

# Compte Rendu SOAP

Lu Chenxin, Xu Jitao, Zhu Zhenghui

Cas d'utilisation:

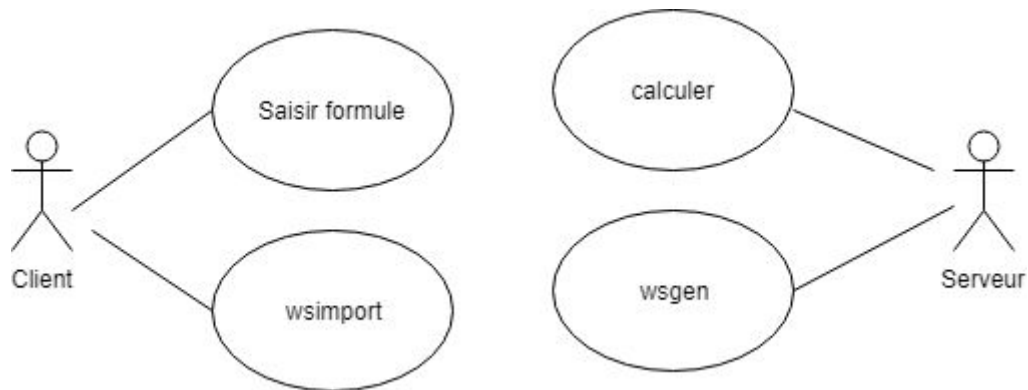


Diagramme de sequence:

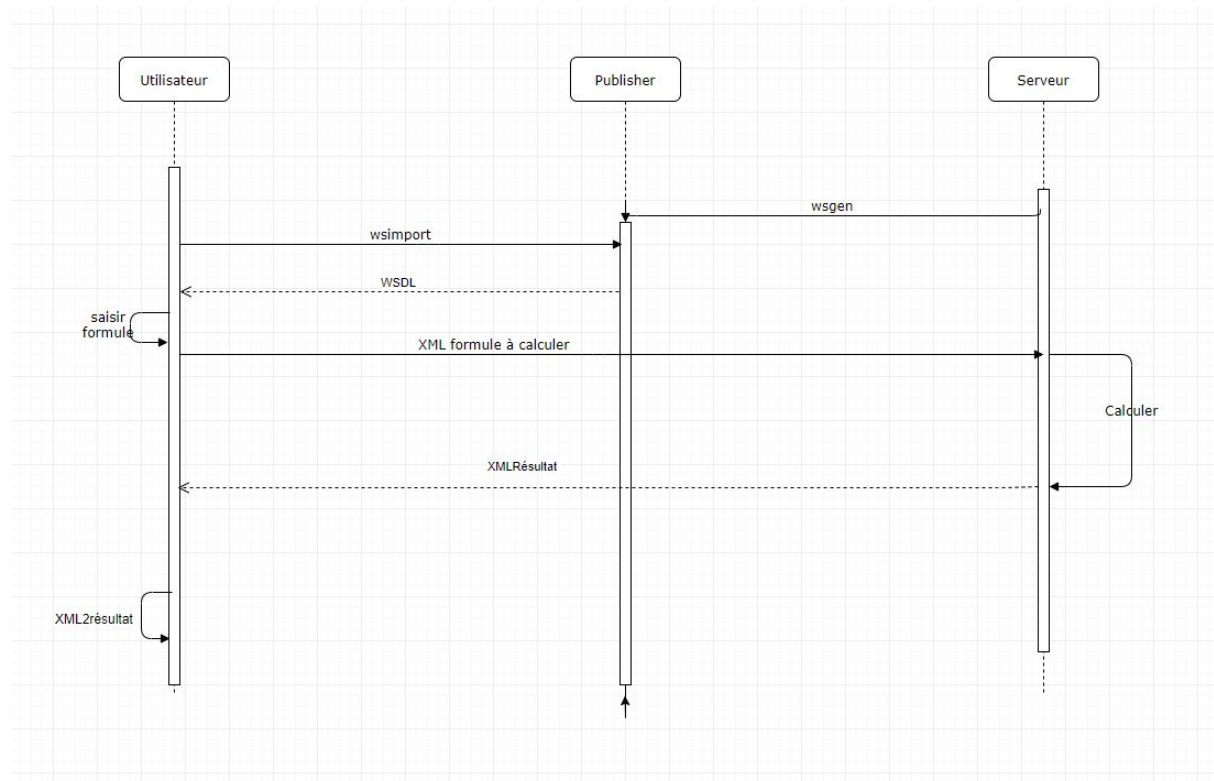
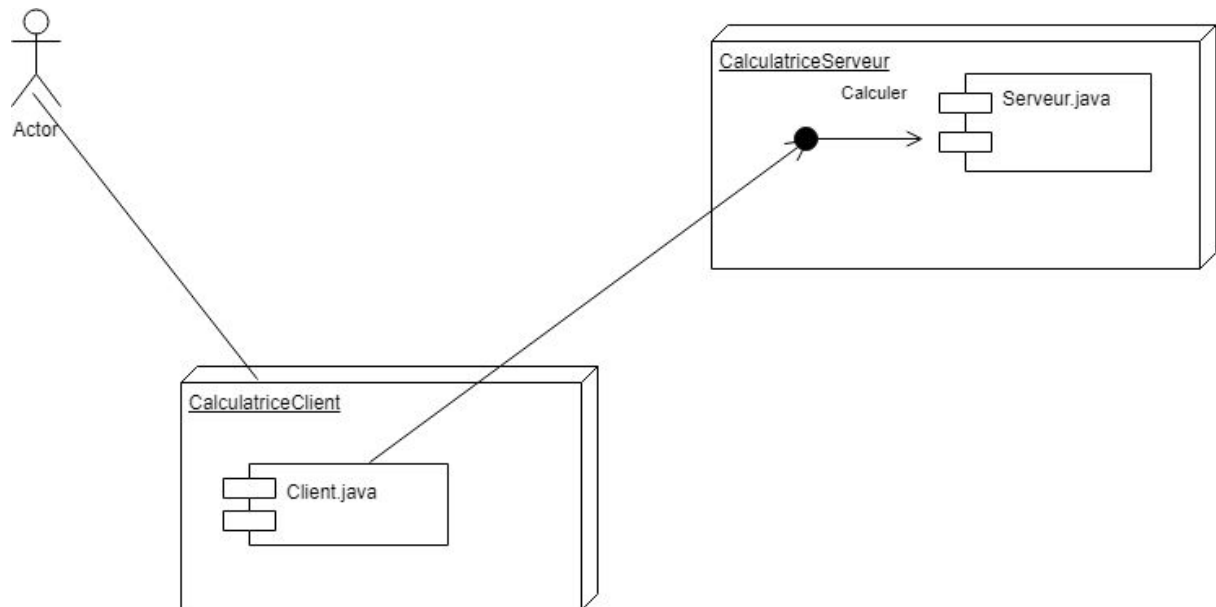


