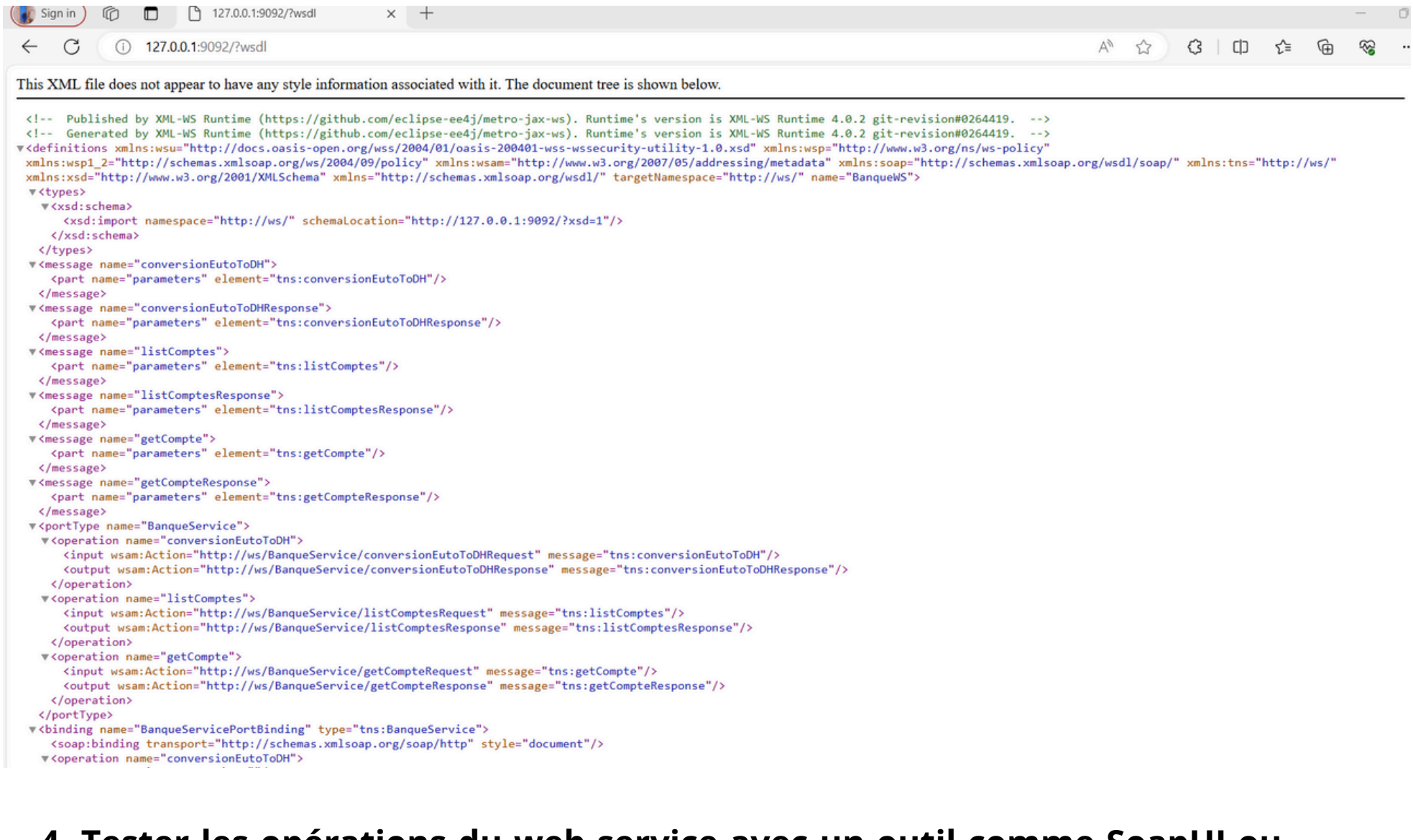
2. Déployer le Web service avec un simple Serveur JaxWS :

Pour y fair on va creer la cla ServerJWS !:

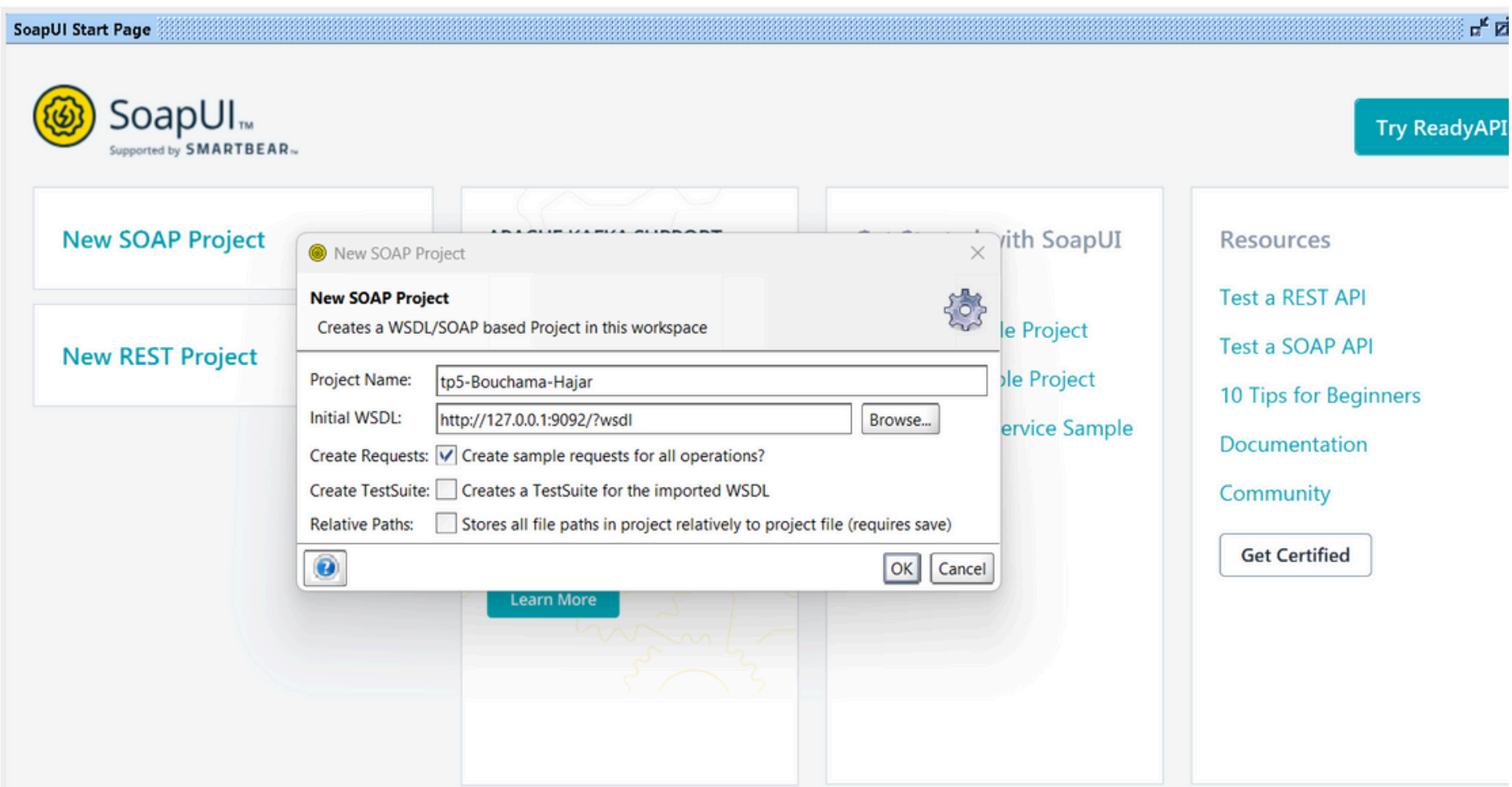
```java
import jakarta.xml.ws.Endpoint;
import ws.BanqueService;

public class ServerJWS {
    public static void main(String[] args) {
        String url = "http://127.0.0.1:9091/";
        Endpoint.publish(url, new BanqueService());
        System.out.println("Web service deployed on " + url);
    }
}
```

