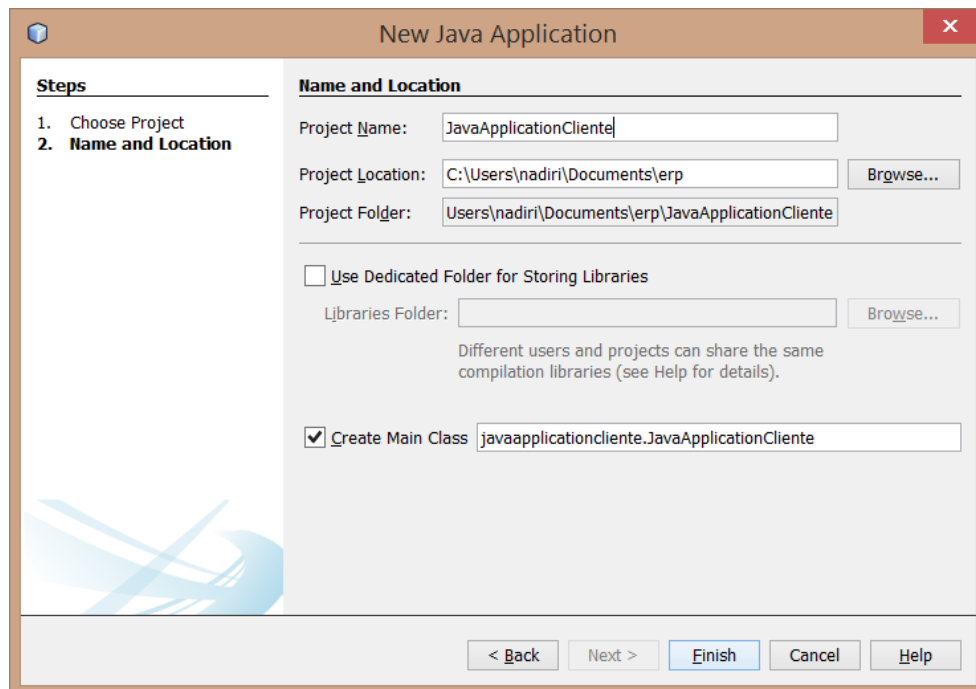


Consommer un web service

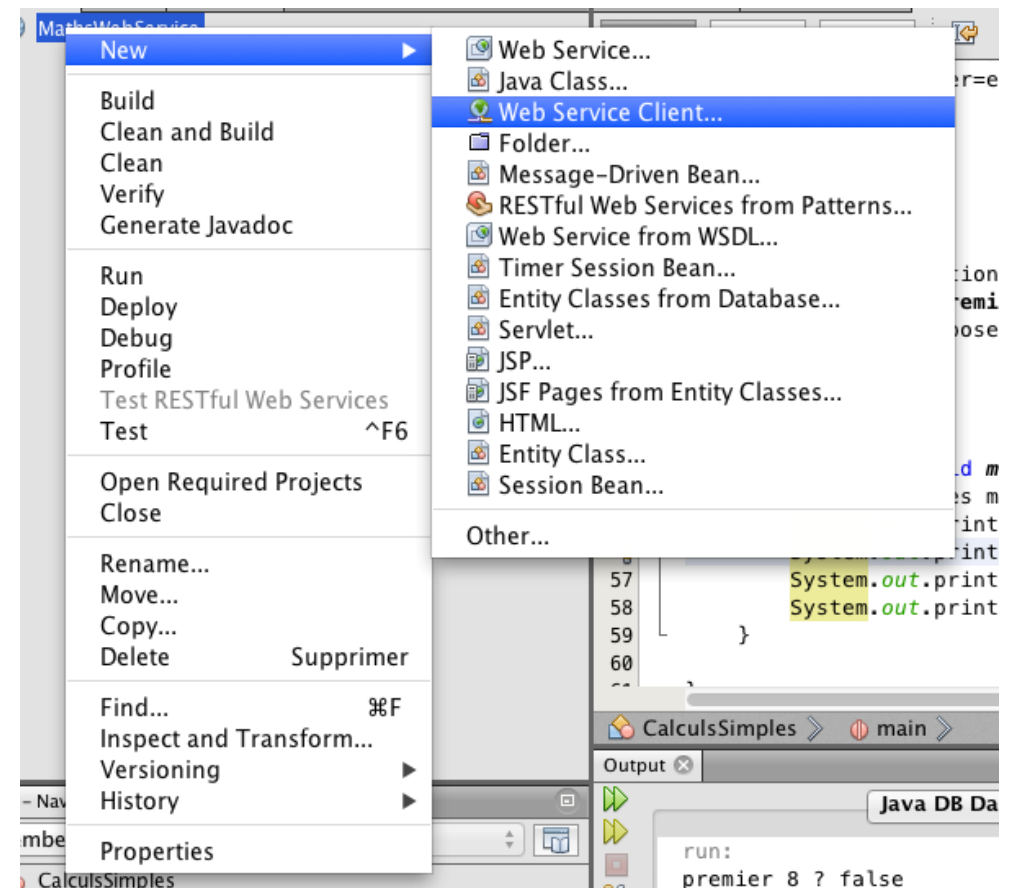
Dans cette section, nous voyons comment consommer un web service à partir d'une application java. On peut considérer un consommateur de web service comme une application cliente (au sens de l'architecture Client-Serveur) c'est-à-dire une application utilisant des services provenant d'une ou plusieurs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

Commençons par créer un nouveau projet Java standard.



