le cnam

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

LIBAN – BEYROUTH

Architectures Logicielles Java

SPECIALITE : Informatique

Par

Aless Hosry

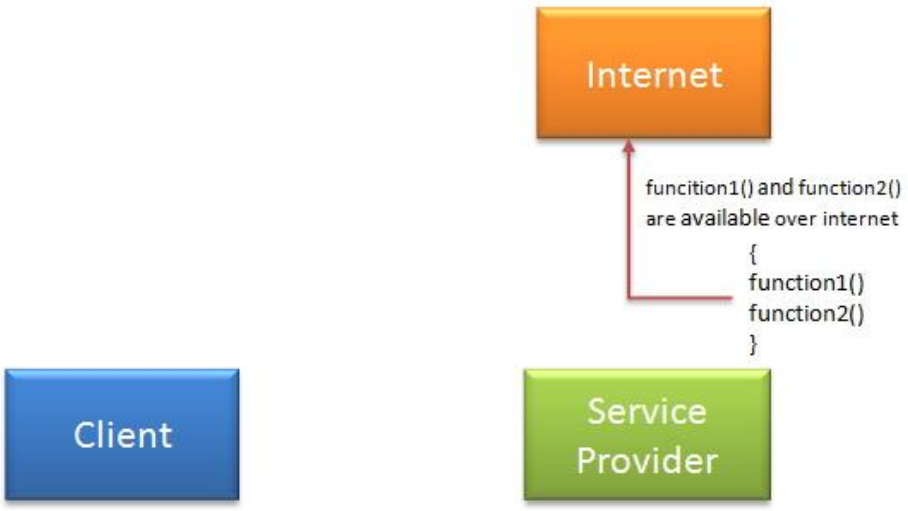
Service Web SOAP avec J2EE

TABLE DES MATIERES

1- Définition d’un service web

Un service Web est un ensemble de protocoles ouverts et standards utilisés pour l'échange de données entre applications ou systèmes. Les applications logicielles écrites dans différents langages de programmation et fonctionnant sur différentes plates-formes peuvent utiliser des services web pour échanger des données sur des réseaux informatiques comme l'Internet dans un semblable à la communication interprocessus sur la manière d'un seul ordinateur. Cette interopérabilité (par exemple, entre les applications Windows et Linux, Java et Python) est due à l'utilisation de normes ouvertes.

Les services Web sont alors des systèmes d'échange d'informations de types multiples (xml, Json ou text plain) qui utilisent l'Internet pour que l'interaction directe entre les applications. Ces systèmes peuvent inclure des programmes, des objets, des messages ou des documents.



2- Types de service web :

Il en existe 2 types de services web :

- SOAP (Simple Object Access Protocol)
- REST(Representational State Transfer)

La différence entre les deux est que le premier utilise le format XML pour échanger les informations, il est plus ancien et moins rapide que le REST, se caractérise par le fichier WSDL qui énumère toutes les méthodes de ce service web, et pour chacune quels sont les paramètres d'entrée sortie et leur types… alors que REST peut accepter des données de formats multiples : XML, JSON, Text/plain … Il est plus rapide et plus moderne… Dans ce cours nous allons expliquer c'est quoi un service web SOAP et nous allons donner un exemple d'un service SOAP écrit en Java sur Netbeans et appelé par un client écrit en vb.net. Les conditions nécessaires pour que notre exemple fonctionne sont les suivantes :

- Librairie Java
- Netbeans
- Serveur Tomcat
- Visual studio

3- Implémentation de service web SOAP avec J2EE

Simple Object Access Protocol (SOAP) est une spécification standard pour le protocole d'échange de messages basé sur XML. La communication entre le service web et le client se passe en utilisant des messages XML.

Une architecture de services Web simple a deux composantes :

- Client
- Prestataire de services web

Alors afin d'amener cette communication le client doit connaitre quelques informations :

- Emplacement du service web
- Fonctions disponibles, signatures et types de fonction de retour
- Protocole de communication
- Formats des données d'entrés et de sorties

Pour cela le développeur du service web va créer un fichier XML standard qui aura toutes les informations nécessaires. Une foi le client aura accès a ce fichier XML, appelé WSDL, il sera capable d'utiliser le service web.

a- C'est quoi WSDL

WSDL signifie Web Service Description Language. Il est un fichier XML décrivant les détails techniques de la façon de mettre en œuvre un service Web, l'URI plus précisément, port, les noms de méthode, les arguments et les types de données. Depuis WSDL est XML, il est à la fois lisible par l'homme et qui contribue à la possibilité d'appeler et de comprendre des choses comme :

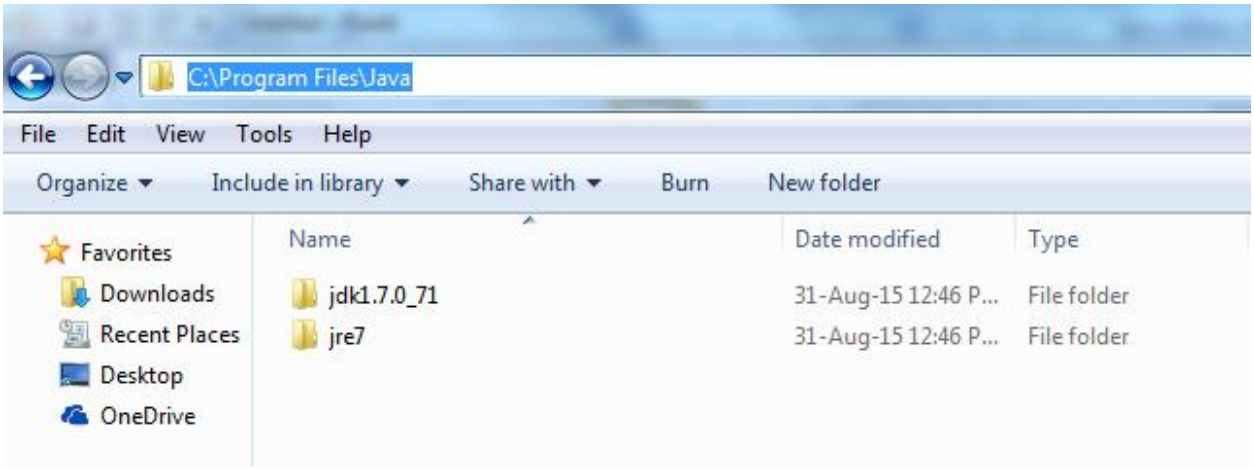
- Port / Endpoint - URL du service Web
- Format de message d'entrée
- Format de message de sortie
- Protocole de sécurité doit être celle suivie

