

TP 4 : Consommation d'un Service Web SOAP

Objectifs

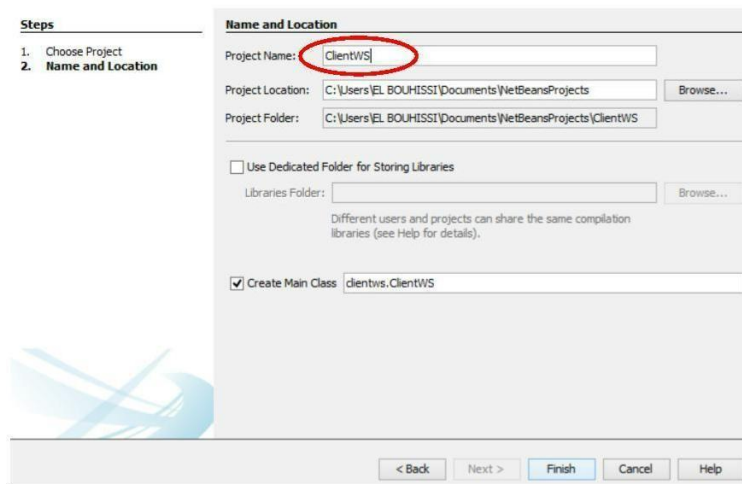
Dans ce TP, vous allez découvrir comment consommer un service Web SOAP.

Activité 01 :(Se référer au TP 2 pour créer un Service Web)

Il est demandé de créer un service web avec une opération addition (a : entier, b : entier) → entier.

Dans cette section, nous voyons comment consommer un web service à partir d'une application java puis à partir d'un script PHP. On peut considérer un consommateur de web service comme une application cliente (au sens de l'architecture Client-Serveur), ou bien comme un Mashup, c'est-à-dire une application utilisant des services provenant d'une ou de plusieurs sources pour proposer un contenu (ou un service).

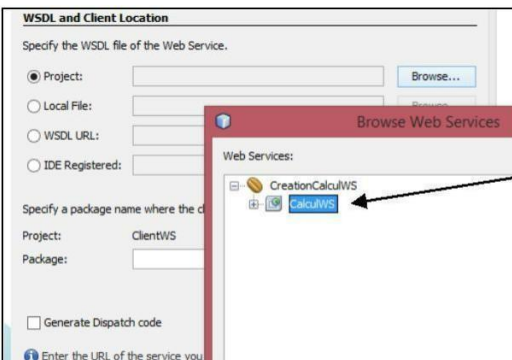
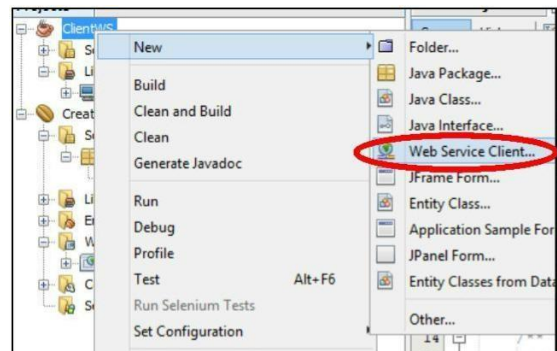
Commencez par créer un nouveau projet Java standard → Netbeans : création d'un projet Java



A l'intérieur du projet, nous créons un nouveau "web service client".

Nous sélectionnons ensuite un projet de type web-service afin d'importer sa définition (son WSDL) pour cela, choisissez le projet déjà créé.

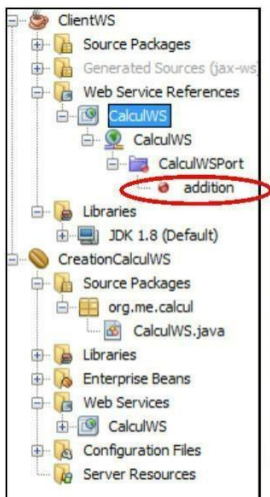
Il serait également possible de créer une application cliente à partir d'un WSDL accessible en ligne.



Votre service web (ou vous pouvez choisir un autre)

Vous pouvez choisir directement le projet, un WSDL local

en indiquant son chemin ou une URL.



Ensuite, nous pouvons choisir quelles sont les opérations que nous souhaitons invoquer dans notre application. Pour cela, nous déplaçons le dossier *Web Service References* dans le projet, puis à l'aide d'un glisser/déposer, nous insérons une des opérations directement dans le code.

Importation d'une opération

Une fois la méthode (static) importée, il ne reste qu'à l'invoquer dans le code de la méthode main, le point d'entrée de votre application :

```
public static void main(String[] args) {  
    System.out.println(addition (2,3));  
}
```

Invocation de l'opération dans la fonction main

Test du client

Après s'être assuré que le service web est toujours lancé, nous pouvons exécuter l'application cliente (Menu Run).

Voilà un extrait du code :

```
package clientws;  
/**  
 *  
 * @author EL BOUHISSI  
 */  
public class ClientWS {  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO  
        System.out.println(addition(2,3));  
    }  
  
    private static int addition(int a, int b) {  
        clientws.CalculWS_Service service = new clientws.CalculWS_Service();  
        clientws.CalculWS port = service.getCalculWSport();  
        return port.addition(a, b);  
    }  
}
```

Maintenant, créer une petite interface (le design selon votre choix), pour introduire les valeurs de paramètres (a et b) dans des cases et en cliquant sur un bouton, on déclenche l'opération d'addition.

Vous pouvez aussi plusieurs services web avec plusieurs opérations.

Consommer un Service Web via PHP

PHP propose en standard les bibliothèques permettant de consommer des Services Web à l'aide de la classe *SoapClient*. Une fois instanciée en indiquant l'URL du WSDL, la classe retournée représente l'objet distant qui propose le web service. Étant donnée la dynamique du langage PHP, les opérations proposées sont directement mappées au sein de cette classe. À noter, les éventuels arguments sont passés à travers un tableau associatif, dont les clés doivent correspondre exactement aux noms des paramètres spécifiés par le web service.

En partant du principe d'un serveur Apache/PHP est installé sur la machine :

(<http://easyphp.org>, <http://wampserver.com> , <http://www.mamp.info>), nous pouvons tester le code suivant :

```
<pre>  
< ?php  
$myservice = new SoapClient(" mettre ici l'URL de votre service ");  
$resultat = $myservice->addition(2,3) ;  
Echo " Le résultat est : ".implode(",",$resultat->return) ;  
.....
```

Question : réaliser un client de service avec le langage Python.

Activité 2 : Choisissez un Service WSDL de l'internet et créer une application client avec le langage de votre choix (par exemple servie de recherche Google ou autre).