COMPTE RENDU TP1 SOA

Sahar Mannai
-3ING-IDL1

NOTION DE BASE

Notion du SOA:

SOA, Acronyme de Service Oriented Architecture, est traduit en français par Architecture Orientée Services. En fait , c'est un modèle de développement logiciel à base de composants applicatifs distribués et doté de fonctions de découverte, de contrôle d'accès, de mappage de données et de sécurité.

Notion de service:

Défini comme un processus de production ayant un point de contact physique entre le client et le processus, et fournissant des biens intangibles ; à ce point de contact, la production et la consommation sont simultanées, un service revêt des spécificités qui posent problème au contrôle de gestion.
Ainsi, les attributs spécifiques des activités de services contraignent le contrôle de gestion à faire preuve d'adaptabilité pour être efficace et efficient.

Notion de API:

API est l'acronyme d'**Application Programming Interface**, que l'on traduit en français par **interface de programmation applicative** ou **interface de programmation d'application**.
L'API est une solution informatique qui permet à des applications de communiquer entre elles et de s'échanger mutuellement des services ou des données.

Différence entre Rest et Soap:

SOAP est un protocole tandis que REST est un style d'architecture.
La différence majeure entre ces 2 éléments est le degré de liaison entre le client et le serveur. Un client développé avec le protocole SOAP ressemble à un logiciel d'ordinateur, car il est étroitement lié au serveur. Si une modification est effectuée d'un côté ou de l'autre, l'ensemble peut ne plus fonctionner. Il faut effectuer des mises à jour du client s'il y a des changements sur le serveur et vice-versa.

PARTIE 1:

Code :

- **Classe Serveur.java :**

serveur.java

```java
package idl1;

import javax.xml.ws.Endpoint;

public class serveur {


    static final String URL ="http://localhost:8082/";

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("running on"+URL);
      Endpoint.publish(URL, new Usercontroller());
    }

}
```

- **Classe Hello.java :**

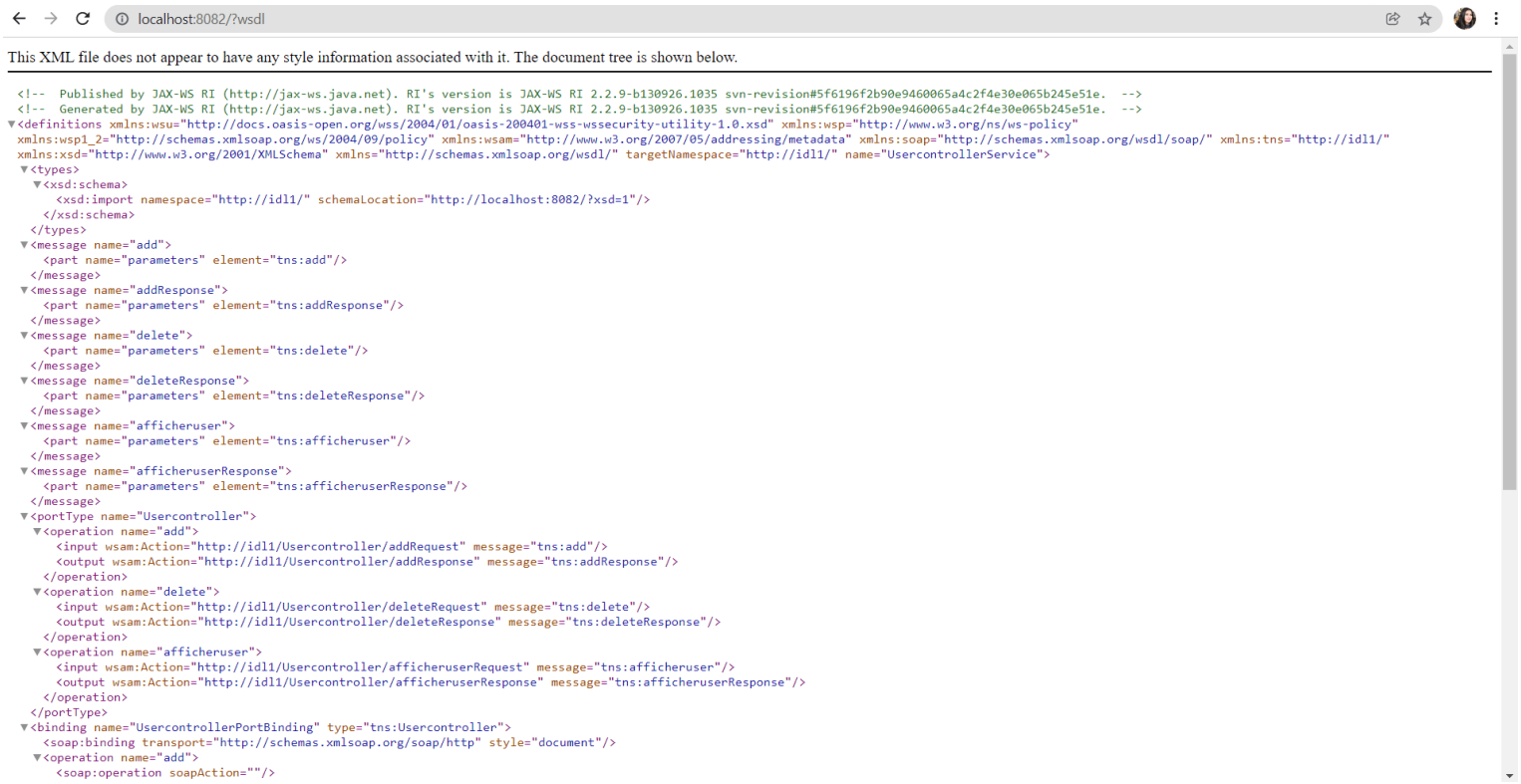
serveur.java | hello.java

```java
package idl1;

import java.util.ArrayList;

@WebService
public class hello {
    //exposé dans le web
    @WebMethod
public String sayHello(){
    return "HELLO WORLD";
}
```