- Quel protocole utiliser le service Web

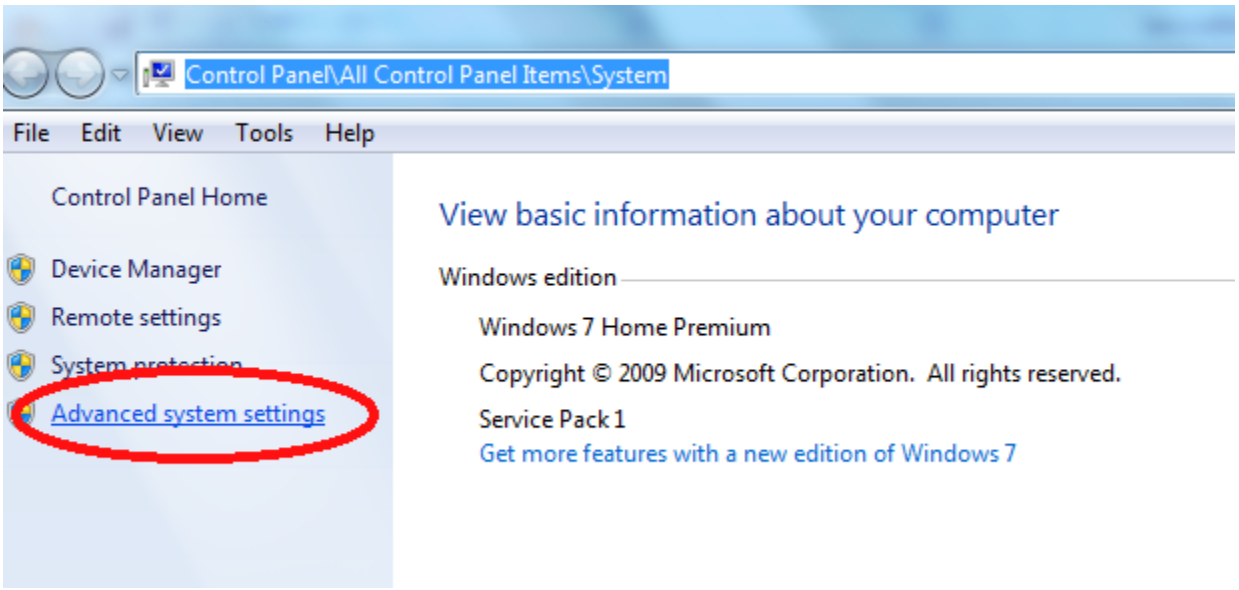
b- Exemple d'un service web SOAP en Java avec Tomcat

1- Tout d'abord il faut mettre en accord que la librairie java est installée sur notre PC, pour cela, pour une machine windows 7 64bit, on cherche tout d'abord dans le fichier Program Files le directoire suivante :

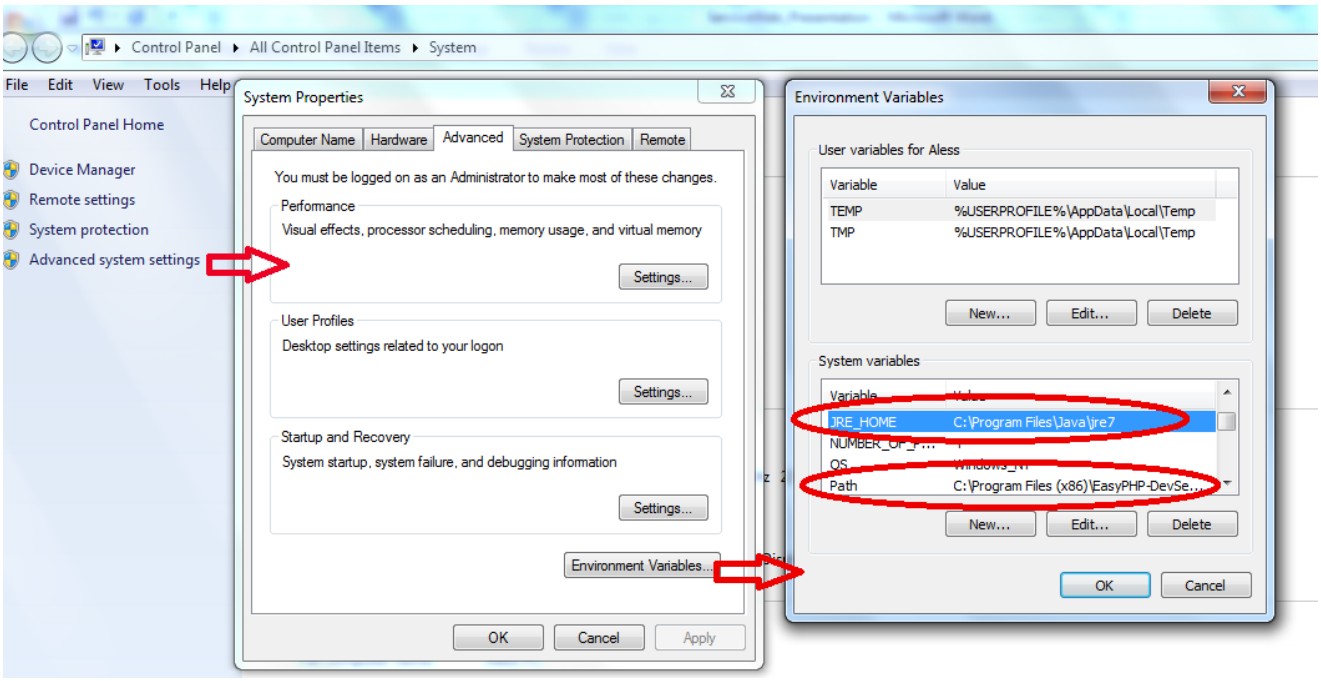
C:\Program Files\Java



Comme le fichier existe, on se met d'accord qu'il est enregistré et reconnu sur notre machine, pour cela il faut accéder dans control panel les réglages avancés :



