



MASTER SYSTÈMES DISTRIBUÉES ET INTELLIGENCE ARTIFICIELLE

Bases de Web Services

SOAP - WSDL - UDDI

Réalisé par :

Soufiane MOUHTARAM

Enseignant :

Pr. Mohamed YOUSSEFI

21 mars 2023

TABLE DES MATIÈRES

Table des matières	1
Table des figures	2
Introduction	3
1 Chapter1	4
1.1 Creation de Web Services	4
1.1.1 BanqueWS	4
1.1.2 Compte	5
1.2 Serveur JaxWS	5
1.2.1 Serveur	6
2 Chapter2	7
2.1 Tester Web Service	7
2.1.1 WSDL	7
2.2 Tester les fonctions de web services	9
2.2.1 Conversion	9
2.2.2 getCompte	10
2.2.3 getCompteS	11
2.3 Les clients	12
2.3.1 Client - JAVA	12
2.3.2 Client - .Net	13
2.3.3 Client - .Python	14

TABLE DES FIGURES

1.1	web service base sur SOAP	4
1.2	Figure 1.2 la classe Compte	5
1.3	Classe Serveur	6
2.1	WSDL	8
2.2	meth-ode Conversion	9
2.3	meth-ode getCompte	10
2.4	meth-ode getCompteS	11
2.5	client java	12
2.6	client C	13
2.7	client Python	14

Un Web Service est une application qui permet d'échanger des données avec d'autres applications web. Même si ces dernières sont construites dans des langages de programmation différents. Parmi les Web Services les plus connus on peut citer SOAP, REST ou HTTP . SOAP est un protocole d'échange d'information structurée dans l'implémentation de services web bâti sur XML 1. WSDL (Web Service Description Language) est un langage XML permettant la description complète d'un service Web 2. Il sert à décrire le protocole de communication, le format de messages requis pour communiquer avec ce service, les méthodes que le client peut invoquer et la localisation du service