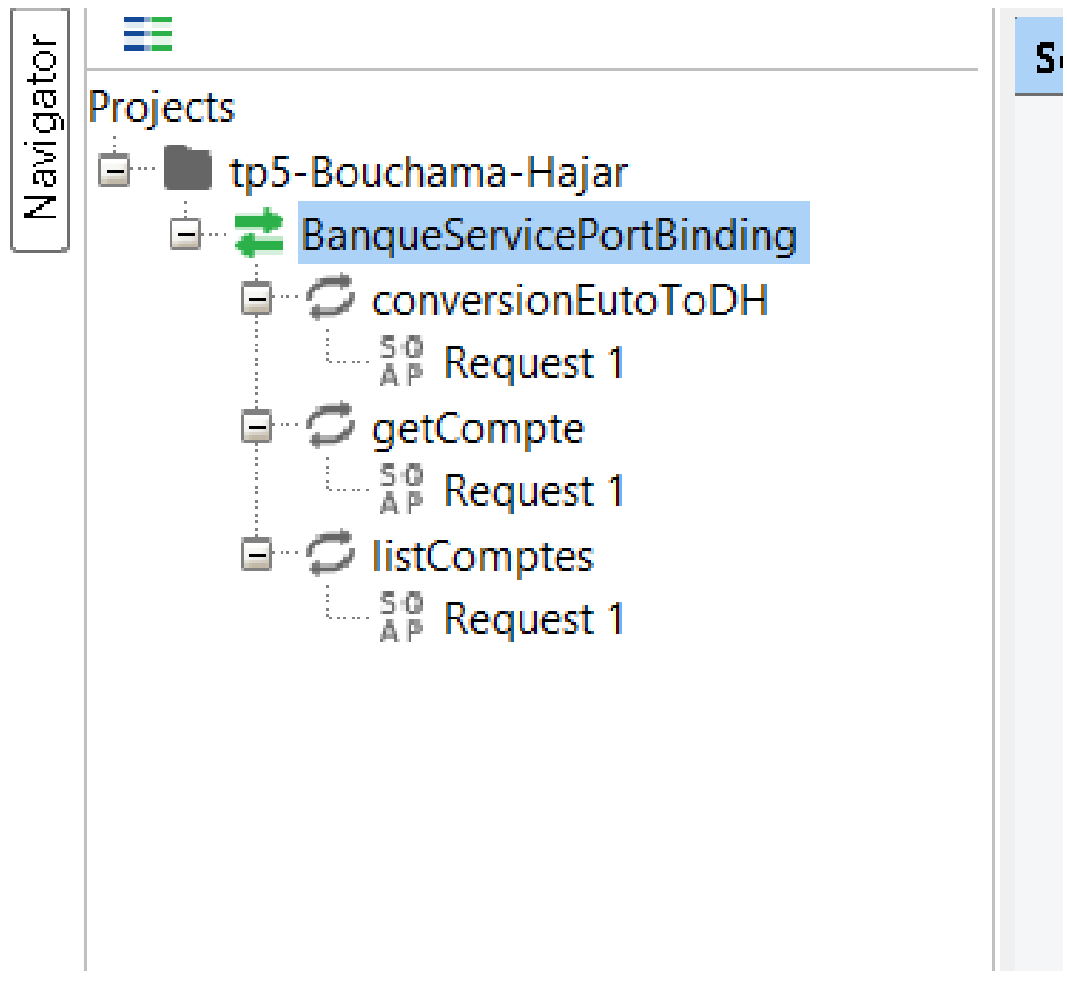
3. Consulter et analyser le WSDL avec un Browser HTTP :

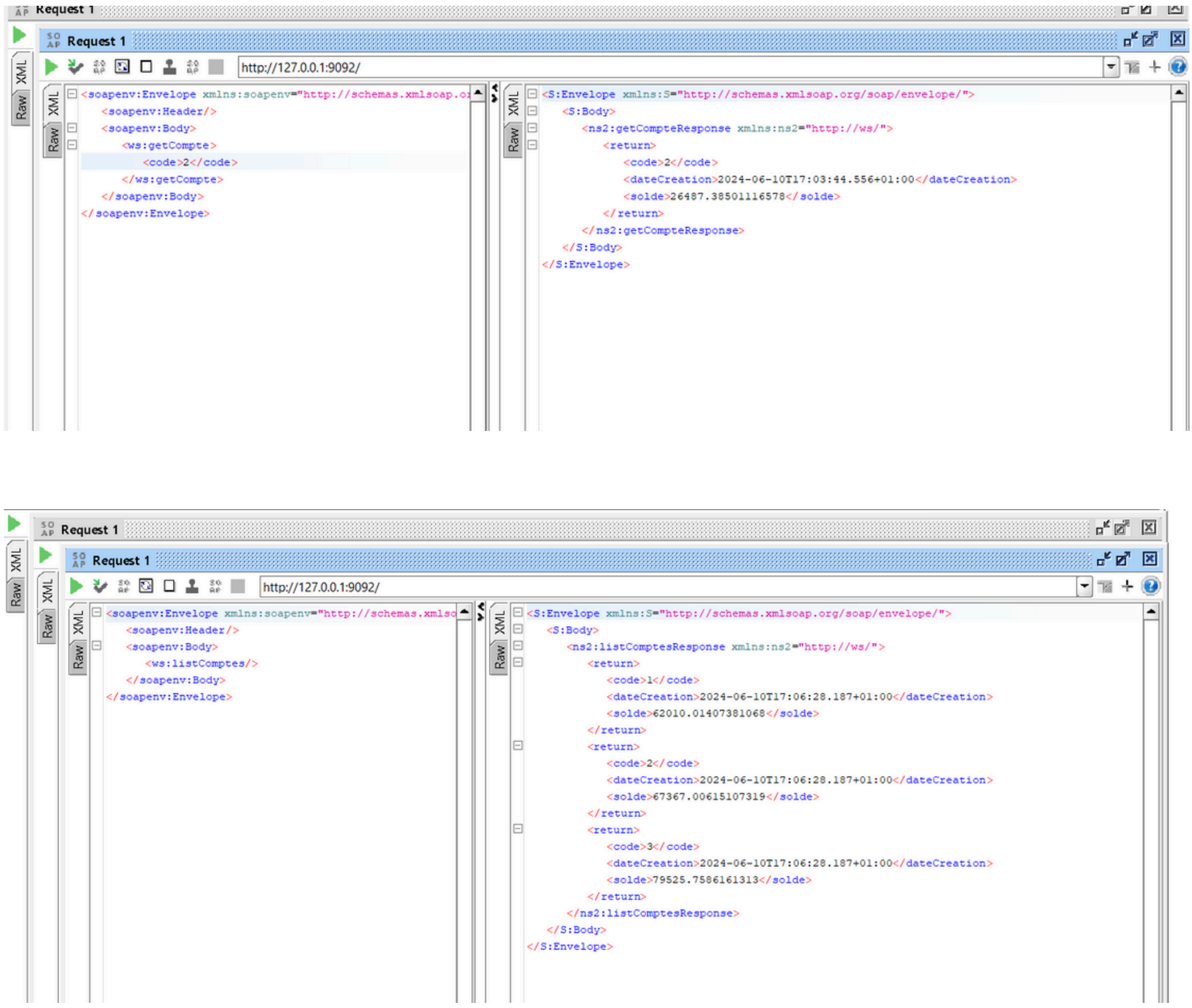




4. Tester les opérations du web service avec un outil comme SoapUI ou Oxygen





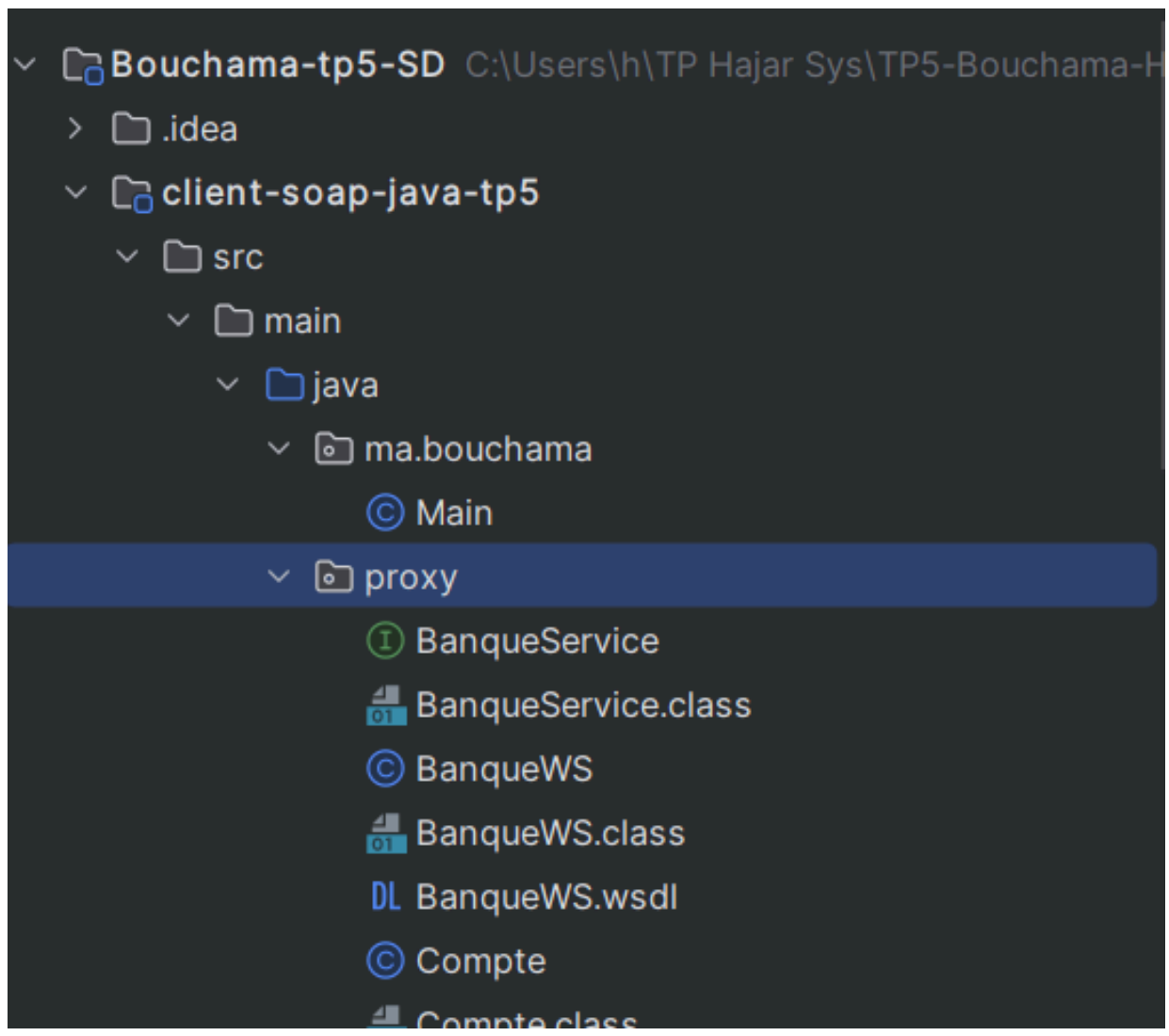


5. Créer un Client SOAP Java

Tout d”abord on a ceer un nouveau prijet sous le non Client-soap-java et on suite on a ajouter les dependaneces dejaxws dans le fichier pom :

```
<dependencies>
    <dependency>
        <groupId>com.sun.xml.ws</groupId>
        <artifactId>jaxws-ri</artifactId>
        <version>4.0.2</version>
        <type>pom</type>
    </dependency>
</dependencies>

/project>
```

Générer le Stub à partir du WSDL



Créer un client SOAP pour le web service

```java
package ma.bouchama;

import proxy.BanqueService;
import proxy.BanqueWS;

public class Main {
    public static void main(String[] args) {

        BanqueService proxy = new BanqueWS().getBanqueServicePort();
        System.out.println(proxy.conversionEutoToDH( montant: 90.0));

    }
}
```

SD) Compte.java BanqueService.java ServerJWS.java pom

```java
package ma.bouchama;

import proxy.BanqueService;
import proxy.BanqueWS;
import proxy.Compte;

public class Main {
    public static void main(String[] args) {

        BanqueService proxy = new BanqueWS().getBanqueServicePort()
        System.out.println(proxy.conversionEutoToDH( montant: 90.0));

        System.out.println("----------------------");
        Compte compte = proxy.getCompte( code: 4);
        System.out.println(compte.getCode());
        System.out.println(compte.getSolde());
        System.out.println(compte.getDateCreation());

        System.out.println("----------------------");

        proxy.listComptes().forEach(cp -> {
            System.out.println("----------------------");
            System.out.println(cp.getCode());
            System.out.println(cp.getSolde());
            System.out.println(cp.getDateCreation());
        });

    }
```

```
"C:\Program Files\Java\jdk-17\bin\java.exe" ...
1017.0000000000001
----------------------
4
72278.43484387691
2024-06-13T20:03:39.600+01:00
----------------------
----------------------
1
51756.73292231988
2024-06-13T20:03:39.613+01:00
----------------------
2
21372.191159617974
2024-06-13T20:03:39.613+01:00
----------------------
3
10005.00758055967
```