Ensuite se mettre en accord des valeurs correctes des variables d'environnements enlevés de l'emplacement du fichier java dans C comme suit :



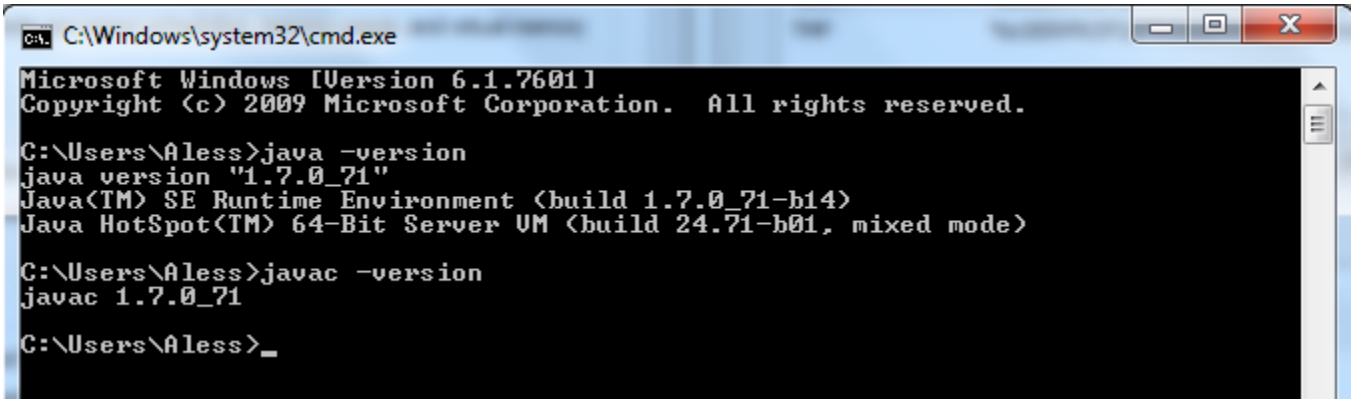
JRE_HOME: *C:\Program Files\Java\jre7*

Path: … (Ici il peut y exister quelques autres définitions séparées par des **;**) *;*

C:\Program Files\Java\jdk1.7.0_71\bin ;

Ensuite dans cmd on écrit les commandes suivantes : *Java –version* et *Javac –version*

Le résultat obtenu doit être la version de java affichée dans la première figure et entrée.



Tant que les résultats sont comme avant alors le processus est correct vous pouvez continuer à la deuxième phase.

2- Ensuite : enregistrer un serveur Tomcat sur la machine d'après le lien suivant : https://tomcat.apache.org/download-60.cgi

Apres le téléchargement de la version correspondante, extracter le fichier sur votre C sous l'emplacement suivant : *C:\apache-tomcat-6.0.48* et manipuler le fichier tomcat-users (*C:\apache-tomcat-6.0.48\conf\ tomcat-users.xml*) en ajoutant les données suivantes :