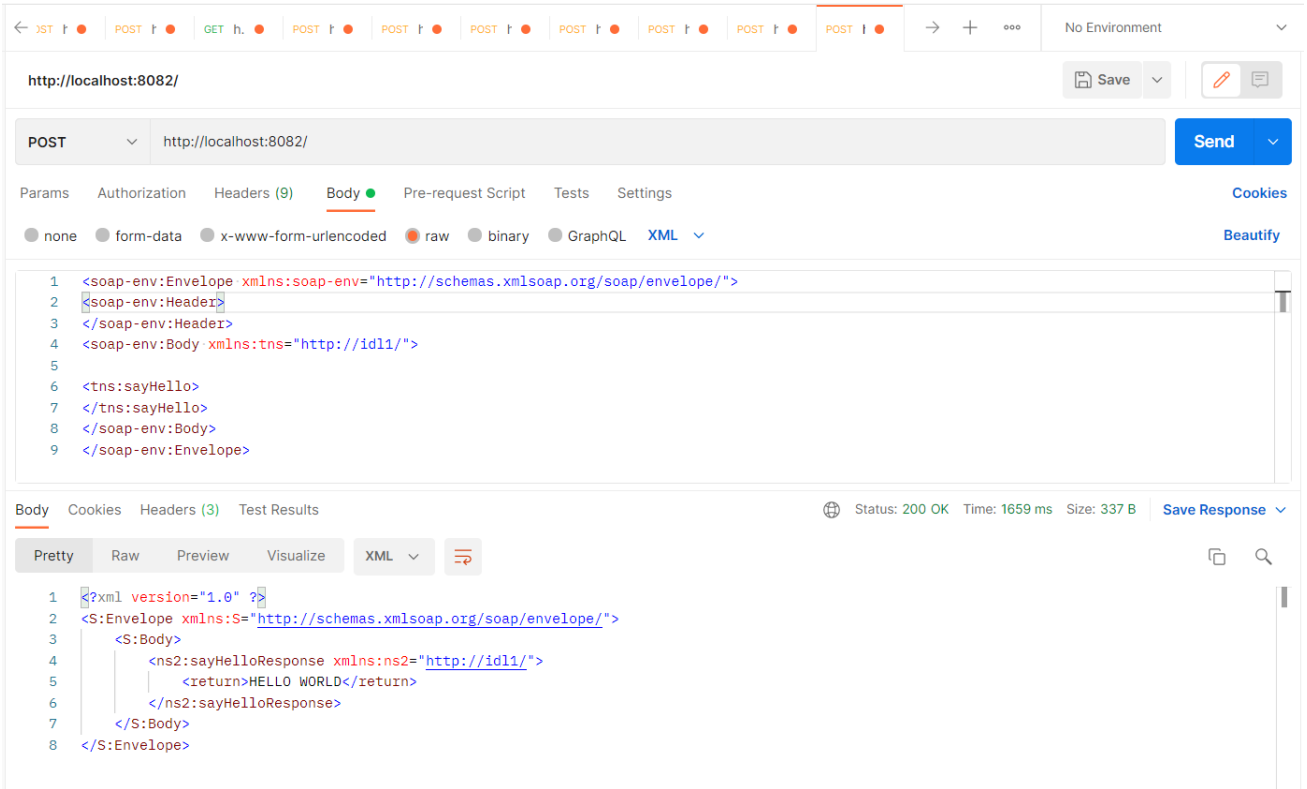
PARTIE 1:

Exécution :

- **Génération du fichier WSDL :**



- **Exécution par le postman :**



PARTIE 2:

Code de la méthode d'additin :

serveur.java | hello.java | User.java | Usercontroller.java

```java
package idl1;

import java.util.ArrayList;

@WebService
public class hello {
    //exposé dans le web
    @WebMethod
public String sayHello(){
    return "HELLO WORLD";
}
    @WebMethod
    public int somme(@WebParam(name="a")int a ,@WebParam(name="b") int b ){
        return a+b;
    }
```

Exécution par le Postman :

POST http://localhost:8082/ Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL XML Beautify

```xml
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
<soap-env:Header>
</soap-env:Header>
<soap-env:Body xmlns:tns="http://idl1/">
<tns:somme>
<a>5</a>
<b>2</b>
</tns:somme>
</soap-env:Body>
</soap-env:Envelope>
```

Body Cookies Headers (3) Test Results Status: 200 OK Time: 546 ms Size: 321 B Save Response

Pretty Raw Preview Visualize XML

```xml
<?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:sommeResponse xmlns:ns2="http://idl1/">
            <return>7</return>
        </ns2:sommeResponse>
    </S:Body>
</S:Envelope>
```

PARTIE 2:

Code de la méthode de la soustraction:

serveur.java | hello.java | User.java | Usercontroller.java

```java
package idl1;

import java.util.ArrayList;

@WebService
public class hello {
    //exposé dans le web
    @WebMethod
public String sayHello(){
    return "HELLO WORLD";
}
    @WebMethod
    public int somme(@WebParam(name="a")int a ,@WebParam(name="b") int b ){
        return a+b;
    }

    @WebMethod
    public int soustraction(@WebParam(name="a")int a ,@WebParam(name="b") int b ){
        return a-b;
    }
```

Exécution par le Postman :

POST http://localhost:8082/ Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL XML Beautify

```xml
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
<soap-env:Header>
</soap-env:Header>
<soap-env:Body xmlns:tns="http://idl1/">
<tns:soustraction>
<a>5</a>
<b>2</b>
</tns:soustraction>
</soap-env:Body>
</soap-env:Envelope>
```

Body Cookies Headers (3) Test Results Status: 200 OK Time: 519 ms Size: 335 B Save Response

Pretty Raw Preview Visualize XML

```xml
<?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:soustractionResponse xmlns:ns2="http://idl1/">
            <return>3</return>
        </ns2:soustractionResponse>
    </S:Body>
</S:Envelope>
```

PARTIE 2:

Code de la méthode de multiplication:

serveur.java | hello.java | User.java | Usercontroller.java

```java
package idl1;

import java.util.ArrayList;

@WebService
public class hello {
    //exposé dans le web
    @WebMethod
public String sayHello(){
    return "HELLO WORLD";
}
    @WebMethod
    public int somme(@WebParam(name="a")int a ,@WebParam(name="b") int b ){
        return a+b;
    }

    @WebMethod
    public int soustraction(@WebParam(name="a")int a ,@WebParam(name="b") int b ){
        return a-b;
    }

    @WebMethod
    public int multiplication(@WebParam(name="a")int a ,@WebParam(name="b") int b ){
        return a*b;
    }



}
```

Exécution par le Postman :

POST http://localhost:8082/ Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL XML Beautify

```xml
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
<soap-env:Header>
</soap-env:Header>
<soap-env:Body xmlns:tns="http://idl1/">
<tns:multiplication>
<a>5</a>
<b>2</b>
</tns:multiplication>
</soap-env:Body>
</soap-env:Envelope>
```

Body Cookies Headers (3) Test Results Status: 200 OK Time: 524 ms Size: 340 B Save Response

Pretty Raw Preview Visualize XML

```xml
<?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:multiplicationResponse xmlns:ns2="http://idl1/">
            <return>10</return>
        </ns2:multiplicationResponse>
    </S:Body>
</S:Envelope>
```

PARTIE 3:

Code :

- **Classe User :**

serveur.java | hello.java | User.java | Usercontroller.java

```java
package idl1;
import javax.xml.bind.annotation.XmlAccessType;
import javax.xml.bind.annotation.XmlAccessorType;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlTransient;

@XmlRootElement
@XmlAccessorType(XmlAccessType.FIELD)
public class User {
private String username;
@XmlTransient
private String password;
public String getUsername() {
    return username;
}
public void setUsername(String username) {
    this.username = username;
}
public String getPassword() {
    return password;
}
public void setPassword(String password) {
    this.password = password;
}
public User(String username, String password) {
    super();
    this.username = username;
    this.password = password;
}
public User() {
    super();
}
}

```

PARTIE 3:

- **Classe UserController :**

serveur.java | hello.java | User.java | Usercontroller.java

```java
package idl1;

import java.util.ArrayList;
@WebService
public class Usercontroller {
    public List<User> listU=new ArrayList<>();

    @WebMethod
    public User add(@WebParam(name="u")User u ){
    listU.add(u);
        return u;
    }

    @WebMethod
    public List<User> afficheruser(){

    return listU;
    }
    @WebMethod
    public List<User> delete(@WebParam(name="id")int id) {
        listU.remove(id);
        return listU ;
    }
}
```

- **Exécution de AddUser :**
  - Ajout de l'utilisateur "Sahar" avec le password "Sahar" :

POST http://localhost:8082/ Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL XML Beautify

```xml
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
<soap-env:Header>
</soap-env:Header>
<soap-env:Body xmlns:tns="http://idl1/">
<tns:add>
<u>
<username>
sahar
</username>
<password>
sahar
</password>
</u>
</tns:add>
</soap-env:Body>
</soap-env:Envelope>
```

PARTIE 3:

Body Cookies Headers (3) Test Results Status: 200 OK Time: 653 ms Size: 344 B Save Response

Pretty Raw Preview Visualize XML

```
<?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:addResponse xmlns:ns2="http://idl1/">
            <return>
                <username>
sahar
</username>
            </return>
        </ns2:addResponse>
    </S:Body>
</S:Envelope>
```

- Ajout de l'utilisateur "Safia" avec le password "Safia" :

POST http://localhost:8082/ Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL XML Beautify

```
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
<soap-env:Header>
</soap-env:Header>
<soap-env:Body xmlns:tns="http://idl1/">
<tns:add>
<u>
<username>
safia
</username>
<password>
safia
</password>
</u>
</tns:add>
</soap-env:Body>
</soap-env:Envelope>
```

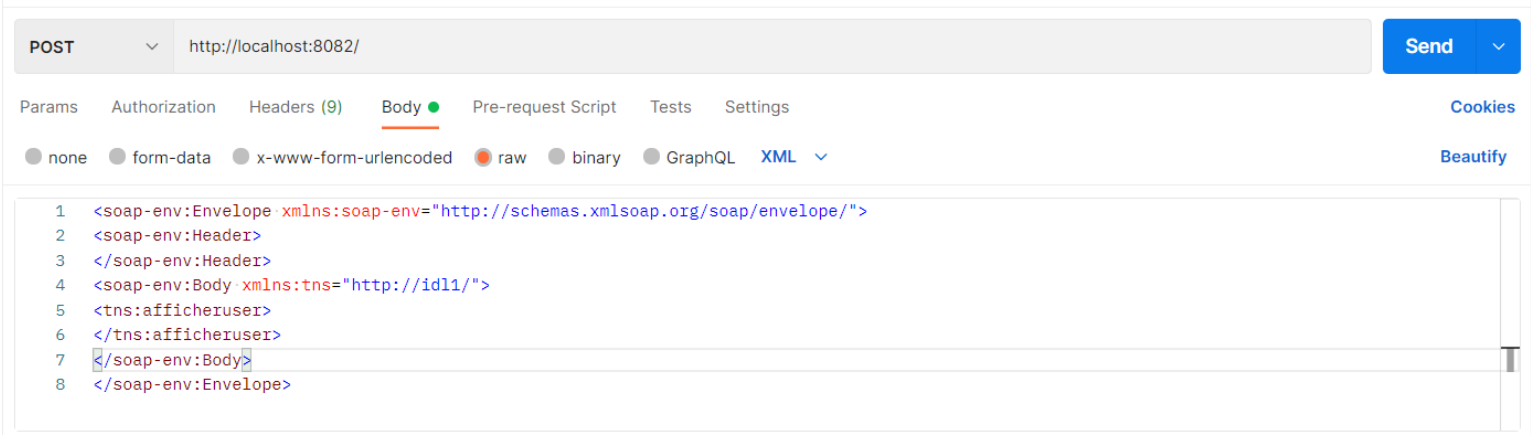
Cookies Headers (3) Test Results Status: 200 OK Time: 529 ms Size: 344 B Save Respon