CHAPITRE 1

CHAPTER1

1.1 Creation de Web Services

1.1.1 BanqueWS

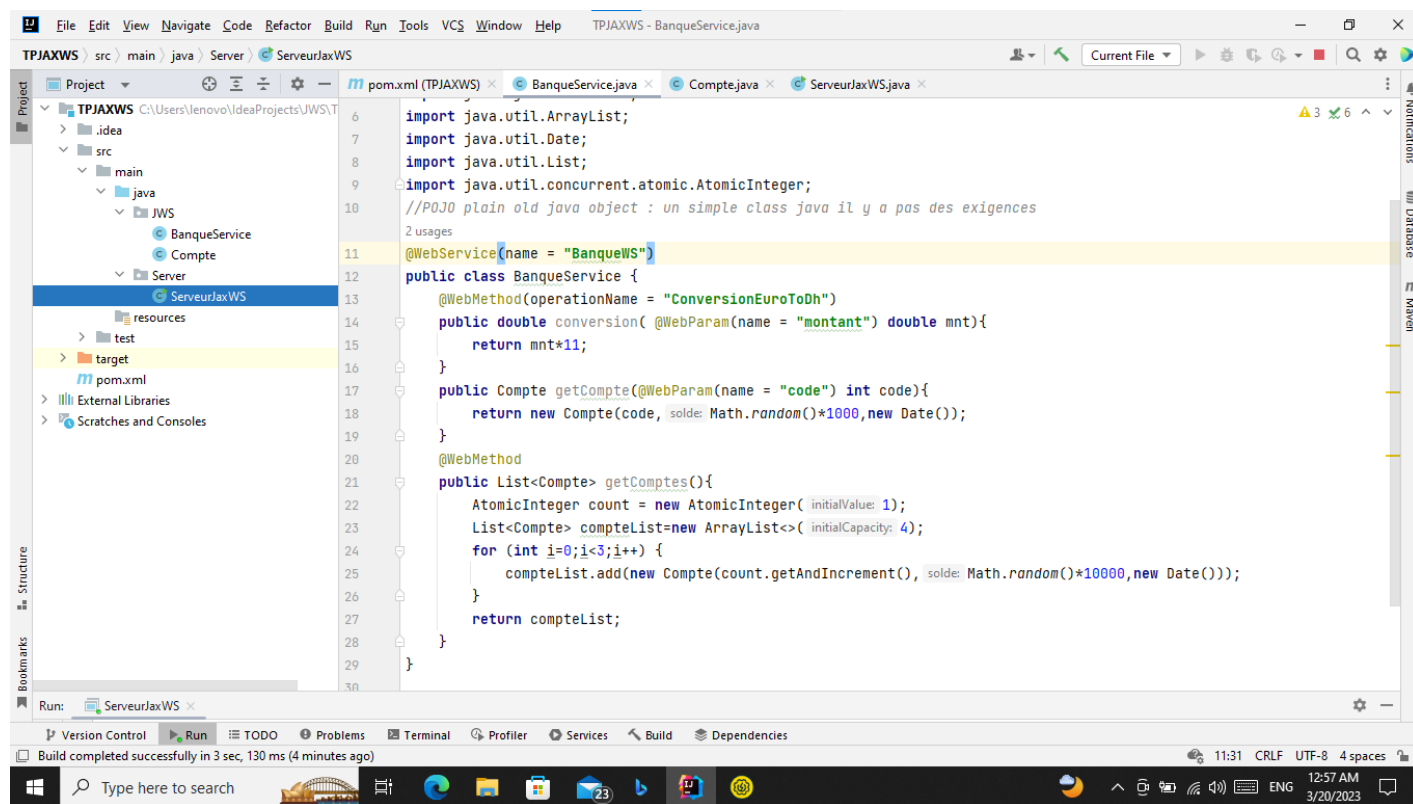


FIGURE 1.1 – web service base sur SOAP

1.1.2 Compte

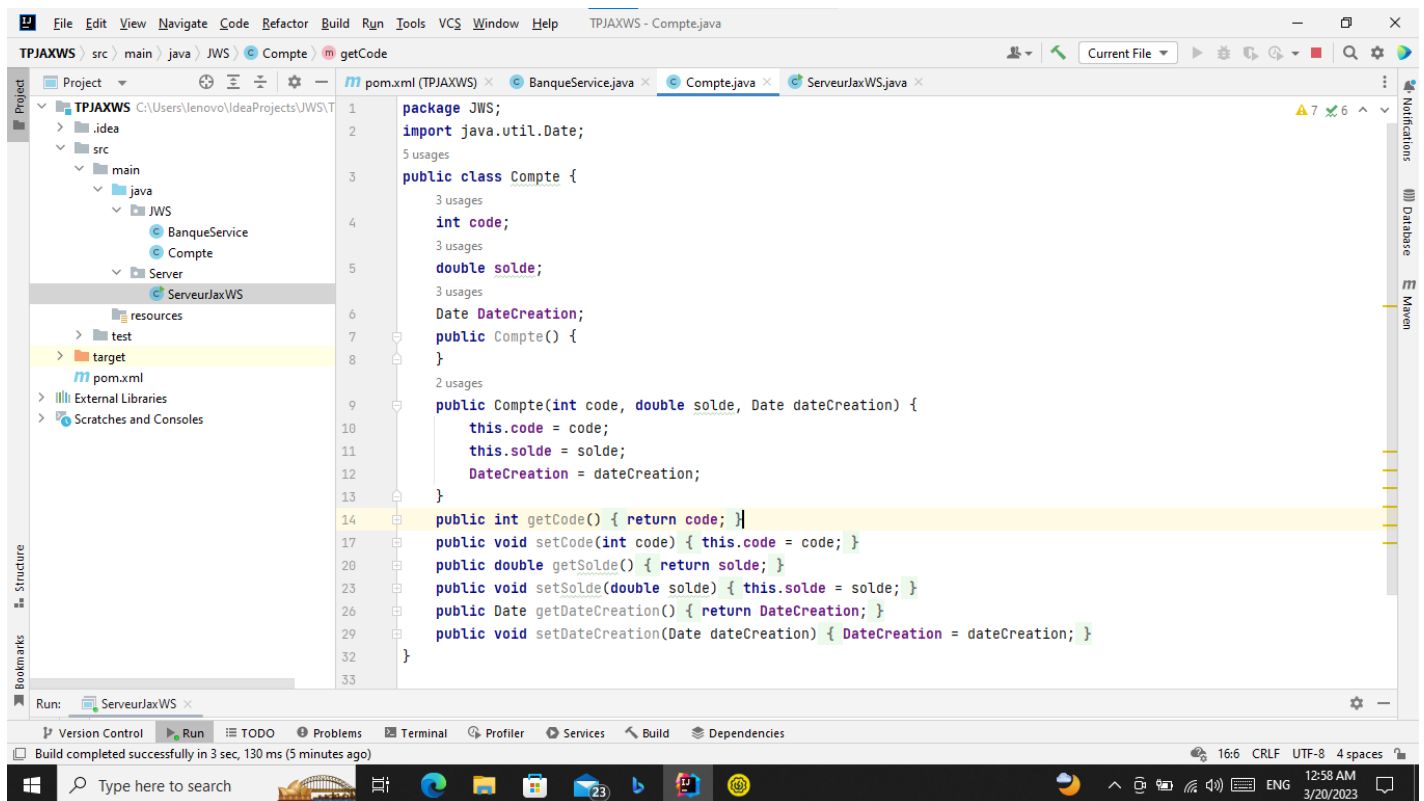


FIGURE 1.2 – Figure 1.2 la classe Compte

1.2 Serveur JaxWS

JAX-WS (Java API for XML-Based Web Services) est le modèle de programmation de services Web de nouvelle génération qui vient compléter la base fournie par le modèle de programmation JAX-RPC (Java API for XML-based RPC) 1. Il simplifie le développement de services Web et de clients grâce à une plus grande indépendance de la plateforme pour les applications Java 1. Un serveur JAX-WS est donc un serveur qui utilise ce modèle pour fournir des services Web.