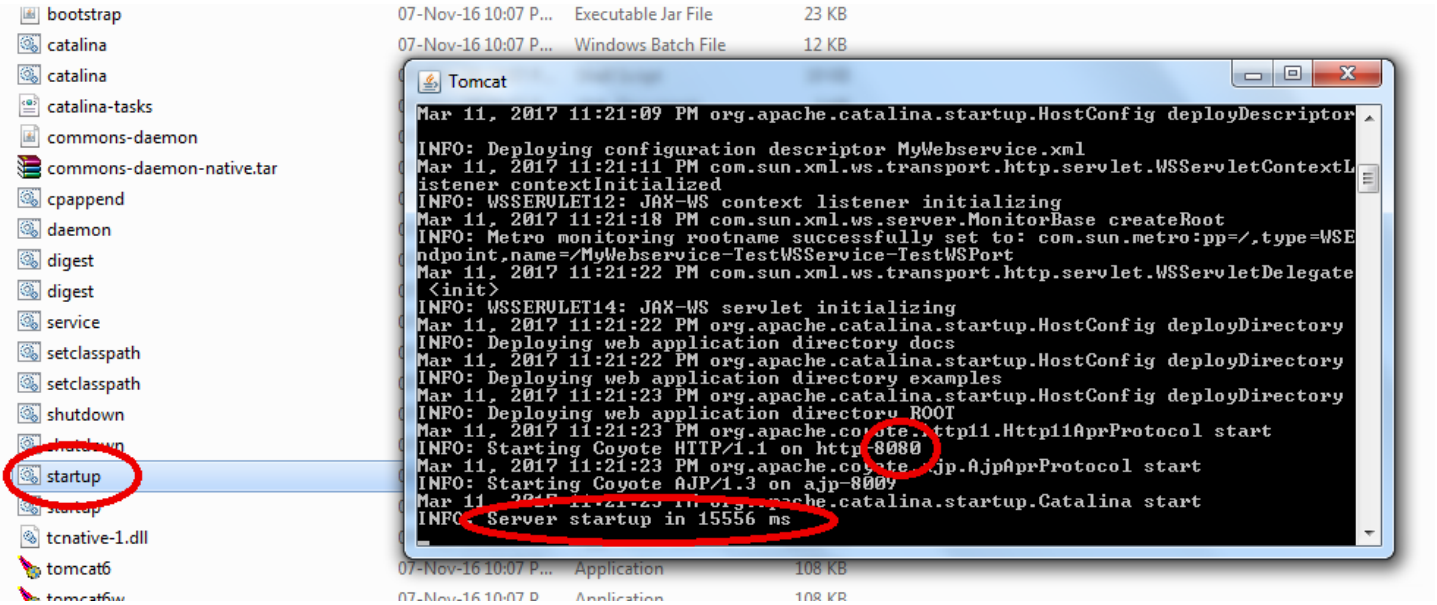
<Tomcat-users>

<role rolename="manager"/>

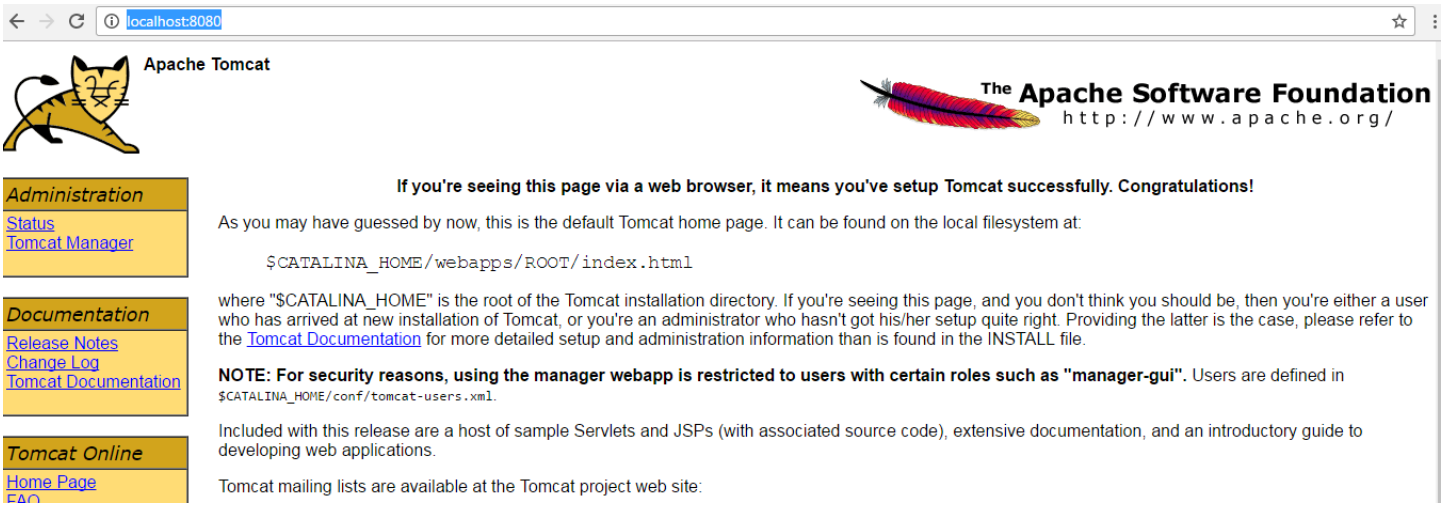
<user username="admin" password="admin" roles="manager"/>

</Tomcat-users>

Ces données seront nécessaires pour qu'on puisse créer un service web publié sur Tomcat. Pour déclencher le serveur on clique doublement sur le fichier *startup* dans *C:\apache-tomcat-6.0.48\bin*, ensuite le cmd s'ouvre en indiquant le numéro de port qui en général 8080



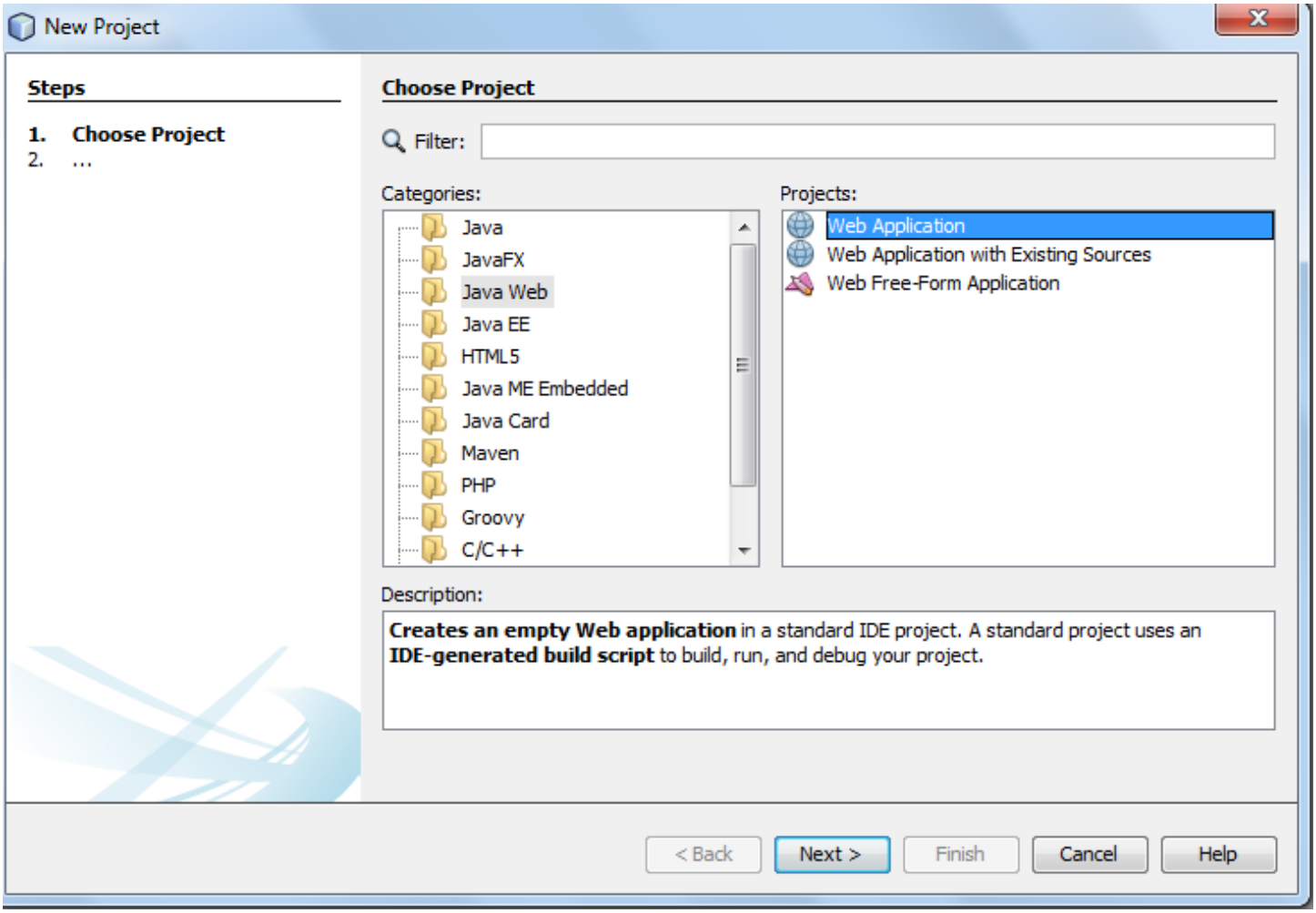
Pour s'assurer que Tomcat s'est déclenché, on ouvre un navigateur et on écrit le suivant : *http://localhost:8080/*. Si le suivant est affiché, alors tout est parfait.



