

TP SOA

# COMPTE RENDU TP SOAP

---

Réalisé par:  
Asma Ben Jemie  
Groupe:3IDL1  
Sous groupe :1

2021-2022

# EXERCICE 1

## classe Hello

**La classe «Hello» implémente les méthodes du web services**

```
Hello.java x
1 package idl1;
2 import java.util.ArrayList;
3 import java.util.List;
4 import javax.jws.WebMethod;
5 import javax.jws.WebParam;
6 import javax.jws.WebService;
7
8 @WebService
9 public class Hello {
10     @WebMethod
11     public String sayHello(){
12         return "HELLO WORLD";
13     }
14     @WebMethod
15     public int somme(@WebParam(name="a")int a ,@WebParam(name="b") int b ){
16         return a+b;
17     }
18     @WebMethod
19     public int soustraction(@WebParam(name="a")int a ,@WebParam(name="b") int b ){
20         return a-b;
21     }
22
23     @WebMethod
24     public int multiplication(@WebParam(name="a")int a ,@WebParam(name="b") int b ){
25         return a*b;
26     }
27 }
```

Cette classe permet de définir l'implémentation des méthodes somme, soustraction, multiplication et sayhello

# EXERCICE 1

## Classe Serveur

La classe «Serveur» permet d'instancier le service et l'associer à une URL pour l'accéder à distance grâce à la méthode `Endpoint.publish`

```
Server.java ×  
1 package idl1;  
2  
3 import javax.xml.ws.Endpoint;  
4  
5 public class Serveur {  
6  
7  
8     static final String URL ="http://localhost:8081/";  
9  
10    public static void main(String[] args) {  
11        // TODO Auto-generated method stub  
12        System.out.println("running on"+URL);  
13        Endpoint.publish(URL, new Hello());  
14    }  
15  
16 }  
17
```

# EXERCICE 1

## Capture Postman

**Exécution de la méthode «sayHello» qui permet d'afficher le message "Hello Word"**

The screenshot shows the Postman interface for a POST request to `http://localhost:8081/`. The 'Body' tab is selected, showing a SOAP envelope. The response is displayed in the 'Test Results' section, showing a SOAP response with the message 'HELLO WORLD'.

Request URL: `http://localhost:8081/`

Method: **POST**

Body (XML):

```
1 <soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap-env:Header>
3   </soap-env:Header>
4   <soap-env:Body xmlns:tns="http://idl1/">
5     <tns:sayHello>
6     </tns:sayHello>
7   </soap-env:Body>
8 </soap-env:Envelope>
```

Response (XML):

```
1 <?xml version="1.0" ?>
2 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
3   <S:Body>
4     <ns2:sayHelloResponse xmlns:ns2="http://idl1/">
5       <return>HELLO WORLD</return>
6     </ns2:sayHelloResponse>
7   </S:Body>
8 </S:Envelope>
```

# EXERCICE 1

## Fichier wsdl

```
localhost:8081/?wsdl
localhost:8081/?wsdl
<?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/ns/ws-policy"
xmlns:wspi_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/wsdl/soap/" xmlns:tns="http://idl1/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://idl1/" name="HelloService">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://idl1/" schemaLocation="http://localhost:8081/?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="sayHello">
    <part name="parameters" element="tns:sayHello"/>
  </message>
  <message name="sayHelloResponse">
    <part name="parameters" element="tns:sayHelloResponse"/>
  </message>
  <message name="somme">
    <part name="parameters" element="tns:somme"/>
  </message>
  <message name="sommeResponse">
    <part name="parameters" element="tns:sommeResponse"/>
  </message>
  <message name="soustraction">
    <part name="parameters" element="tns:soustraction"/>
  </message>
  <message name="soustractionResponse">
    <part name="parameters" element="tns:soustractionResponse"/>
  </message>
  <message name="multiplication">
    <part name="parameters" element="tns:multiplication"/>
  </message>
  <message name="multiplicationResponse">
    <part name="parameters" element="tns:multiplicationResponse"/>
  </message>
  <portType name="Hello">
    <operation name="sayHello">
      <input wsam:Action="http://idl1/Hello/sayHelloRequest" message="tns:sayHello"/>
      <output wsam:Action="http://idl1/Hello/sayHelloResponse" message="tns:sayHelloResponse"/>
    </operation>
    <operation name="somme">
      <input wsam:Action="http://idl1/Hello/sommeRequest" message="tns:somme"/>
      <output wsam:Action="http://idl1/Hello/sommeResponse" message="tns:sommeResponse"/>
    </operation>
    <operation name="soustraction">
      <input wsam:Action="http://idl1/Hello/soustractionRequest" message="tns:soustraction"/>
      <output wsam:Action="http://idl1/Hello/soustractionResponse" message="tns:soustractionResponse"/>
    </operation>
    <operation name="multiplication">
      <input wsam:Action="http://idl1/Hello/multiplicationRequest" message="tns:multiplication"/>
      <output wsam:Action="http://idl1/Hello/multiplicationResponse" message="tns:multiplicationResponse"/>
    </operation>
  </portType>
  <binding name="HelloPortBinding" type="tns:Hello">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="sayHello">
      <soap:operation soapAction="">
        <input>
          <soap:body use="literal"/>
        </input>
        <output>
          <soap:body use="literal"/>
        </output>
      </operation>
    <operation name="somme">
      <soap:operation soapAction="">
        <input>
          <soap:body use="literal"/>
        </input>
        <output>
          <soap:body use="literal"/>
        </output>
      </operation>
    <operation name="soustraction">
      <soap:operation soapAction="">
        <input>
          <soap:body use="literal"/>
        </input>
        <output>
          <soap:body use="literal"/>
        </output>
      </operation>
    <operation name="multiplication">
      <soap:operation soapAction="">
        <input>
          <soap:body use="literal"/>
        </input>
        <output>
          <soap:body use="literal"/>
        </output>
      </operation>
    </binding>
  </service name="HelloService">
    <port name="HelloPort" binding="tns:HelloPortBinding">
      <soap:address location="http://localhost:8081/">
    </port>
  </service>
</definitions>
```