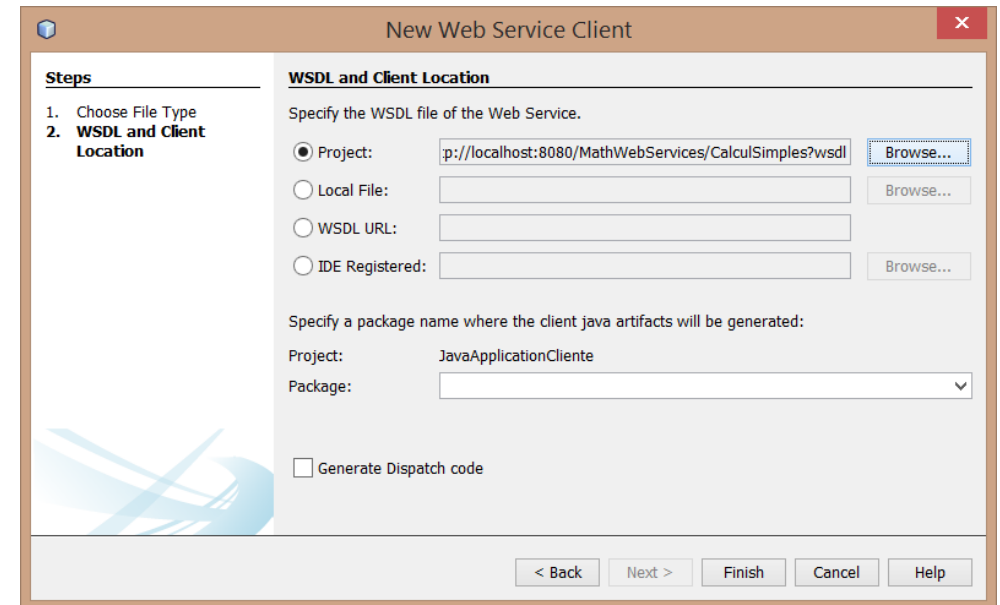
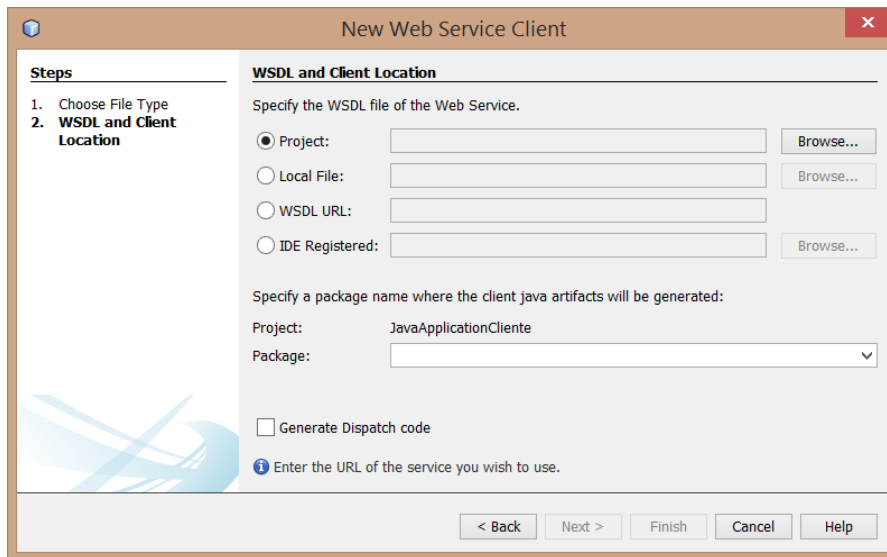
Netbeans : création d'un projet Java

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

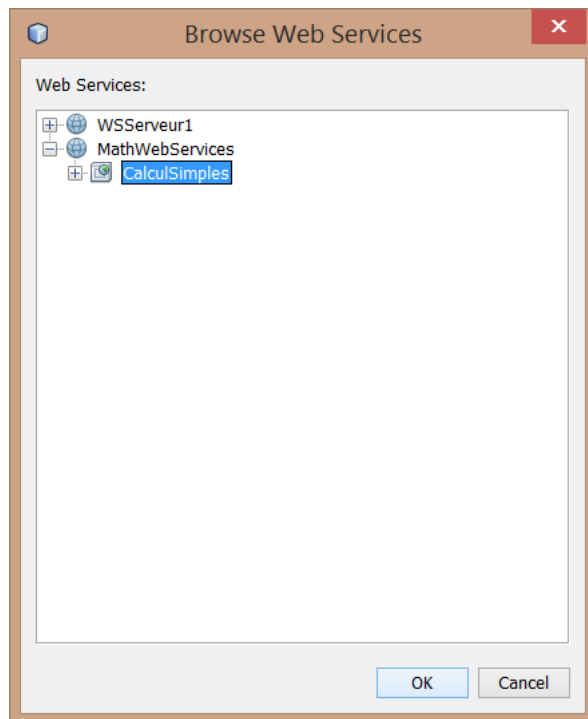


Création d'un client web-service

Nous sélectionnons ensuite un projet de type web-service afin d'importer sa définition (son WSDL). Il serait également possible de créer une application cliente à partir d'un WSDL accessible en ligne.

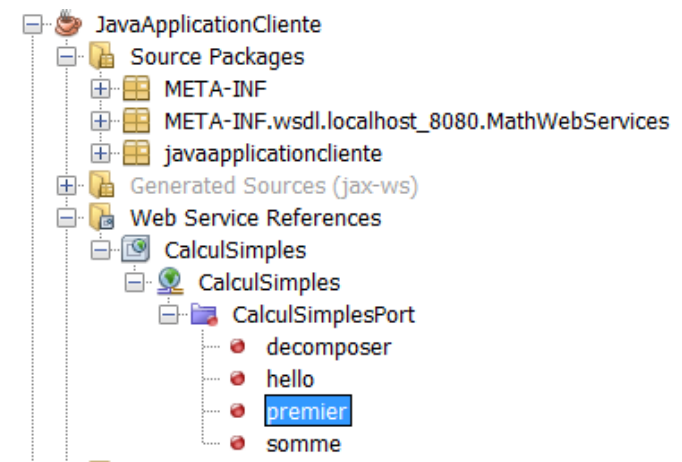


Choix du WSDL



Choix du projet

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.



```
public class JavaApplicationCliente {  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
    }  
  
    private static boolean premier(int entier) {  
        erp.CalculSimples_Service service = new erp.CalculSimples_Service();  
        erp.CalculSimples port = service.getCalculSimplesPort();  
        return port.premier(entier);  
    }  
}
```

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 notre application.

```
public static void main(String[] args) {  
    System.out.println(premier (501));  
}
```

Invocation de l'opération dans la fonction main.

Test du client

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

```
public class JavaApplicationCliente {  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        System.out.println("appel distant decomposer() :"+decomposer(642880));  
    }  
  
    private static java.util.List<java.lang.Object> decomposer(int entier) {  
        org.emiage.CalculSimples_Service service = new org.emiage.CalculSimples_Service();  
        org.emiage.CalculSimples port = service.getCalculSimplesPort();  
        return port.decomposer(entier);  
    }  
}
```



JavaApplicationCliente > main >

Java DB Database Process x GlassFish Server 4.0 x Retriever Output x JavaApplicationCliente (run) x

Created dir: /Users/david/Code/Workspace-Netbeans/D314-WS/JavaApplicationCliente/build/generated-sources
Compiling 11 source files to /Users/david/Code/Workspace-Netbeans/D314-WS/JavaApplicationCliente/build
Copying 3 files to /Users/david/Code/Workspace-Netbeans/D314-WS/JavaApplicationCliente/build/classes
compile:
run:
appel distant decomposer() :[2, 2, 2, 2, 2, 2, 5, 7, 7, 41]
BUILD SUCCESSFUL (total time: 2 seconds)

Lancement du client et invocation du web service

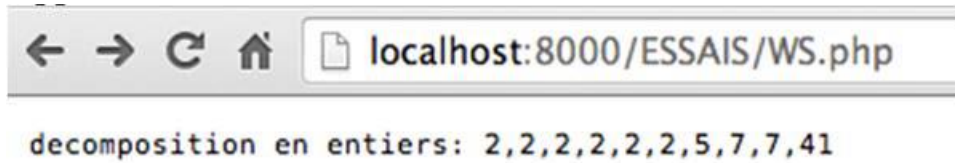
Consommer un web service via PHP

PHP propose maintenant en standard les bibliothèques permettant de consommer des web services 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 installé sur la machine (<http://easyphp.org>, <http://wampserver.com>, <http://www.mamp.info>), nous pouvons tester le code suivant.

```
1 <pre>
2 <?php
3 $mathservice = new SoapClient("http://localhost:8080/MathsWebService/CalculsSimples?wsdl");
4 $result = $mathservice->decomposer(array("entier"=>642880));
5 echo "decomposition en entiers: ". implode(",",$result->return);
6
7
```

Appel d'un web-service en PHP



Aperçu dans le navigateur