Université Abderrahmane Mira de Bejaia
Faculté des Sciences Exactes
Département d'informatique
Année Universitaire : 2017/2018

2ème Année Master GL – ReSyD - IA (S3) Module : **Services Web**

TP 05 : Invocation d'un Service Web SOAP

Objectif :
Le but de ce TP est de voir l'invocation d'un Service web, le fichier WSDL et l'échange de message avec le protocole SOAP. Nous utilisions le langage Java ainsi que la plateforme NetBeans pour créer, déployer et invoquer le Service Web.

Etape 01 : **Création et déploiement du Service Web**
Nb : Se référer au TP03 pour voir comment créer une application Service Web. Voilà un extrait du code d'un service Web qui contient une seule méthode avec un paramètre « chaine de caractères » et elle renvoie cette chaine.

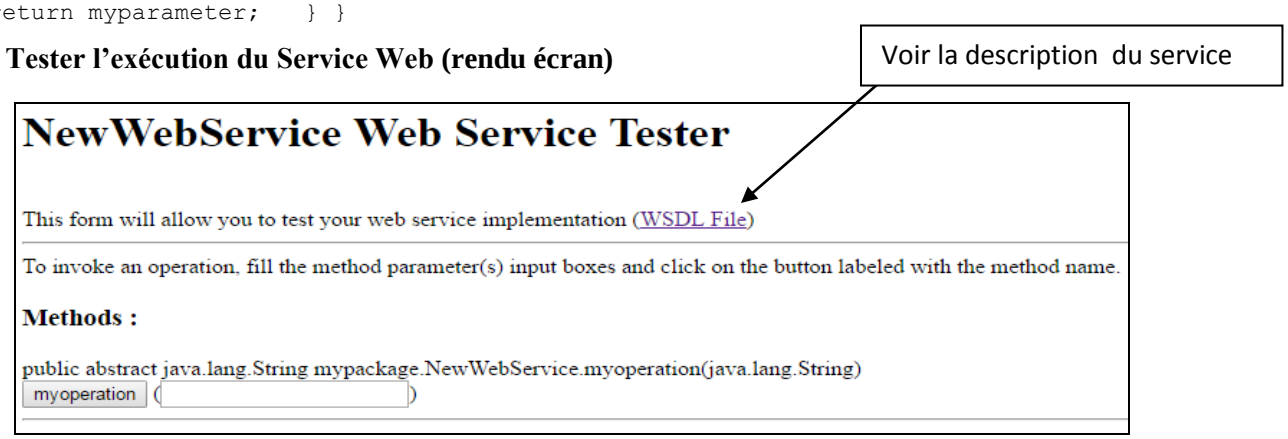
/ /Fichier : NewWebService.java

```
package mypackage;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.ejb.Stateless;

@WebService(serviceName = "NewWebService")
@Stateless()
public class NewWebService {

 @WebMethod(operationName = "myoperation")
 public String myoperation(@WebParam(name = "myparameter") String myparameter) {
        //TODO write your implementation code here:
        return myparameter;    } }
```

Etape 02 : **Tester l'exécution du Service Web (rendu écran)**



Fichier de description 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: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/wsdl/soap/"xmlns:tns="http://
mypackage/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
 targetNamespace="http://mypackage/" name="NewWebService">
<types>
<xsd:schema>
<xsd:import namespace="http://mypackage/" schemaLocation="http://localhost:8080/NewWebService/New
WebService?xsd=1"/>
</xsd:schema>
</types>
<message name="myoperation">
<part name="parameters" element="tns:myoperation"/>
</message>
<message name="myoperationResponse">
<part name="parameters" element="tns:myoperationResponse"/>
</message>
```

```
<portType name="NewWebService">
<operation name="myoperation">
<input wsam:Action="http://mypackage/NewWebService/myoperationRequest" message="tns:myoperation"/
>
<output wsam:Action="http://mypackage/NewWebService/myoperationResponse" message="tns:myoperation
Response"/>
</operation>
</portType>
<binding name="NewWebServicePortBinding" type="tns:NewWebService">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
<operation name="myoperation">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
</binding>
<service name="NewWebService">
<port name="NewWebServicePort" binding="tns:NewWebServicePortBinding">
<soap:address location="http://localhost:8080/NewWebService/NewWebService"/>
</port>
</service>
</definitions>
```

<u>Etape 03</u> : Invocation de la méthode « myoperation » du Service Web
Voilà le rendu écran de l'invocation de la méthode du Service. Analysez les échanges SOAP

myoperation Method invocation

Method parameter(s)

Type	Value
java.lang.String	Bonjour tout le monde

Method returned
java.lang.String : "**Bonjour tout le monde**"

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:myoperation xmlns:ns2="http://mypackage/">
            <myparameter>Bonjour tout le monde</myparameter>
        </ns2:myoperation>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
         <ns2:myoperationResponse xmlns:ns2="http://mypackage/">
            <return>Bonjour tout le monde</return>
        </ns2:myoperationResponse>
    </S:Body>
</S:Envelope>
```

- @WebService : annotation obligatoire, pouvant accepter des paramètres tels que serviceName ou portName
- @SOAPBinding : définit le style "document" ou le style "RPC"
- @WebParam : définit le nom d'un paramètre
- @WebResult : définit l'argument de retour