# EXERCICE 1

## Capture Postman

### Exécution de la méthode «somme»

http://localhost:8081/

POST

http://localhost:8081/

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

XML

```
1 <soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap-env:Header>
3   </soap-env:Header>
4   <soap-env:Body xmlns:tns="http://idl1/">
5     <tns:somme>
6       <a>7</a>
7       <b>8</b>
8     </tns:somme>
9   </soap-env:Body>
10 </soap-env:Envelope>
```

Body

Cookies

Headers (3)

Test Results

Pretty

Raw

Preview

Visualize

XML



```
1 <?xml version="1.0" ?>
2 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
3   <S:Body>
4     <ns2:sommeResponse xmlns:ns2="http://idl1/">
5       <return>15</return>
6     </ns2:sommeResponse>
7   </S:Body>
8 </S:Envelope>
```

# EXERCICE 1

## Capture Postman

### Exécution de la méthode «soustraction»

The screenshot displays the Postman interface for a SOAP POST request. The URL bar shows `http://localhost:8081/`. The request method is **POST**, and the URL is `http://localhost:8081/`. The **Body** tab is selected, showing the raw XML request body. The response is visible at the bottom, showing the XML response body.

**Request Body (XML):**

```
1 <soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap-env:Header>
3   </soap-env:Header>
4   <soap-env:Body xmlns:tns="http://idl1/">
5     <tns:soustraction>
6       <a>10</a>
7       <b>8</b>
8     </tns:soustraction>
9   </soap-env:Body>
10  </soap-env:Envelope>
```

**Response Body (XML):**

```
1 <?xml version="1.0" ?>
2 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
3   <S:Body>
4     <ns2:soustractionResponse xmlns:ns2="http://idl1/">
5       <return>2</return>
6     </ns2:soustractionResponse>
7   </S:Body>
8 </S:Envelope>
```

# EXERCICE 1

## Capture Postman

### Exécution de la méthode «multiplication»

The screenshot displays the Postman interface for a POST request to `http://localhost:8081/`. The 'Body' tab is selected, showing a SOAP envelope with a multiplication request. The 'raw' radio button is chosen. Below the request, the 'Body' tab is also selected, showing the XML response in 'Pretty' format. The response is a SOAP envelope containing a multiplication response with the result 20.

**Request:**

```
1 <soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap-env:Header>
3   </soap-env:Header>
4   <soap-env:Body xmlns:tns="http://idl1/">
5     <tns:multiplication>
6       <a>10</a>
7       <b>2</b>
8     </tns:multiplication>
9   </soap-env:Body>
10  </soap-env:Envelope>
```

**Response:**

```
1 <?xml version="1.0" ?>
2 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
3   <S:Body>
4     <ns2:multiplicationResponse xmlns:ns2="http://idl1/">
5       <return>20</return>
6     </ns2:multiplicationResponse>
7   </S:Body>
```



# EXERCICE 2

Création du web services permettant d'ajouter un utilisateur, supprimer un utilisateur, d'afficher les utilisateurs

## Captures

L'image présente l'implémentation de la classe User contenant les attributs username et password:

```
User.java ×
 8 @XmlElement
 9 @XmlAccessorType(XmlAccessType.FIELD)
10 public class User {
11     private String username;
12     @XmlTransient
13     private String password;
14     public String getUsername() {
15         return username;
16     }
17     public void setUsername(String username) {
18         this.username = username;
19     }
20     public String getPassword() {
21         return password;
22     }
23     public void setPassword(String password) {
24         this.password = password;
25     }
26     public User(String username, String password) {
27         super();
28         this.username = username;
29         this.password = password;
30     }
31     public User() {
32         super();
33     }
34 }
```