Diagramme de déploiement:



### Code Serveur:

```

package SOAP.service;

import javax.xml.ws.WebService;
import javax.xml.ws.WebMethod;
import javax.xml.ws.soap.SOAPBinding;
import javax.xml.ws.soap.SOAPBinding.Style;

@WebService
public interface CalculatriceServer {

    @WebMethod
    public int addition(int a, int b);

    @WebMethod
    public int soustraction(int a, int b) throws ResultatNegatifException;

    @WebMethod
    public int multiplication(int a, int b);

    @WebMethod
    public int division(int a, int b) throws DiviseurNullException;

    @WebMethod
    public int getOperande(String operande) throws EntreeNegatifException,
    EntreeNonEntierException;
}

```

### Code ServeurImpl:

```

package SOAP.service;

import javax.xml.ws.WebService;

@WebService(endpointInterface = "SOAP.service.CalculatriceServer")
public class CalculatriceServerImpl implements CalculatriceServer {

    public int addition(int a, int b) {
        return a + b;
    }
}

```

```

    }

    public int soustraction(int a, int b) throws ResultatNegatifException {
        if (a - b < 0) {
            throw new ResultatNegatifException("Erreur: Le résultat est négatif.");
        }
        return a - b;
    }

    public int multiplication(int a, int b) {
        return a * b;
    }

    public int division(int a, int b) throws DiviseurNullException{
        if (b == 0) {
            throw new DiviseurNullException("Pour la division, le diviseur ne peut pas être
null.");
        }
        return a / b;
    }

    public int getOperande(String operande) throws EntreeNegatifException,
EntreeNonEntierException{
        int opInt;
        try {
            opInt = Integer.parseInt(operande);
        }
        catch(Exception e) {
            throw new EntreeNonEntierException("L'opérande entré doit être entier.");
        }
        if (Integer.parseInt(operande) < 0) {
            throw new EntreeNegatifException("L'opérande entré doit être non négatif.");
        }
        return opInt;
    }
}

```

### Code ServeurPublisher.

```

package SOAP.service;

import javax.xml.ws.Endpoint;

public class CalculatriceServerPublisher {
    public static void main(String[] args) {
        Endpoint.publish("http://192.168.1.111:9876/calculatrice", new
CalculatriceServerImpl());
    }
}

```

### Code Client.

```

package client;

import client.CalculatriceServerImplService;

```

```

import client.CalculatriceServer;
import java.util.Scanner;

public class Client {

    public static void main(String[] args) throws DiviseurNullException_Exception,
EntreeNegatifException_Exception, EntreeNonEntierException_Exception,
ResultatNegatifException_Exception {
        Scanner sc;
        CalculatriceServerImplService service = new CalculatriceServerImplService();
        CalculatriceServer port = service.getCalculatriceServerImplPort();
        while(true) {
            System.out.println("Saisir la formule à calculer, séparer l'opérande et
l'opération par espace");
            sc = new Scanner(System.in);
            String calcul = sc.nextLine();
            if (calcul.equalsIgnoreCase("exit"))
                break;
            String[] composants = calcul.split(" ");
            String operande1 = composants[0];
            String operande2 = composants[2];
            String operation = composants[1];
            int result = 0;
            try {
                int op1 = port.getOperande(operande1);
                int op2 = port.getOperande(operande2);
                switch(operation) {
                    case "+":
                        result = port.addition(op1, op2);
                        break;
                    case "-":
                        result = port.soustraction(op1, op2);
                        break;
                    case "*":
                        result = port.multiplication(op1, op2);
                        break;
                    case "/":
                        result = port.division(op1, op2);
                        break;
                }
                System.out.println("Le résultat est : " + result);
            }
            catch (Exception e) {
                System.err.println(e.getMessage());
            }
        }
    }
}

```

## Conclusion:

On a développé une caculatrice qui fonctionne comme Web Service. Le service SOAP permet de publier les fichiers de web service du côté serveur avec un programme de

Publisher. Une fois le poste client a récupéré et a compilé ces fichiers, les formules saisies dans le terminal client peuvent invoquer directement les méthodes du côté serveur par requêtes XML. Comparé à la calculatrice on a réalisé avec sockets, avec le Web Service on peut facilement définir et publier l'adresse du serveur sans modifier les code en utilisant WSDL.