1.2.1 Serveur

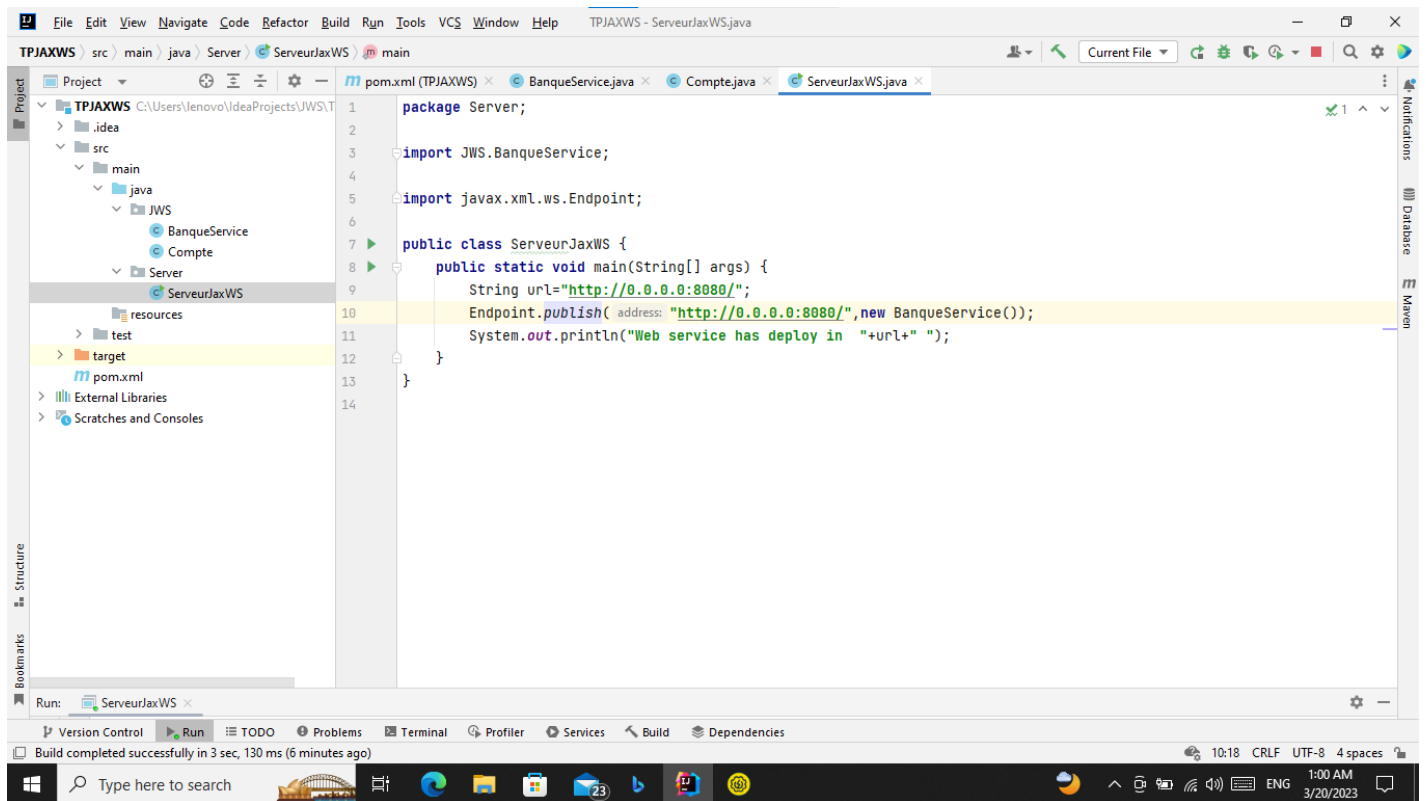


FIGURE 1.3 – Classe Serveur

2.1 Tester Web Service

Pour tester le web service il y a beaucoup des outils pour le tester mais pour nous on va tester avec SoapUI.

2.1.1 WSDL

On va Ouvrir un web Browser et tapez `http ://localhost :8080/BanqueWS?wsdl` pour generer le WSDL qui permet de decire un interface de Web Service.