## La classe«UserController»:

```
UserController.java
1 package idl1;
2
3 import java.util.ArrayList;
4
5 @WebService
6 public class UserController {
7     public List<User> listU=new ArrayList<>();
8
9     @WebMethod
10    public User addUser(@WebParam(name="u")User u ){
11        listU.add(u);
12        return u;
13    }
14
15    @WebMethod
16    public List<User> afficheruser(){
17
18        return listU;
19    }
20
21    @WebMethod
22    public List<User> supprimerUser(@WebParam(name="id")int id ){
23
24        listU.remove(id);
25        return listU;
26    }
27 }
28
29
30
31
32 }
```

Cette classe permet de définir l'implémentation des méthodes du web services

## La classe« Serveur»:

```
serveur.java
1 package idl1;
2
3 import javax.xml.ws.Endpoint;
4
5 public class serveur {
6
7
8     static final String URL ="http://localhost:8081/";
9
10    public static void main(String[] args) {
11        // TODO Auto-generated method stub
12        System.out.println("running on"+URL);
13        Endpoint.publish(URL, new UserController());
14    }
15
16 }
17 }
```

Cette classe permet d'instancier le service, l'associer à une URL pour l'accéder à distance grâce à la méthode Endpoint.publish

# EXERCICE 2

## Fichier wsdl

localhost:8081/?wsdl

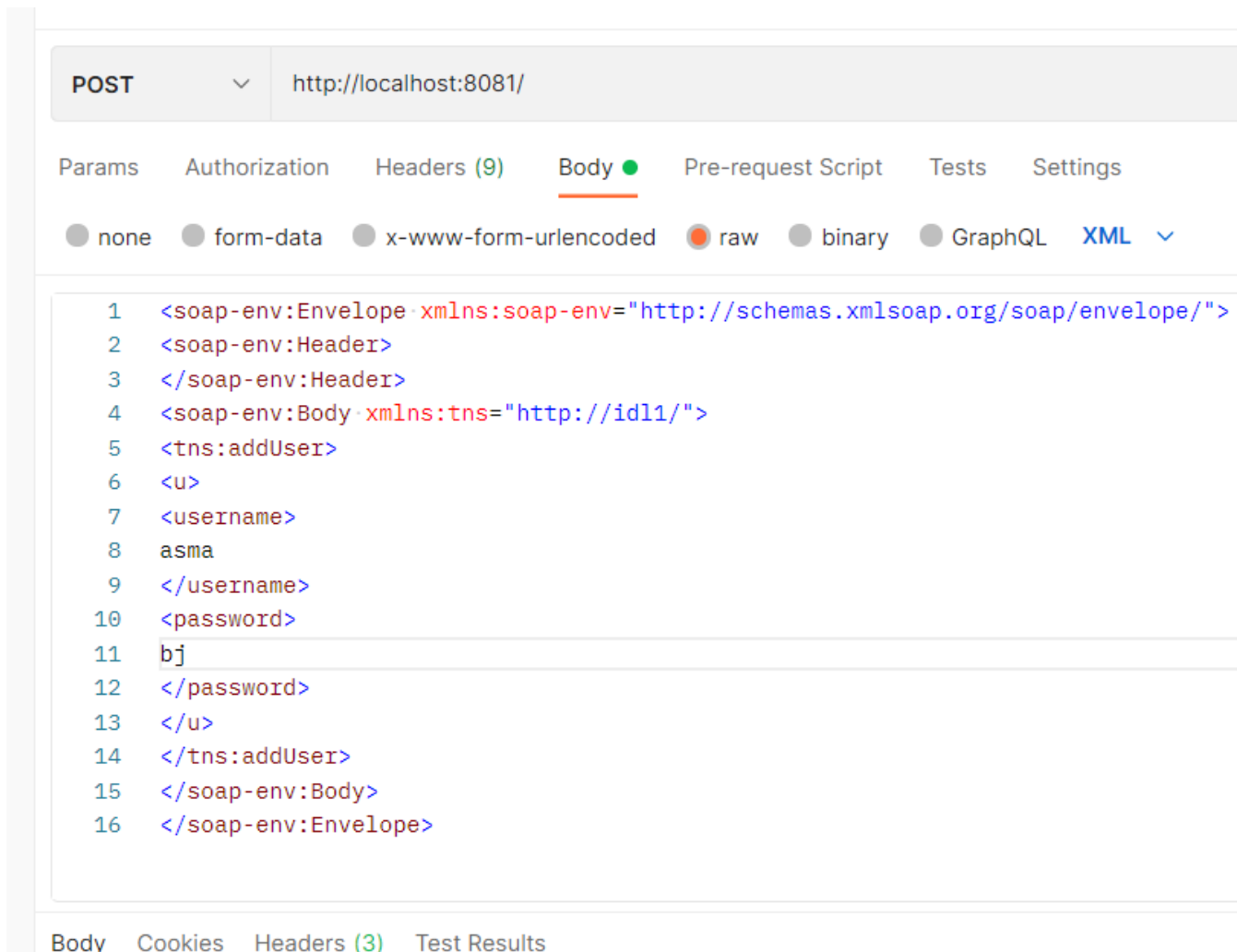
← → ↻ localhost:8081/?wsdl

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#5f6196f2b90e9460065a4c2f4e30e065b245e51e. -->
<!-- Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e51e. -->
<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/ns/ws-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/" xmlns:tns="http://idl1/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://idl1/" name="UserControllerService">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://idl1/" schemaLocation="http://localhost:8081/?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="afficherUser">
    <part name="parameters" element="tns:afficherUser"/>
  </message>
  <message name="afficherUserResponse">
    <part name="parameters" element="tns:afficherUserResponse"/>
  </message>
  <message name="addUser">