Pour l'éteindre il suffit de taper Ctrl C dans CMD et cliquer doublement sur le fichier *shutdown* aussi dans *C:\apache-tomcat-6.0.48\bin*. Rafraichir la page au dessus, rien ne sera affiché.

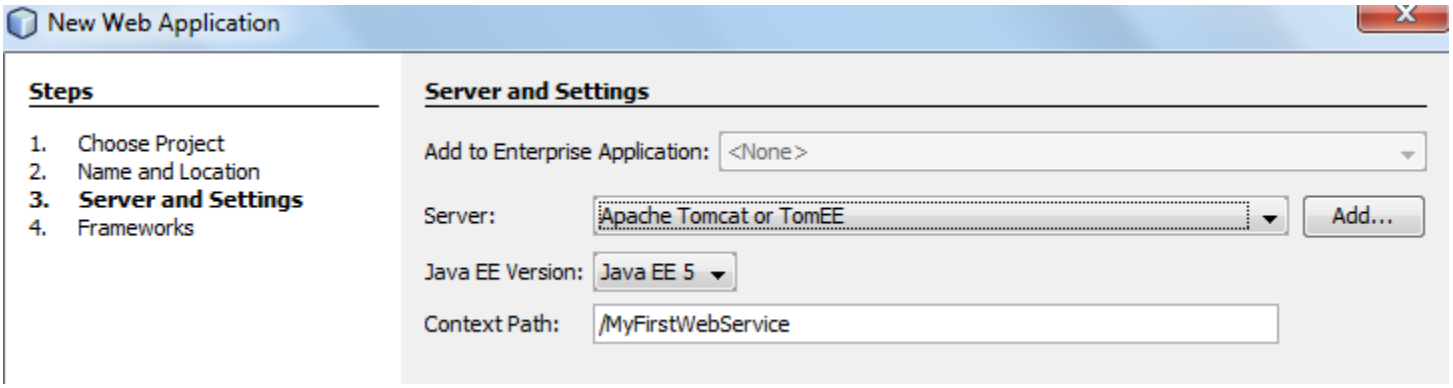
3- Maintenant on va créer notre service web sur netbeans (version 8.0.1) déjà téléchargé …

Tout d'abord on crée un nouveau projet qui sera une application web :



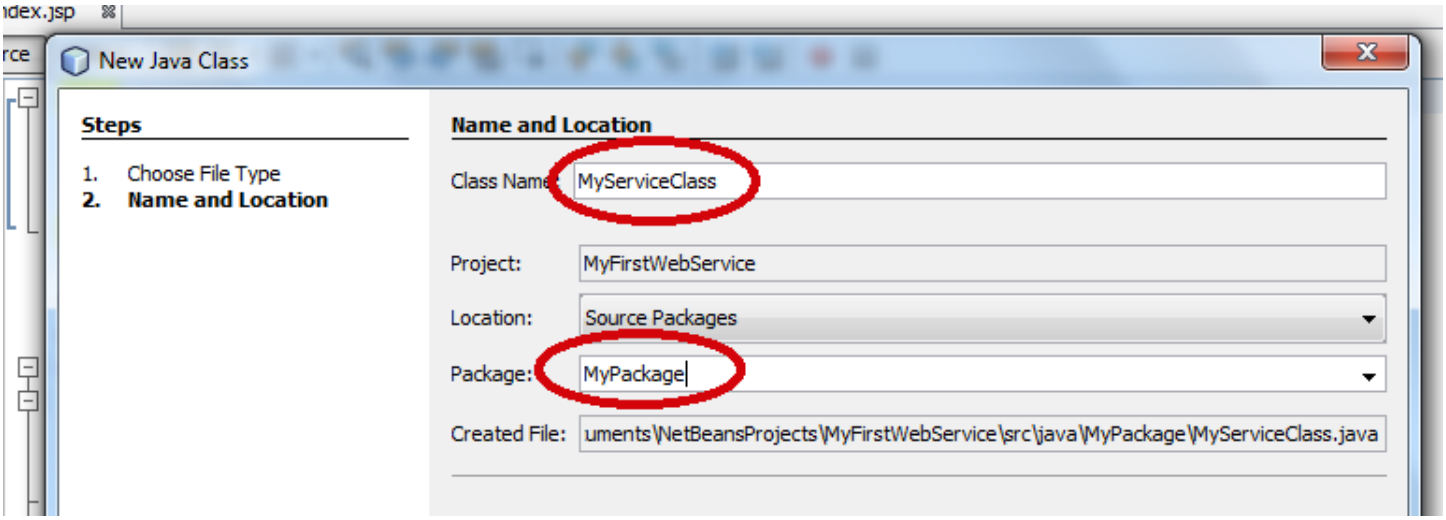
Dans l'étape suivante on nomme le projet comme suit : *MyFirstWebService*.

Apres cela on choisit le serveur Tomcat où on va déployer notre service web.



Apres on dépasse l'étape suivante pour finir la création.

Un nouveau projet sera créé et par défaut il aura une page index.jsp . On crée ainsi une nouvelle classe dans ce projet en l'implémentant dans un paquet spécifique :



Ensuite en utilisant la librairie JAX-WS (Java Api XML - Web Service) on ajoute l'attribut *@webservice* au début de la page (avant la création de la classe) qui donne une signification que cette classe est un service web et non pas une classe normale. Et puisqu'on parle ici d'un service web SOAP on ajoute aussi l'attribut *@SOAPBinding(style=SOAPBinding.Style.RPC)*. On peut ainsi commencer par la création des méthodes du service web. Pour chacune on va ajouter aussi avant sa création un attribut *@WebMethod* pour que cette méthode sera vu par le client et pour qu'il puisse l'utiliser. Cela signifie que le service web peut contenir des méthodes

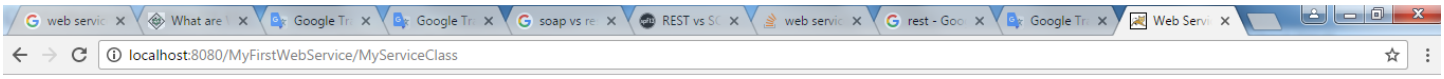
non publiées mais implémentées dans d'autres méthodes vues par les clients. Voici un exemple d'un service web SOAP créé sur NetBeans :

```
package MyPackage;

import javax.jws.WebMethod;
import javax.jws.WebService;
import javax.jws.soap.SOAPBinding;
/**
 *
 * @author Aless
 *
 */
@WebService
@SOAPBinding(style=SOAPBinding.Style.RPC)
public class MyServiceClass {

    @WebMethod
    public int AddNumbers(int a,int b){
      return a+b;
    }
}
```

Maintenant on peut déclencher notre service web pour voir ce qui se passe, mais avant tout il faut déclencher le serveur Tomcat comme on a déjà expliqué avant. Apres de cliquer sur le bouton RUN le navigateur s'ouvre en affichant la page index.jsp par défaut, pour accéder alors notre service web il s'suffit d'ajouter le nom du service web (nom de la classe) à l'URL :

*http://localhost:8080/MyFirstWebService/**MyServiceClass***

Si la page suivante est affichée alors félicitations, vous avez créé un service web SOAP avec J2EE.



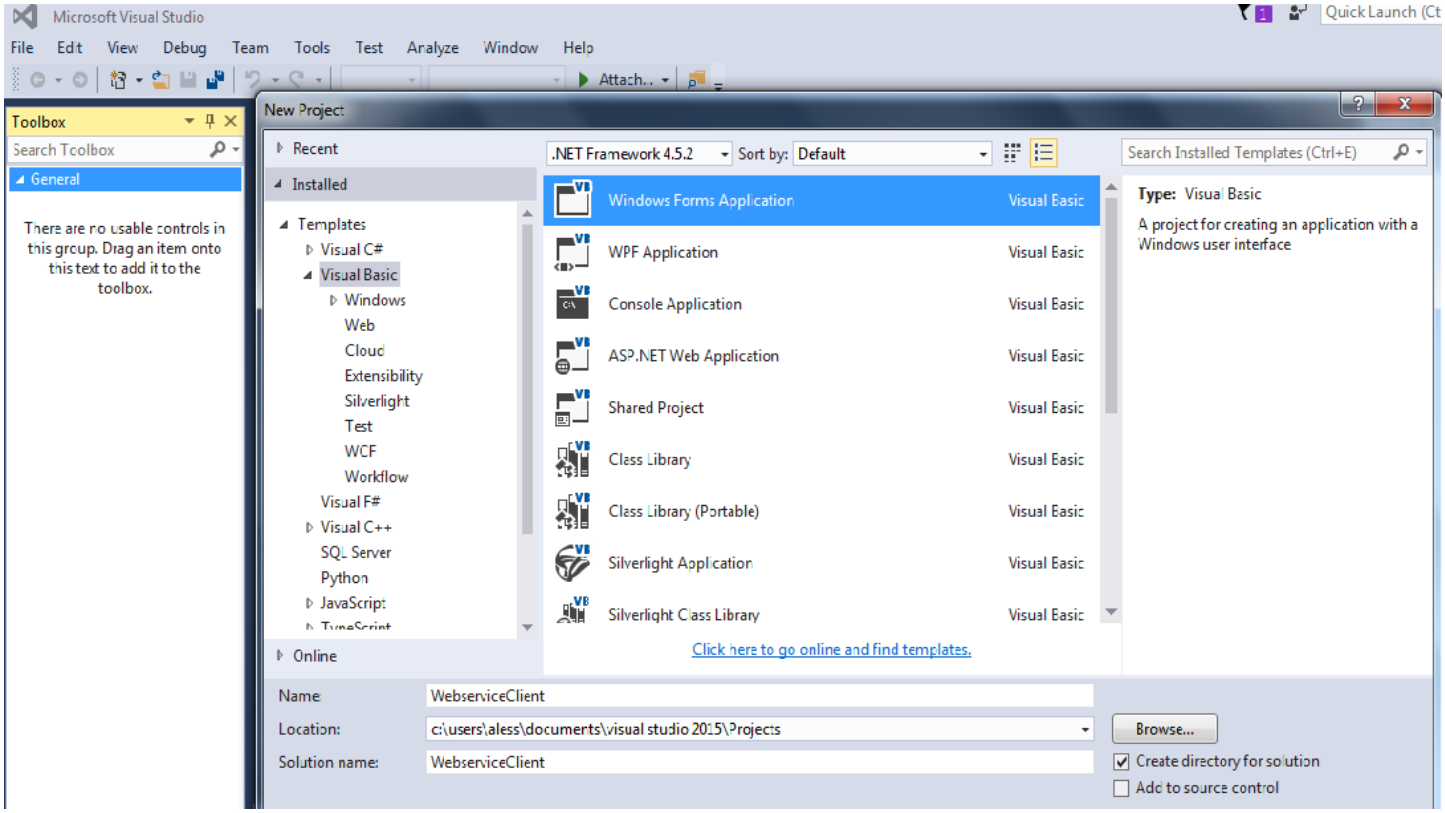
Web Services

Endpoint	Information
Service Name: {http://MyPackage/}MyServiceClassService Port Name: {http://MyPackage/}MyServiceClassPort	Address: http://localhost:8080/MyFirstWebService/MyServiceClass WSDL: http://localhost:8080/MyFirstWebService/MyServiceClass?wsdl Implementation class: MyPackage.MyServiceClass

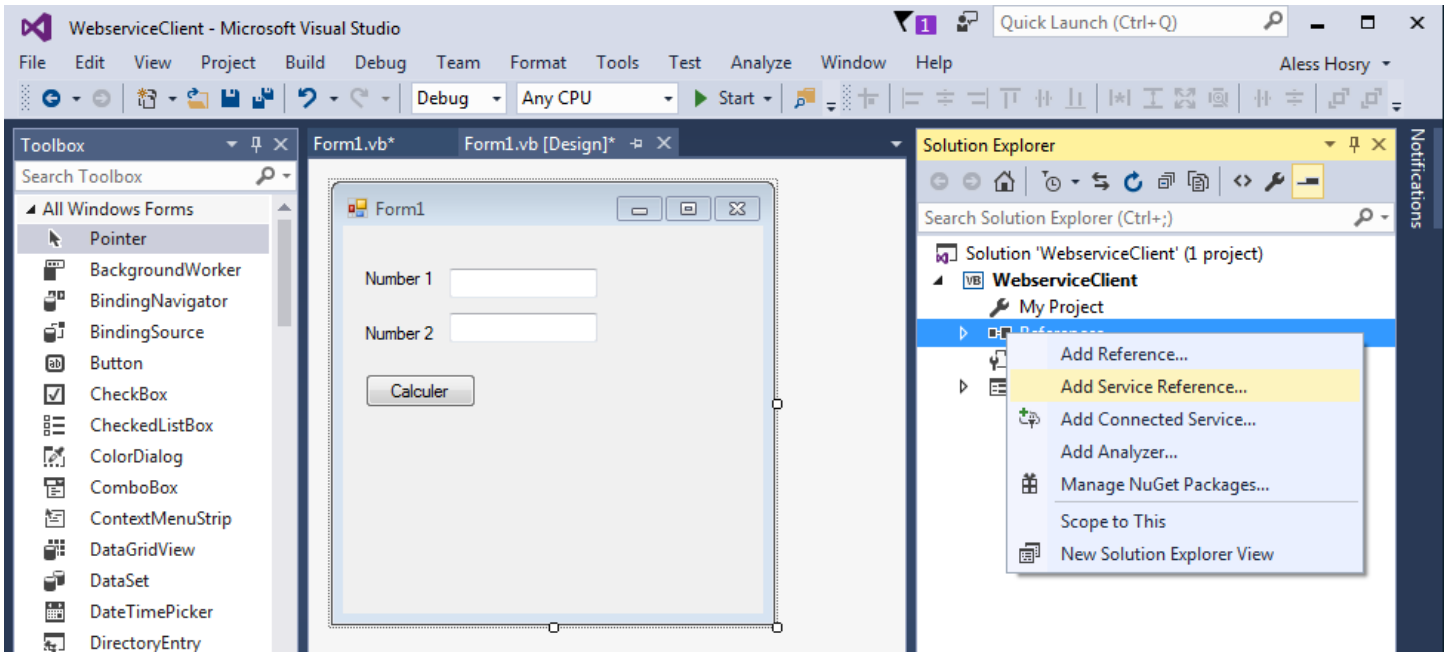
Exemple de fichier WSDL :

4- Exemple client .Net utilisant SOAP

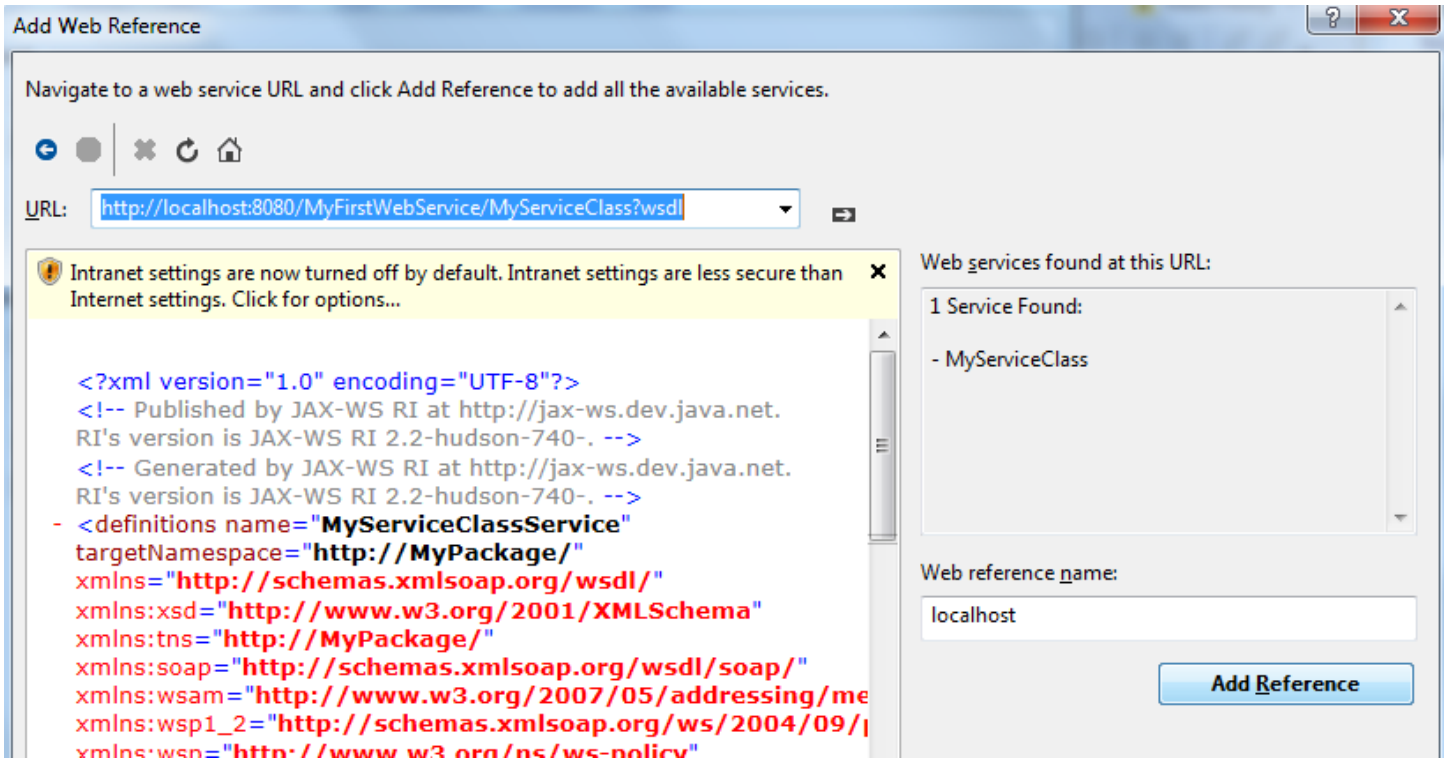
Tout d'abord on ouvre Visual Studio et on crée un nouveau projet :



Tout d'abord on crée une simple forme qui peut accepter deux nombres et un bouton qui va calculer leur somme en appelant la méthode du service web déjà créée… Pour cela on ajoute le service aux références du projet de la manière suivante :

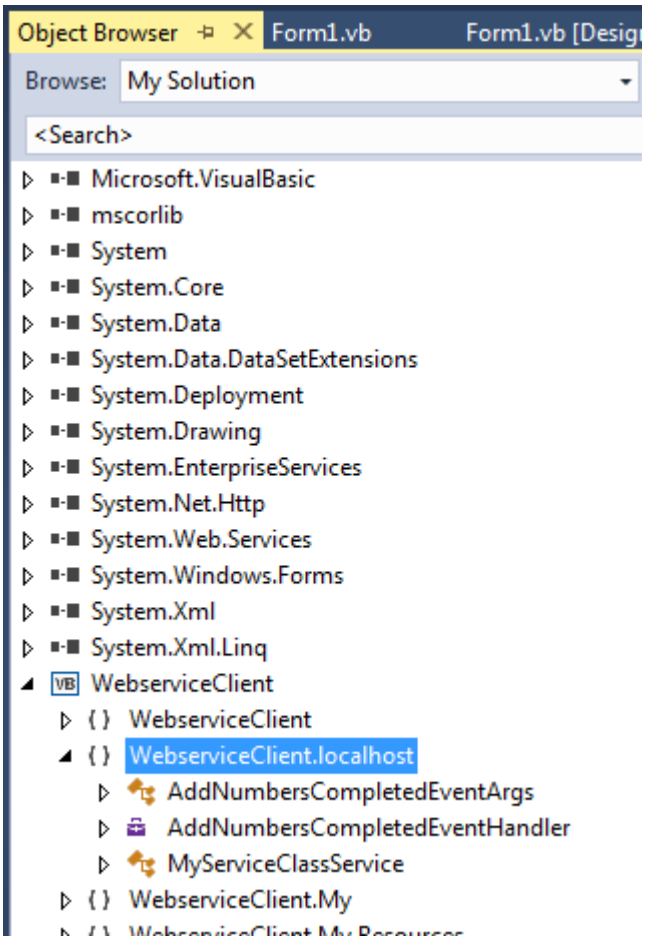


Une nouvelle fenêtre s'ouvre, on clique sur le bouton *Advanced* au dessus de la page, ensuite sur le bouton *Add Web Reference*, dans la 3eme nouvelle fenêtre qui s'ouvre on ajoute le lien de notre service web du fichier WSDL *'http://localhost:8080/MyFirstWebService/MyServiceClass ?wsdl'*, et on clique ainsi sur le bouton GO qui est une petite flèche à droite du lien. En cas de réussite vous verrez l'interface suivante :



Cliquez alors sur Add Reference, ainsi les méthodes et les classes (entités) du service web seront générées et ajoutées au projet afin de les utiliser dans les méthodes du client.

Apres l'ajout de la nouvelle référence on clique sur localhost (nom donné à la référence créée) au-dessus de webreferences qui apparait nouvellement dans la solution. Cela va ouvrir une nouvelle fenêtre titrée 'Object Browser' qui va lister le nom du service web concaténé avec le mot 'service', ajoutons les méthodes et les classes de ce service web.



Ainsi dans la méthode qui gère le bouton de calcule on peut créer une instance du service web et utiliser la méthode ‘AddNumbers’ :

```
Imports WebserviceClient.localhost
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim s As New MyServiceClassService
        Dim number1 = TextBox1.Text
        Dim number2 = TextBox2.Text
        Dim result = s.AddNumbers(number1, number2)
        Label3.Text = result
    End Sub
End Class
```

Déclenchez le projet par le bouton run et amusez-vous avec les résultats obtenus !

Si l'utilisateur ajoute une nouvelle méthode au service web, il suffit déclencher le service web de nouveau afin d'avoir un nouveau fichier wsdl … chez le client il suffit de cliquer sur nom de la reference → Update web reference , ainsi les nouvelles méthodes seront affichées dans Object Browser.

Pour plus d'informations vous pouvez visiter les liens suivants :

https://www.youtube.com/watch?v=fE1pVSiXNkU