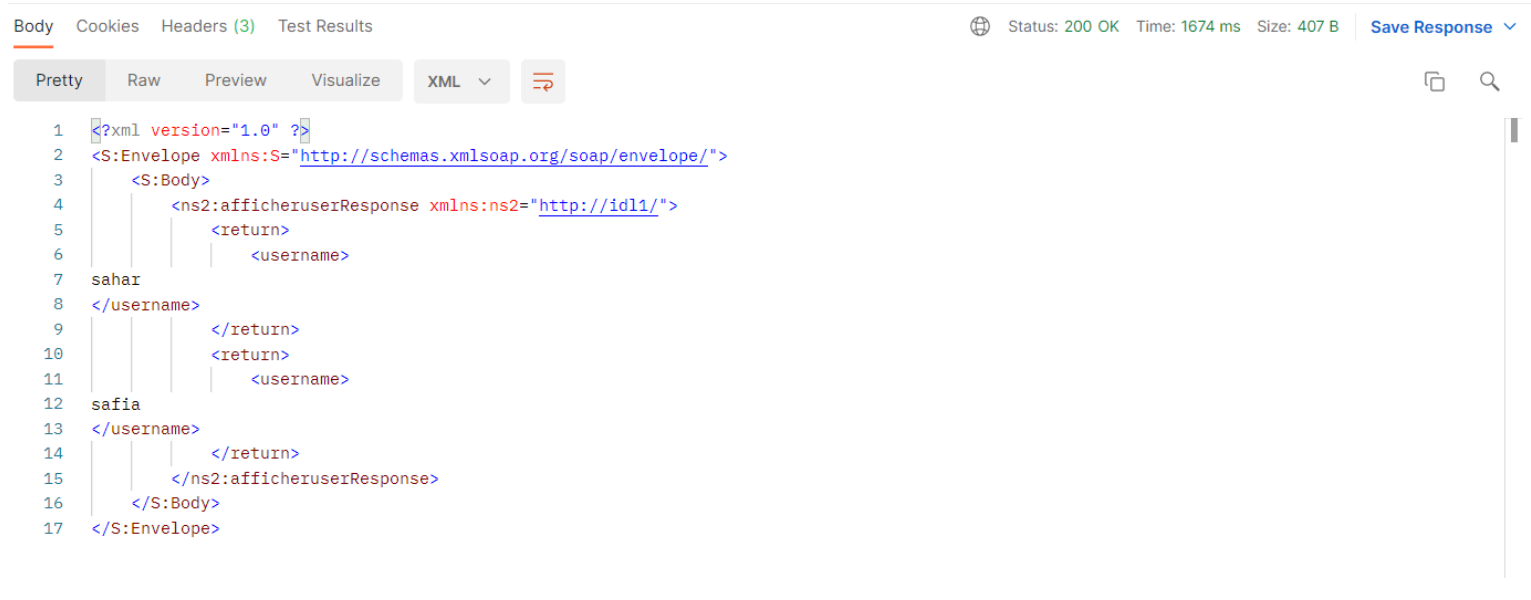
ty Raw Preview Visualize XML

```
<?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:addResponse xmlns:ns2="http://idl1/">
            <return>
                <username>
safia
</username>
            </return>
        </ns2:addResponse>
    </S:Body>
</S:Envelope>
```

PARTIE 3:

- **Exécution de ListUser :**





PARTIE 3:

- **Exécution de DeleteUser :**

POST http://localhost:8082/ Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL XML Beautify

```
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
<soap-env:Header>
</soap-env:Header>
<soap-env:Body xmlns:tns="http://idl1/">
<tns:delete>
<id>
0
</id>
</tns:delete>
</soap-env:Body>
</soap-env:Envelope>
```

Body Cookies Headers (3) Test Results Status: 200 OK Time: 800 ms Size: 350 B Save Response

Pretty Raw Preview Visualize XML

```
<?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:deleteResponse xmlns:ns2="http://idl1/">
            <return>
                <username>
safia
</username>
            </return>
        </ns2:deleteResponse>
    </S:Body>
</S:Envelope>
```