    <part name="parameters" element="tns:addUser"/>
  </message>
  <message name="addUserResponse">
    <part name="parameters" element="tns:addUserResponse"/>
  </message>
  <message name="supprimerUser">
    <part name="parameters" element="tns:supprimerUser"/>
  </message>
  <message name="supprimerUserResponse">
    <part name="parameters" element="tns:supprimerUserResponse"/>
  </message>
  <portType name="UserController">
    <operation name="afficherUser">
      <input wsam:Action="http://idl1/Usercontroller/afficherUserRequest" message="tns:afficherUser"/>
      <output wsam:Action="http://idl1/Usercontroller/afficherUserResponse" message="tns:afficherUserResponse"/>
    </operation>
    <operation name="addUser">
      <input wsam:Action="http://idl1/Usercontroller/addUserRequest" message="tns:addUser"/>
      <output wsam:Action="http://idl1/Usercontroller/addUserResponse" message="tns:addUserResponse"/>
    </operation>
    <operation name="supprimerUser">
      <input wsam:Action="http://idl1/Usercontroller/supprimerUserRequest" message="tns:supprimerUser"/>
      <output wsam:Action="http://idl1/Usercontroller/supprimerUserResponse" message="tns:supprimerUserResponse"/>
    </operation>
  </portType>
  <binding name="UserControllerPortBinding" type="tns:UserController">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="afficherUser">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
    <operation name="addUser">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
    <operation name="supprimerUser">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="UserControllerService">
    <port name="UserControllerPort" binding="tns:UserControllerPortBinding">
      <soap:address location="http://localhost:8081/">
    </port>
  </service>
</definitions>
```

# Captures Postman

## Ajouter le premier utilisateur de la méthode "addUser"

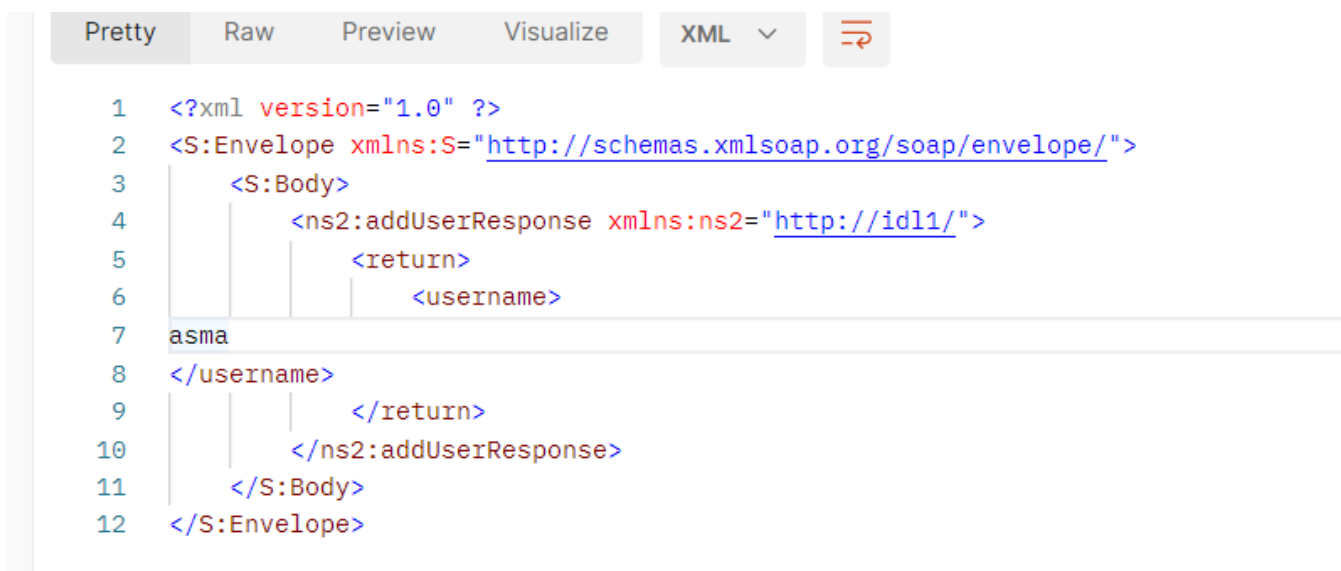


The screenshot shows the Postman interface for a POST request. The URL is `http://localhost:8081/`. The 'Body' tab is selected, and the 'raw' radio button is chosen. The XML body is as follows:

```
1 <soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
2 <soap-env:Header>
3 </soap-env:Header>
4 <soap-env:Body xmlns:tns="http://idl1/">
5 <tns:addUser>
6 <u>
7 <username>
8 asma
9 </username>
10 <password>
11 bj
12 </password>
13 </u>
14 </tns:addUser>
15 </soap-env:Body>
16 </soap-env:Envelope>
```

Below the body, the 'Body' tab is selected, and the 'Test Results' section is visible.

## Resultat:

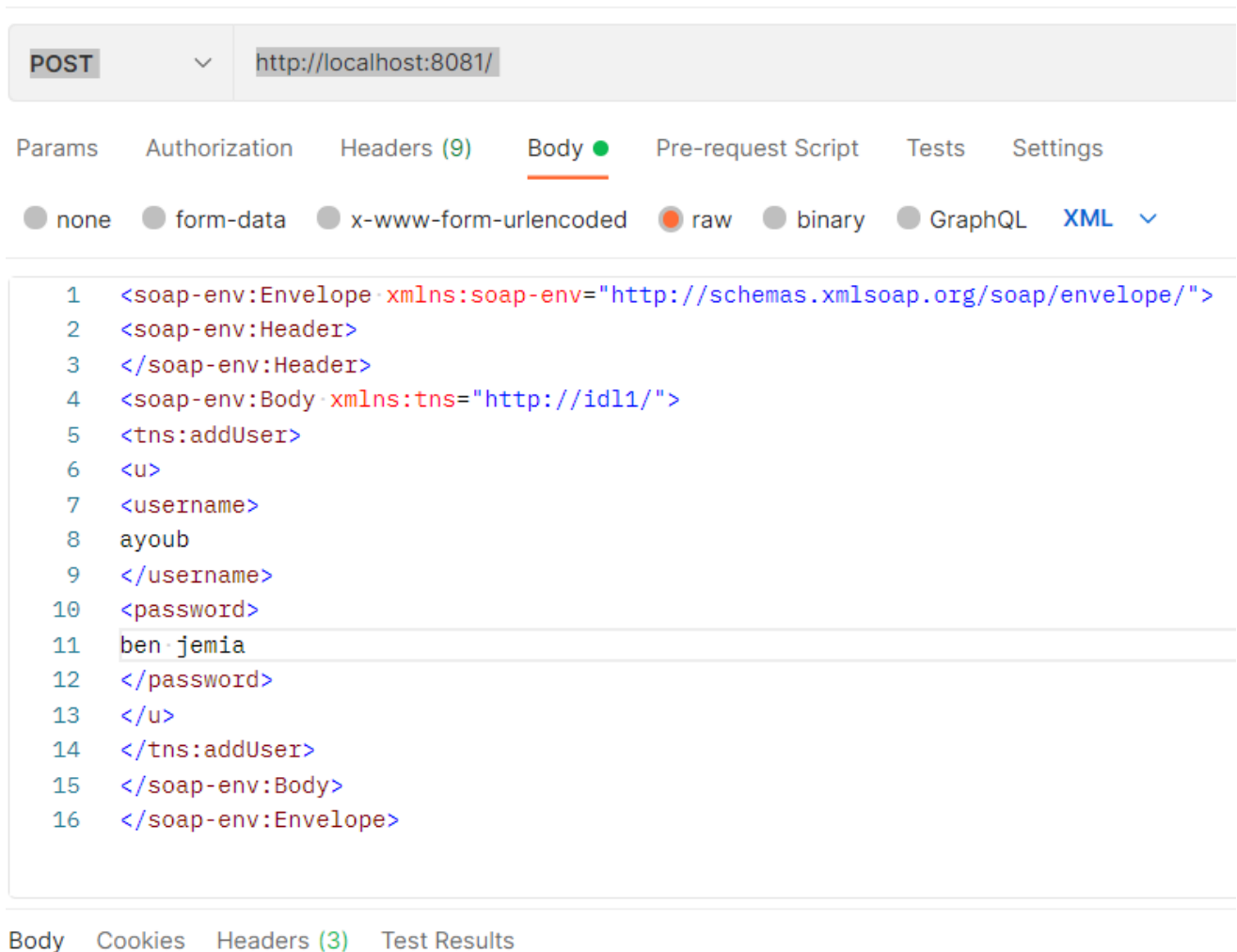


The screenshot shows the Postman interface for the response of the POST request. The 'XML' tab is selected, and the response is as follows:

```
1 <?xml version="1.0" ?>
2 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
3 |   <S:Body>
4 | |   <ns2:addUserResponse xmlns:ns2="http://idl1/">
5 | | |   <return>
6 | | | |   <username>
7 | | | | asma
8 | | | | </username>
9 | | | | </return>
10 | | | </ns2:addUserResponse>
11 | | </S:Body>
12 </S:Envelope>
```

# Captures Postman

## Ajouter le deuxième utilisateur de la méthode "addUser"

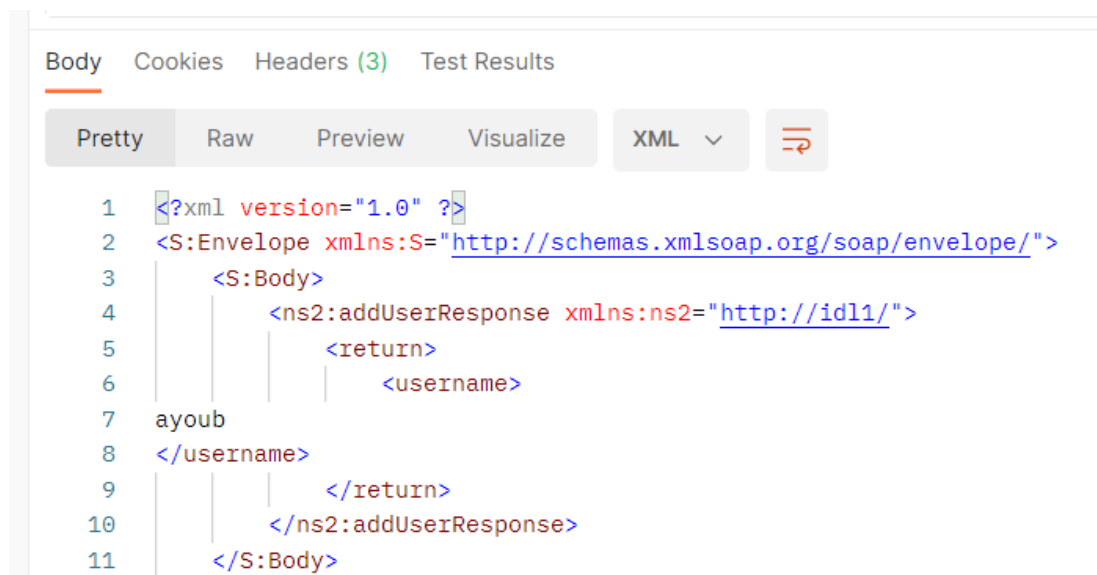


The screenshot shows the Postman interface for a POST request to `http://localhost:8081/`. The 'Body' tab is selected, and the body is formatted as XML. The XML structure is as follows:

```
1 <soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
2 <soap-env:Header>
3 </soap-env:Header>
4 <soap-env:Body xmlns:tns="http://idl1/">
5 <tns:addUser>
6 <u>
7 <username>
8 ayoub
9 </username>
10 <password>
11 ben-jemia
12 </password>
13 </u>
14 </tns:addUser>
15 </soap-env:Body>
16 </soap-env:Envelope>
```

Below the body, the 'Test Results' tab is visible.

## Resultat:

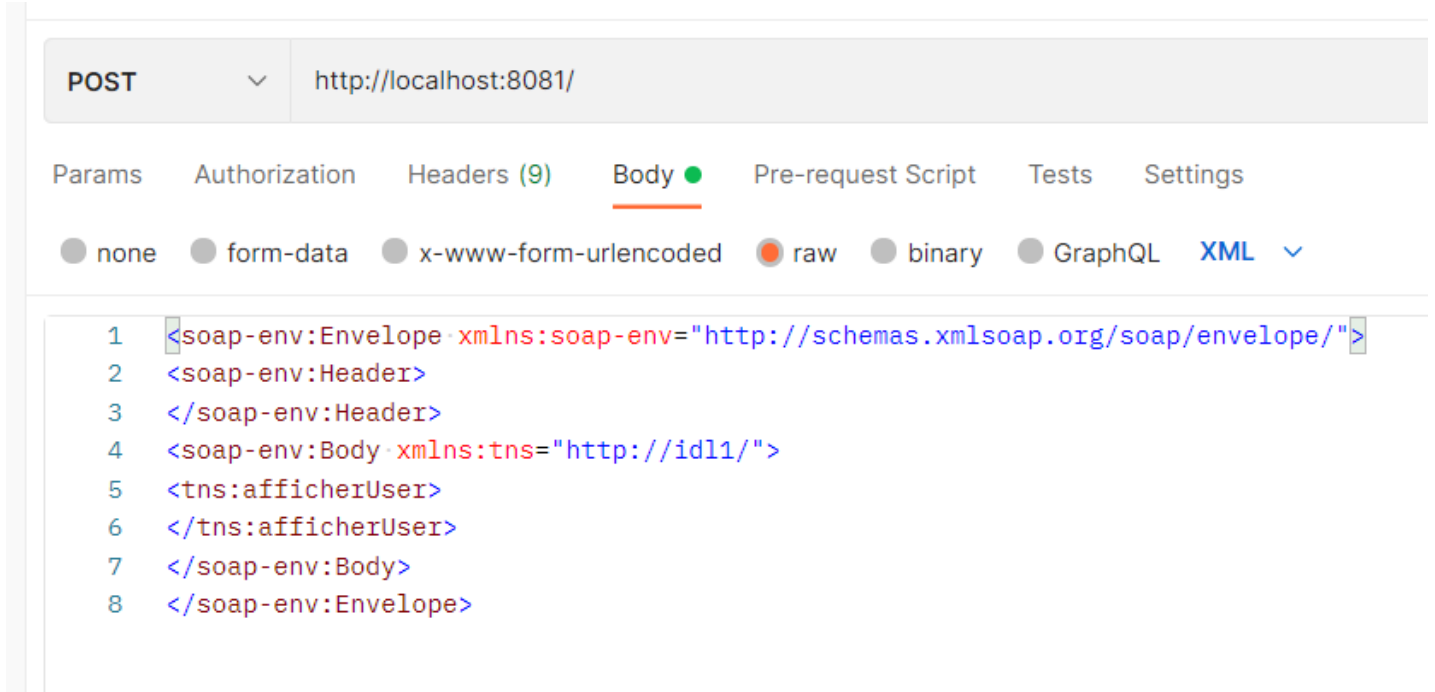


The screenshot shows the Postman interface displaying the XML response of the POST request. The 'Body' tab is selected, and the response is formatted as XML. The XML structure is as follows:

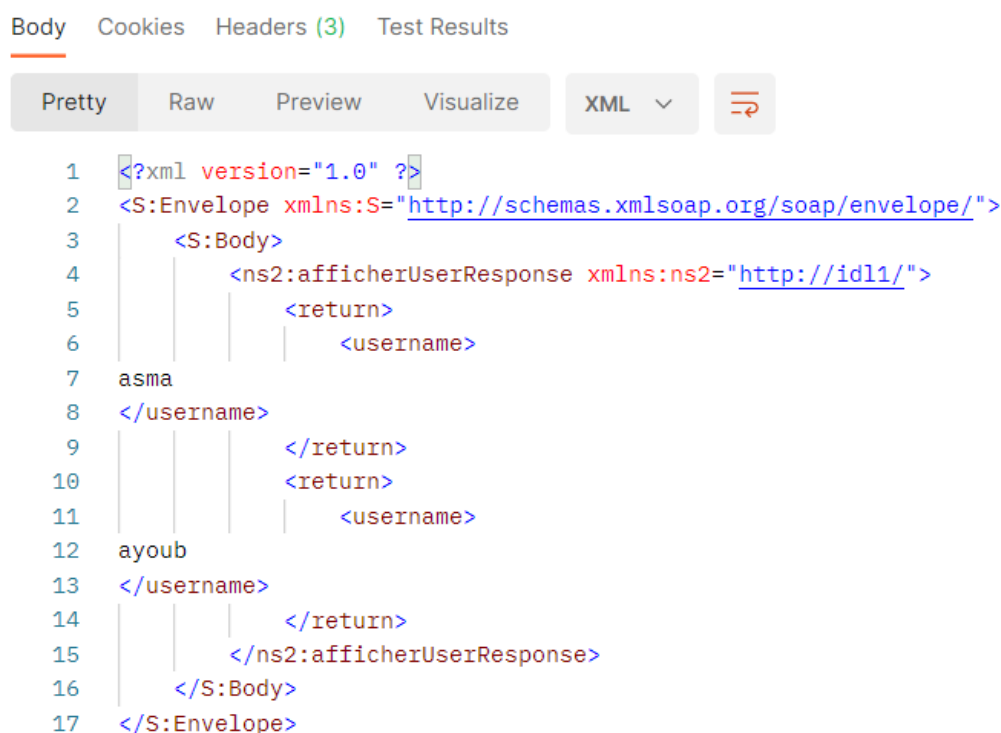
```
1 <?xml version="1.0" ?>
2 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
3 |   <S:Body>
4 |       <ns2:addUserResponse xmlns:ns2="http://idl1/">
5 |           <return>
6 |               <username>
7 |                   ayoub
8 |               </username>
9 |           </return>
10 |       </ns2:addUserResponse>
11 |   </S:Body>
```

# Captures Postman

## Test de la méthode "afficherUser"



## Resultat:



# Captures Postman

## Test de la méthode "supprimerUser"

Overview | http://localhost:8081/ | http://localhost:8081/ | http://localhost:8081/ | http://localhost:8081/

http://localhost:8081/

POST http://localhost:8081/

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

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