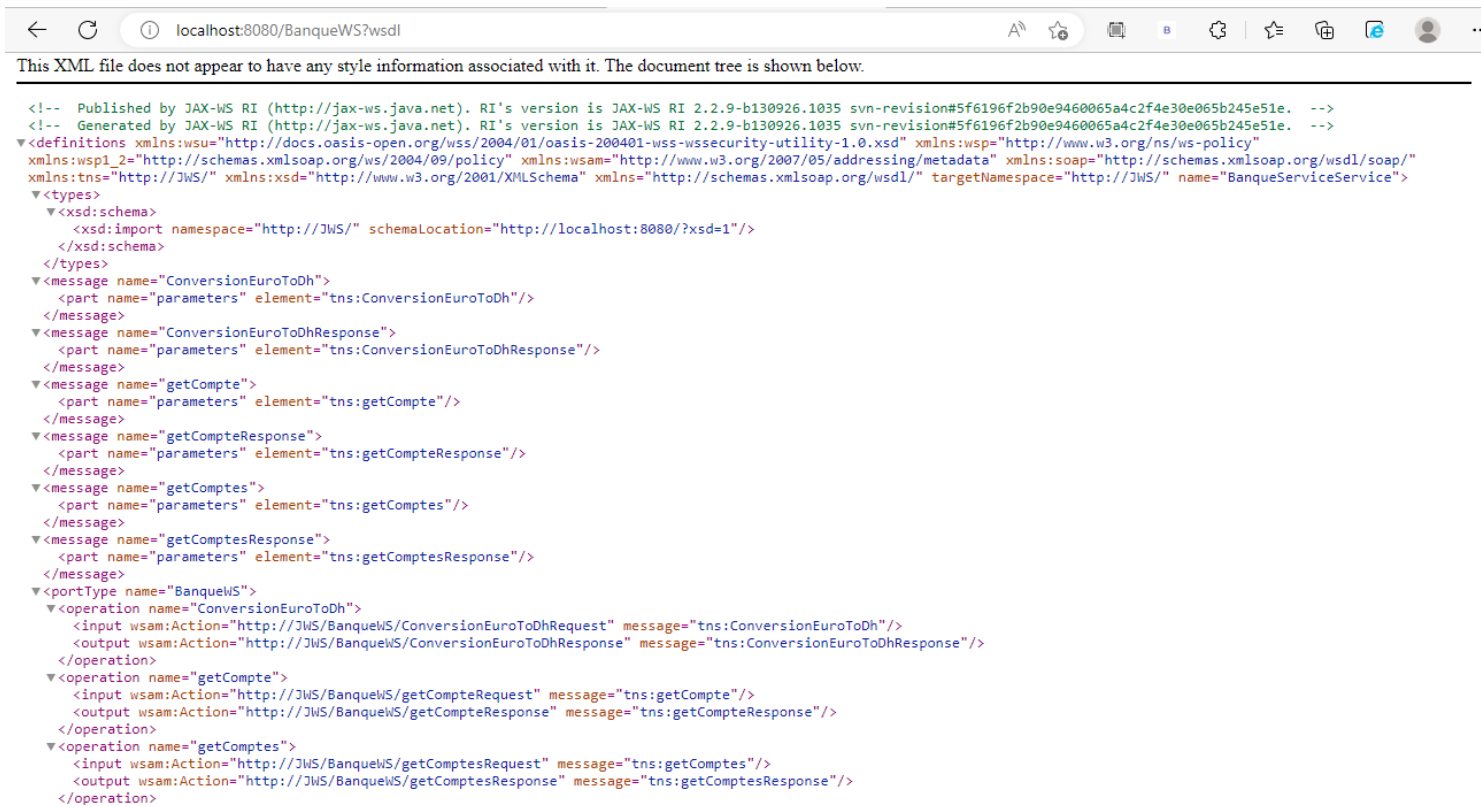


FIGURE 2.1 – WSDL

2.2 Tester les fonctions de web services

2.2.1 Conversion

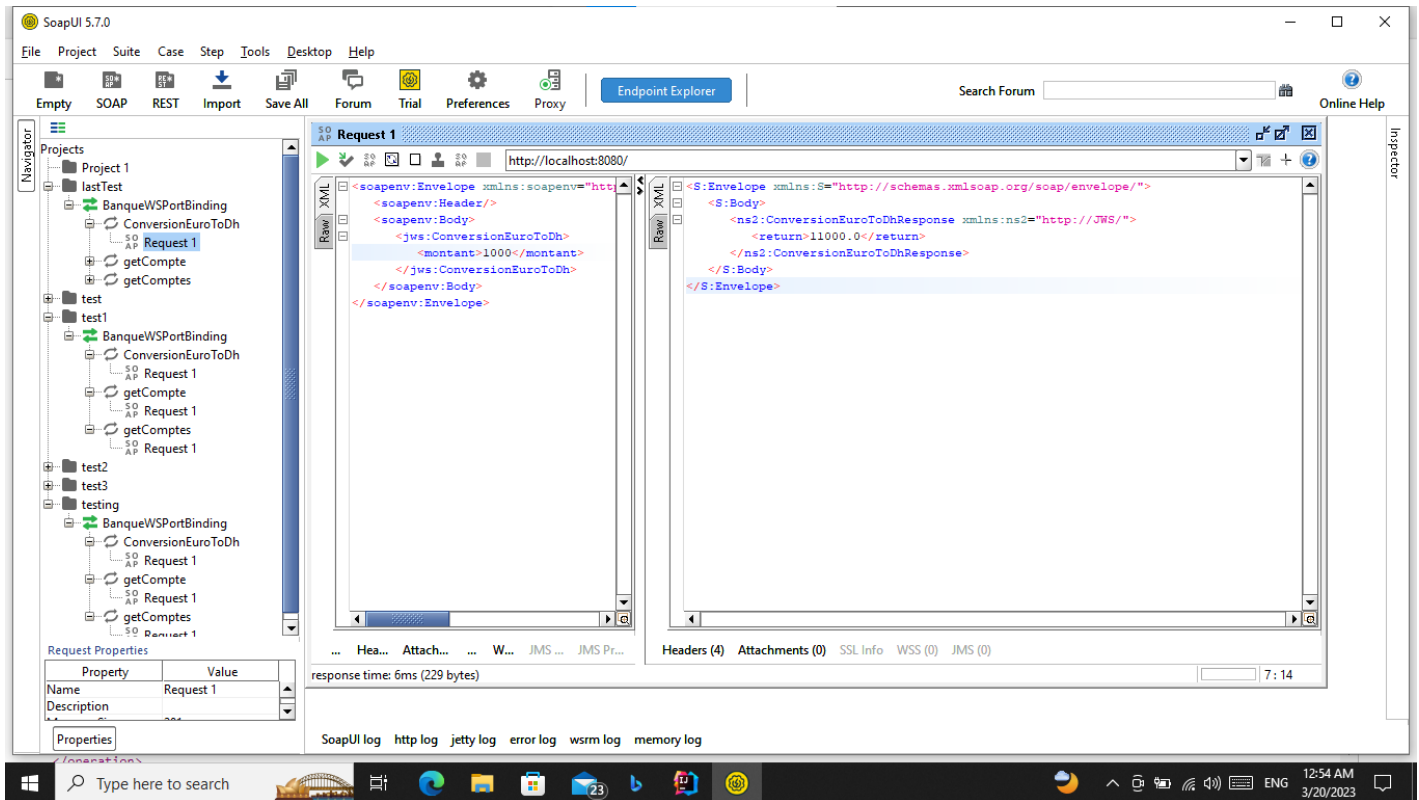


FIGURE 2.2 – meth-ode Conversion

2.2.2 getCompte

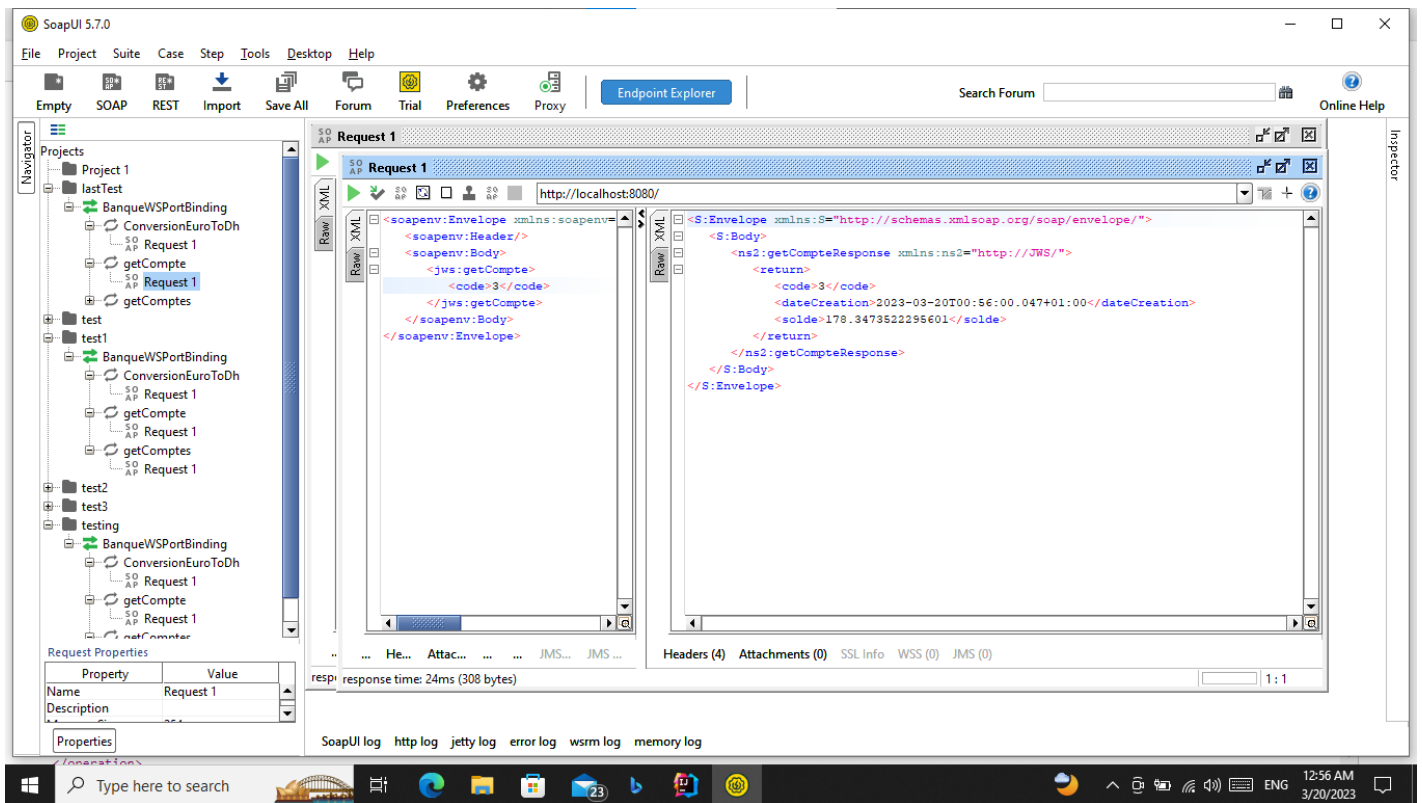


FIGURE 2.3 – meth-ode getCompte

2.2.3 getCompteS

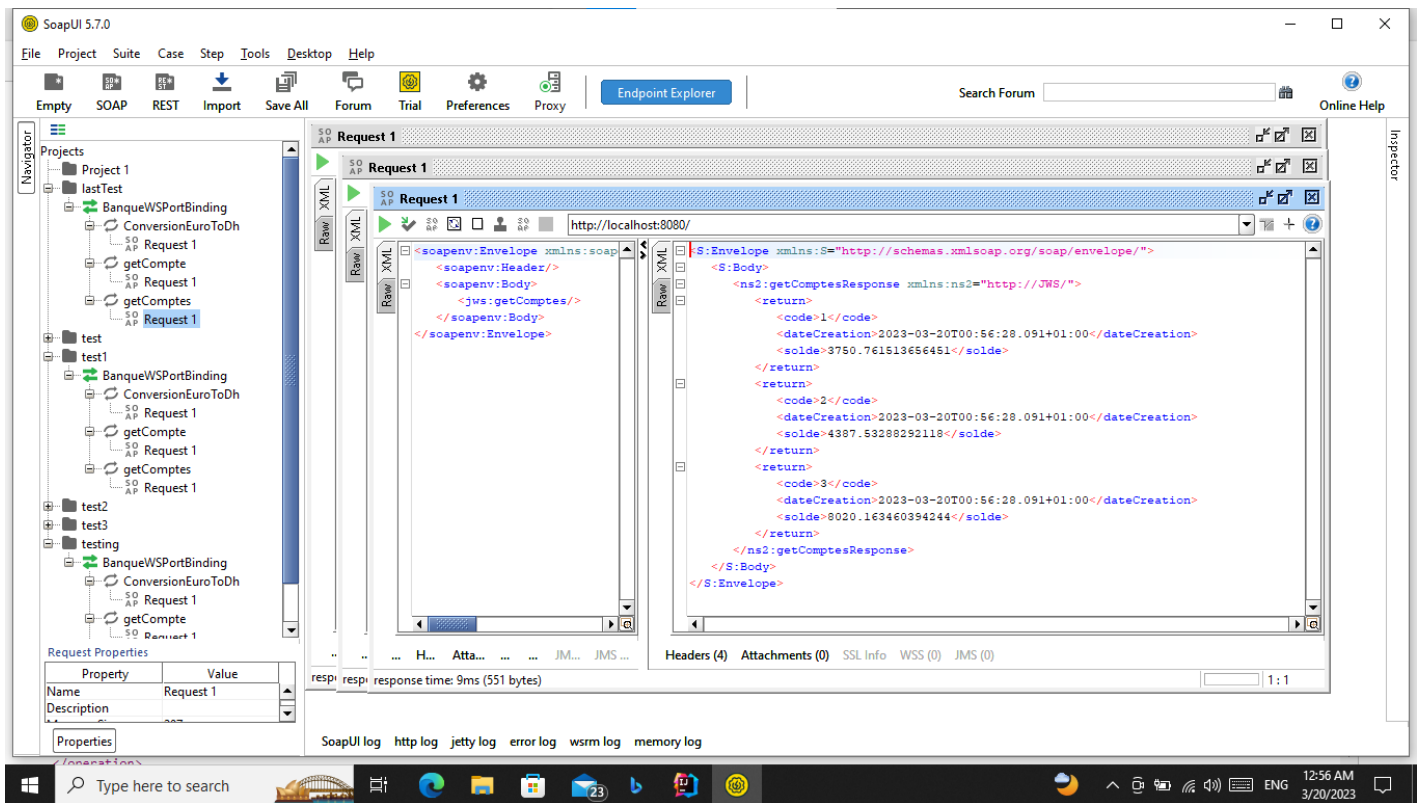


FIGURE 2.4 – meth-ode getCompteS

2.3 Les clients

2.3.1 Client - JAVA

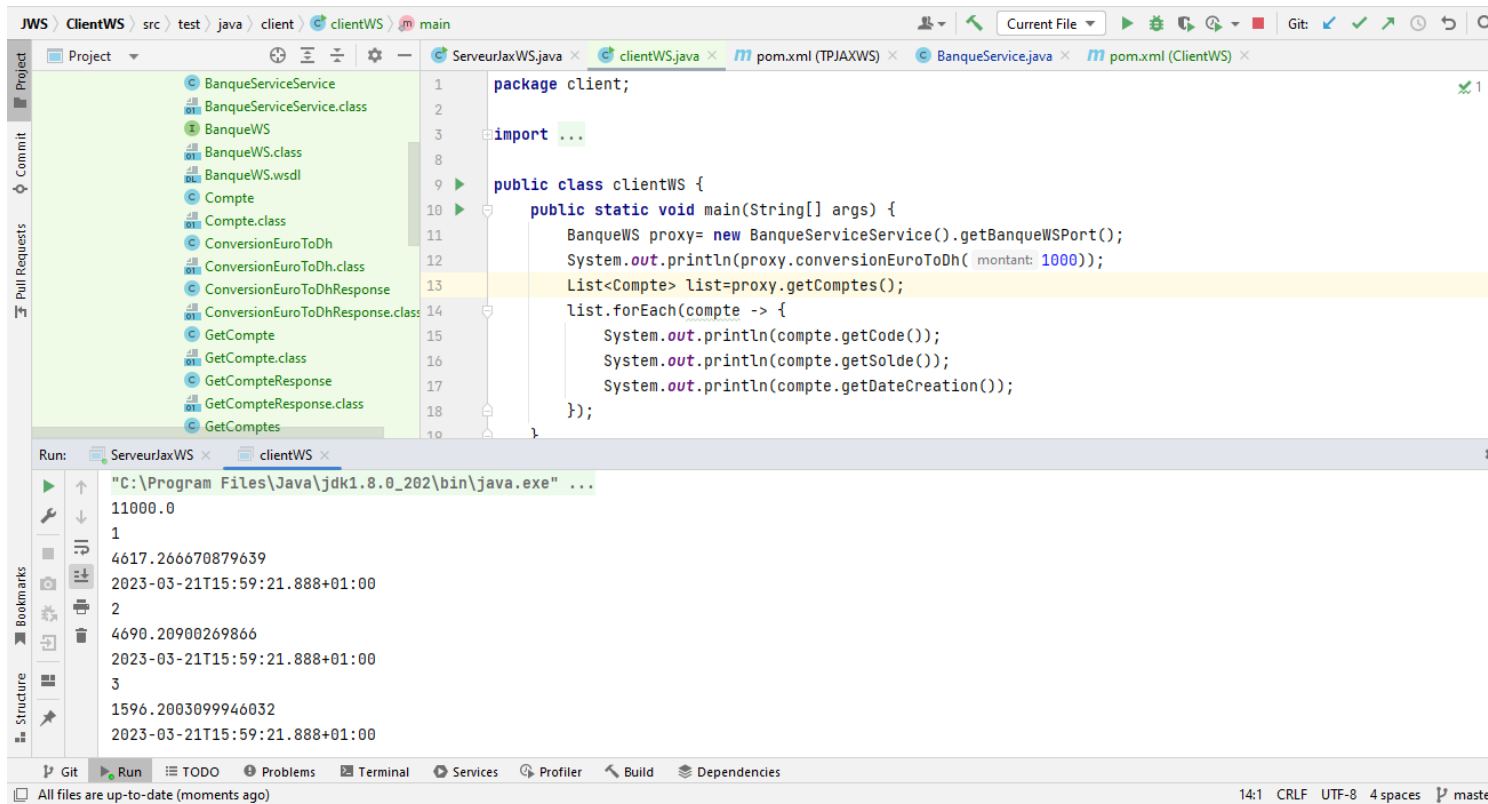
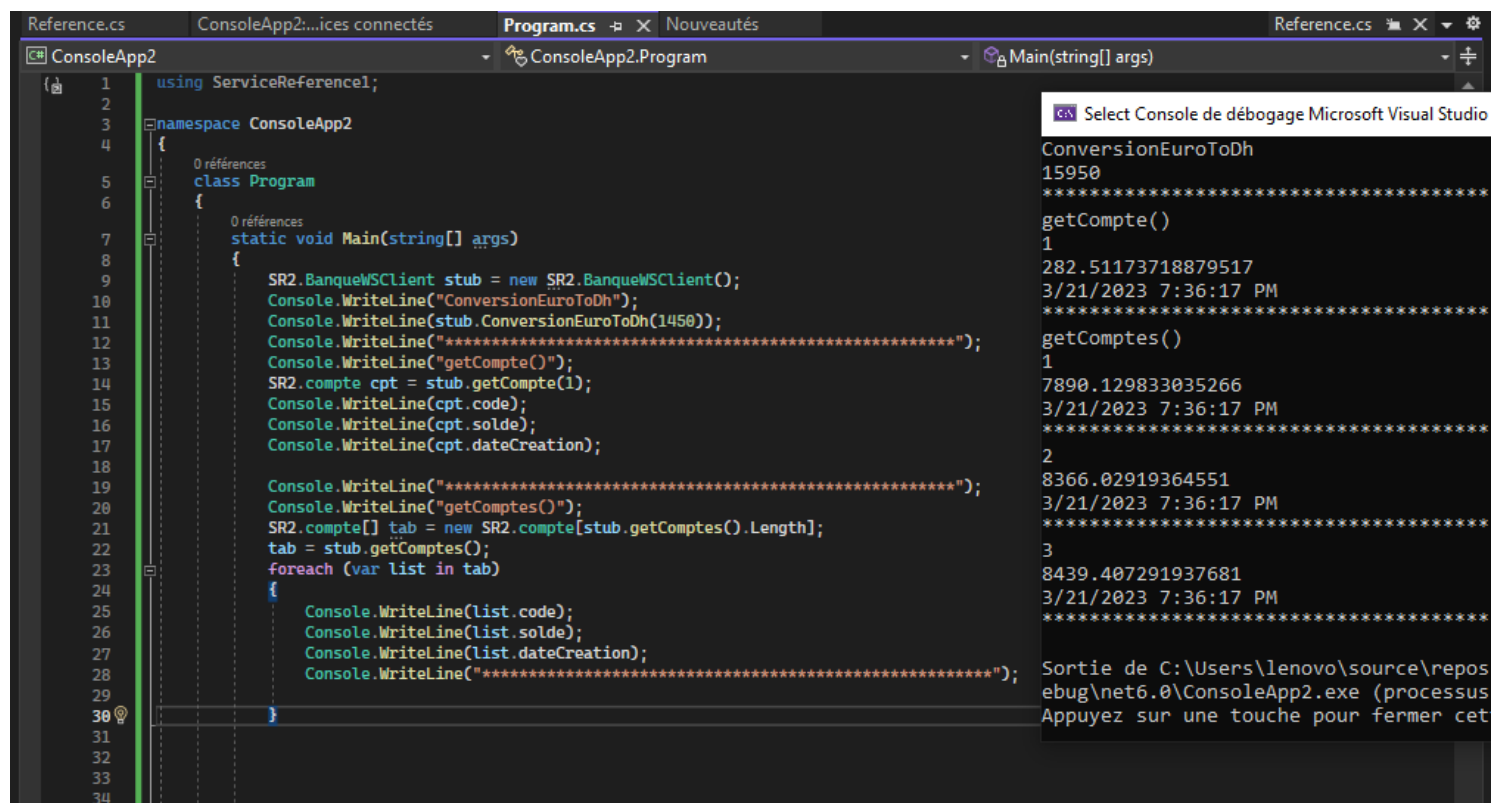


FIGURE 2.5 – client java

2.3.2 Client - .Net



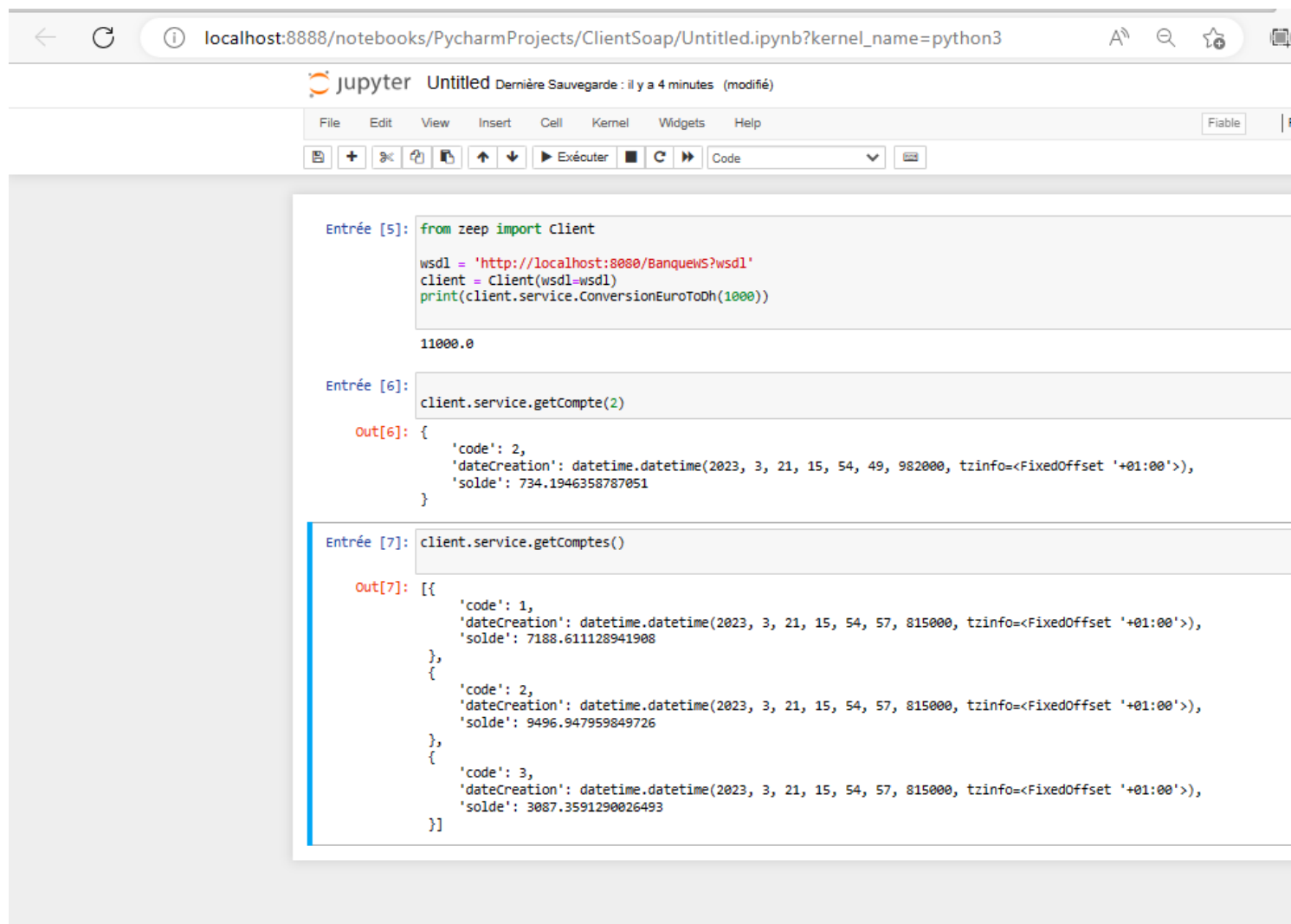
```
1 using ServiceReference1;
2
3 namespace ConsoleApp2
4 {
5     0 références
6     class Program
7     {
8         0 références
9         static void Main(string[] args)
10        {
11            SR2.BanqueWSClient stub = new SR2.BanqueWSClient();
12            Console.WriteLine("ConversionEuroToDh");
13            Console.WriteLine(stub.ConversionEuroToDh(1450));
14            Console.WriteLine("*****");
15            Console.WriteLine("getCompte()");
16            SR2.compte cpt = stub.getCompte(1);
17            Console.WriteLine(cpt.code);
18            Console.WriteLine(cpt.solde);
19            Console.WriteLine(cpt.dateCreation);
20
21            Console.WriteLine("*****");
22            Console.WriteLine("getComptes()");
23            SR2.compte[] tab = new SR2.compte[stub.getComptes().Length];
24            tab = stub.getComptes();
25            foreach (var list in tab)
26            {
27                Console.WriteLine(list.code);
28                Console.WriteLine(list.solde);
29                Console.WriteLine(list.dateCreation);
30            }
31            Console.WriteLine("*****");
32        }
33    }
34}
```

Select Console de débogage Microsoft Visual Studio

```
ConversionEuroToDh
15950
*****
getCompte()
1
282.51173718879517
3/21/2023 7:36:17 PM
*****
getComptes()
1
7890.129833035266
3/21/2023 7:36:17 PM
*****
2
8366.02919364551
3/21/2023 7:36:17 PM
*****
3
8439.407291937681
3/21/2023 7:36:17 PM
*****
Sortie de C:\Users\lenovo\source\repos
ebug\net6.0\ConsoleApp2.exe (processus
Appuyez sur une touche pour fermer cet
```

FIGURE 2.6 – client C

2.3.3 Client - .Python



The screenshot shows a Jupyter Notebook titled 'Untitled' with the URL 'localhost:8888/notebooks/PycharmProjects/ClientSoap/Untitled.ipynb?kernel_name=python3'. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, execution, and search. The notebook contains three code cells:

```
Entrée [5]: from zeep import Client
            wsdl = 'http://localhost:8080/BanqueWS?wsdl'
            client = Client(wsdl=wsdl)
            print(client.service.ConversionEuroToDh(1000))

            11000.0
```

```
Entrée [6]: client.service.getCompte(2)

            out[6]: {'code': 2,
                    'dateCreation': datetime.datetime(2023, 3, 21, 15, 54, 49, 982000, tzinfo=<FixedOffset '+01:00'>),
                    'solde': 734.1946358787051
                   }
```

```
Entrée [7]: client.service.getComptes()

            out[7]: [{'code': 1,
                    'dateCreation': datetime.datetime(2023, 3, 21, 15, 54, 57, 815000, tzinfo=<FixedOffset '+01:00'>),
                    'solde': 7188.611128941908
                   },
                    {'code': 2,
                    'dateCreation': datetime.datetime(2023, 3, 21, 15, 54, 57, 815000, tzinfo=<FixedOffset '+01:00'>),
                    'solde': 9496.947959849726
                   },
                    {'code': 3,
                    'dateCreation': datetime.datetime(2023, 3, 21, 15, 54, 57, 815000, tzinfo=<FixedOffset '+01:00'>),
                    'solde': 3087.3591290026493
                   }
                   ]
```

FIGURE 2.7 – client Python

CONCLUSION

en fin comme vous voyez dans ce rapport on peut consommer un web service si on a WSDL avec ne importe lquel langage utilise Client.