```
1 <soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
2 <soap-env:Header>
3 </soap-env:Header>
4 <soap-env:Body xmlns:tns="http://idl1/">
5 <tns:supprimerUser>
6 <id>
7 0
8 </id>
9 </tns:supprimerUser>
10 </soap-env:Body>
11 </soap-env:Envelope>
```

**Body** Cookies Headers (3) Test Results

## Resultat:

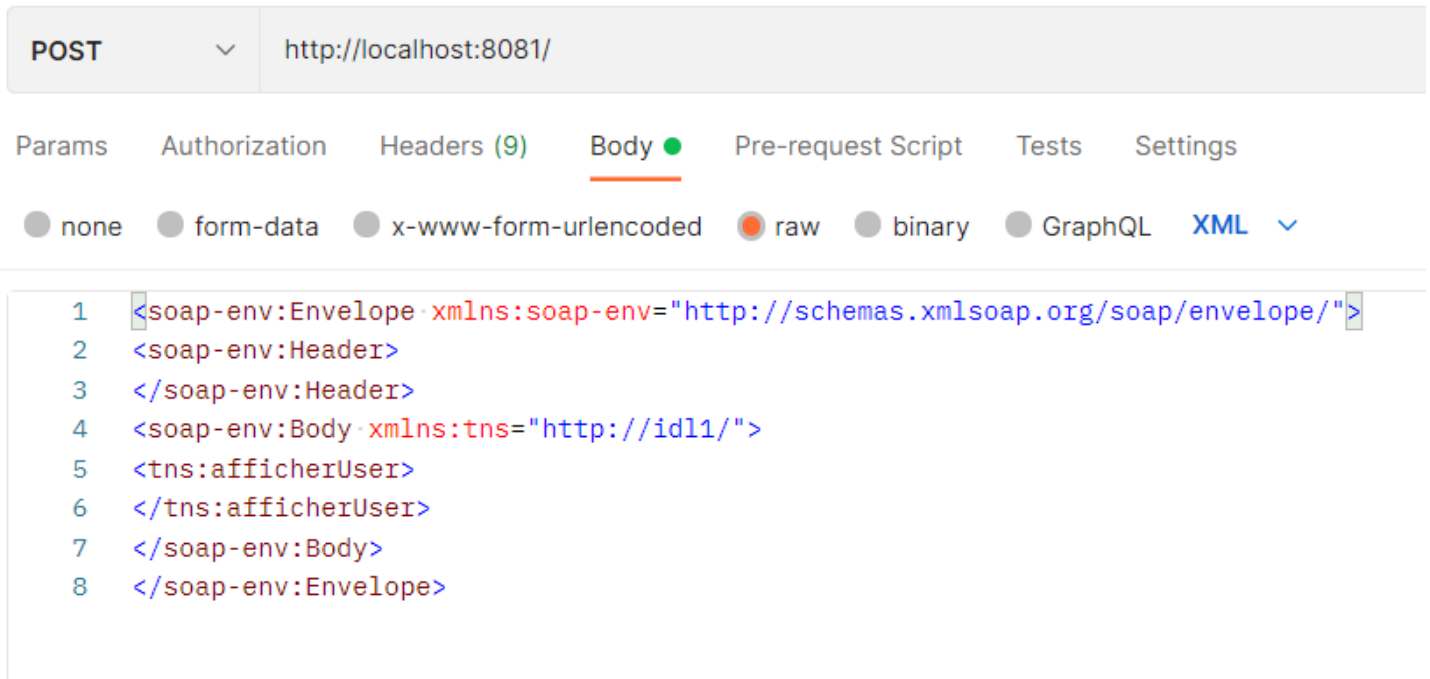
**body** Cookies Headers (3) Test Results

Pretty Raw Preview Visualize XML

```
1 <?xml version="1.0" ?>
2 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
3   <S:Body>
4     <ns2:supprimerUserResponse xmlns:ns2="http://idl1/">
5       <return>
6         <username>
7         ayoub
8       </username>
9     </return>
10   </ns2:supprimerUserResponse>
11 </S:Body>
12 </S:Envelope>
```

# Captures Postman

## Vérification avec la méthode "afficherUser"



## Resultat:

