

WCF

Programmation .Net avec C#

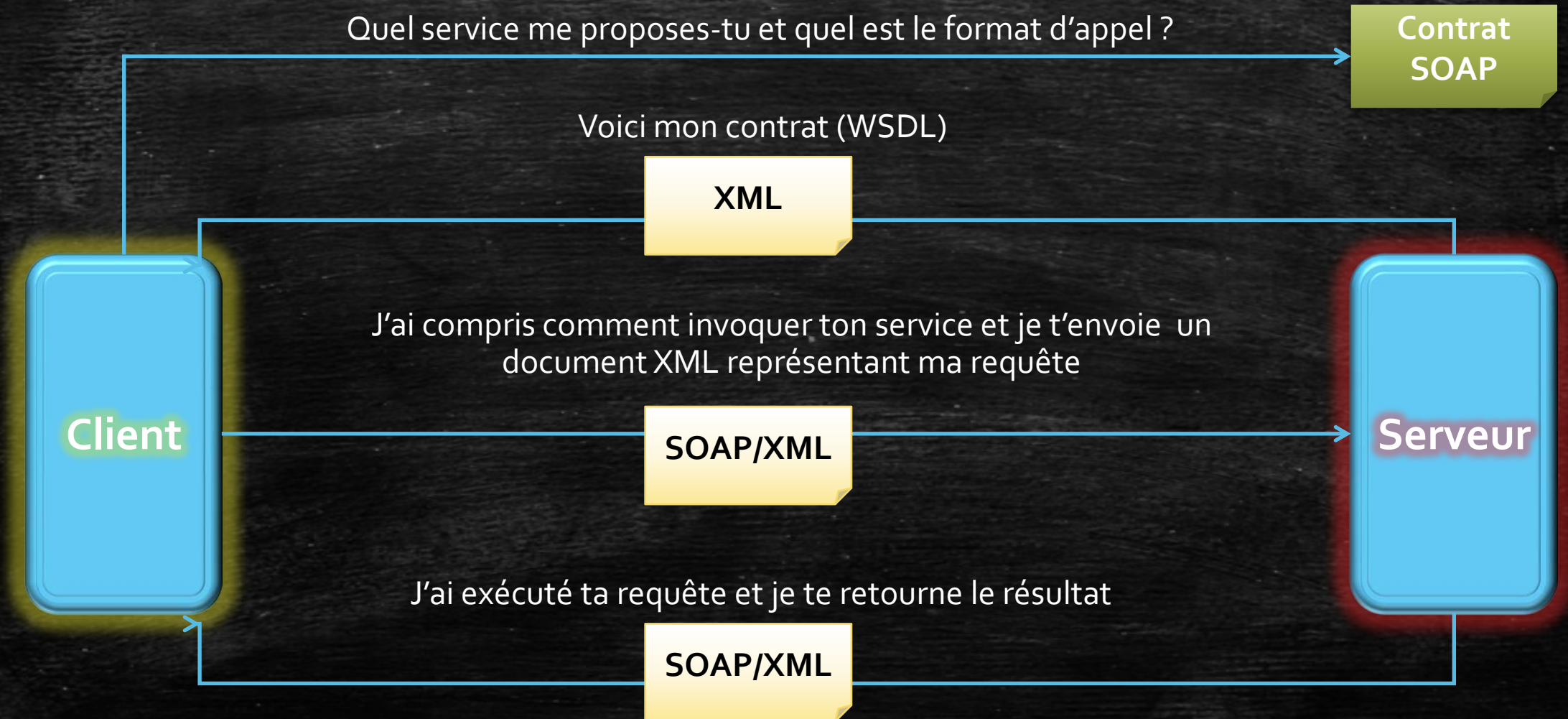
Présentation

- Définition :
 - Un service web est une unité logique applicative (un composant) accessible en utilisant les protocoles standards d'internet.
- Caractéristiques :
 - Réutilisable
 - Indépendant de :
 - La plate-forme (Unix, Windows, ...).
 - L'implémentation (VB, C#, Java, PHP, ...).
 - L'architecture sous-jacente (.Net, J2EE, ...).

Présentation

- Utilité :
 - Besoin d'interopérabilité entre systèmes et technologies hétérogènes.
 - Avec les applications RIA (Rich Internet Application) comme Silverlight et Flex.
 - Echange de documents sous forme XML.
 - Les applications « Services Provider ».
 - ...

Etapes d'utilisation



WCF

- Windows Communication Foundation.
 - Technologie qui permet de faciliter la mise en place des applications distribuées en servant de socle commun aux architectures orientées services (SOA : Service Oriented Architecture).
 - Modèle uniforme de distribution de services adapté à toutes situations et besoins de communication :
 - Disponibilité.
 - Interopérabilité.
 - Sécurité.
- La mise en œuvre de WCF repose principalement sur 2 espaces de nom du Framework .NET :
 - System.ServiceModel
 - System.Runtime.Serialization

Étapes de mise en œuvre de WCF

- 1^{ère} étape :
 - Définition du contrat
- 2^{ème} étape :
 - Implémentation du contrat
- 3^{ème} étape :
 - Configuration du service
- 4^{ème} étape :
 - Hébergement du service

Définition du contrat de service WCF

- Le contrat du service est l'entité qui va :
 - Etre échangée entre le serveur et le client.
 - Permettre au client de savoir quelles sont les méthodes proposées par le service et comment les appeler.
- L'élaboration d'un contrat de service s'effectue au travers des 3 métadonnées suivantes :
 - ServiceContract
 - OperationContract
 - DataContract
 - DataMember

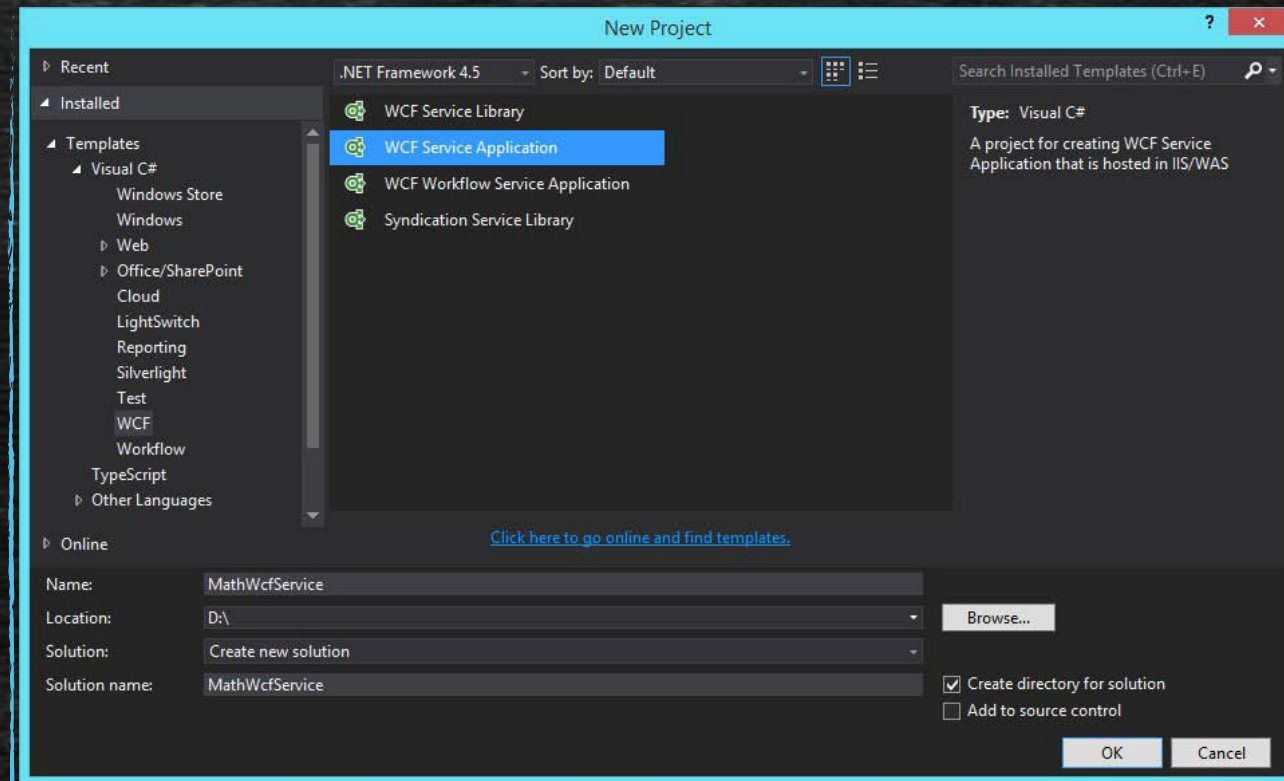
Définition du contrat de service WCF

- **ServiceContract**
 - Cette métadonnée est attachée à la définition d'une classe ou, d'une interface. Elle sert à indiquer que la classe ou l'interface est un contrat de service.
- **OperationContract**
 - Cette métadonnée est attachée aux méthodes que l'on souhaite exposer au travers du service WCF. Ainsi, il est techniquement possible de n'exposer au client que certaines méthodes d'une classe.
- **DataContract**
 - Cet attribut se place avant les classes qui définissent les objets que l'on va devoir passer en paramètre au service, ou que celui-ci va être retourné.
- **DataMember**
 - Cet attribut se place avant les propriétés des classes qui définissent les objets que l'on va devoir passer en paramètre au service, ou que celui-ci va être retourné.

Création d'un Service WCF

- Création d'un nouveau projet :
 - Sélectionner le langage « C# ».
 - Sélectionner le type de projets « WCF ».
 - Sélectionner, dans la liste des « Templates », le projet « WCF Service Application ».
 - Spécifier le nom du projet, de la solution, et l'emplacement physique.

Création d'un Service WCF

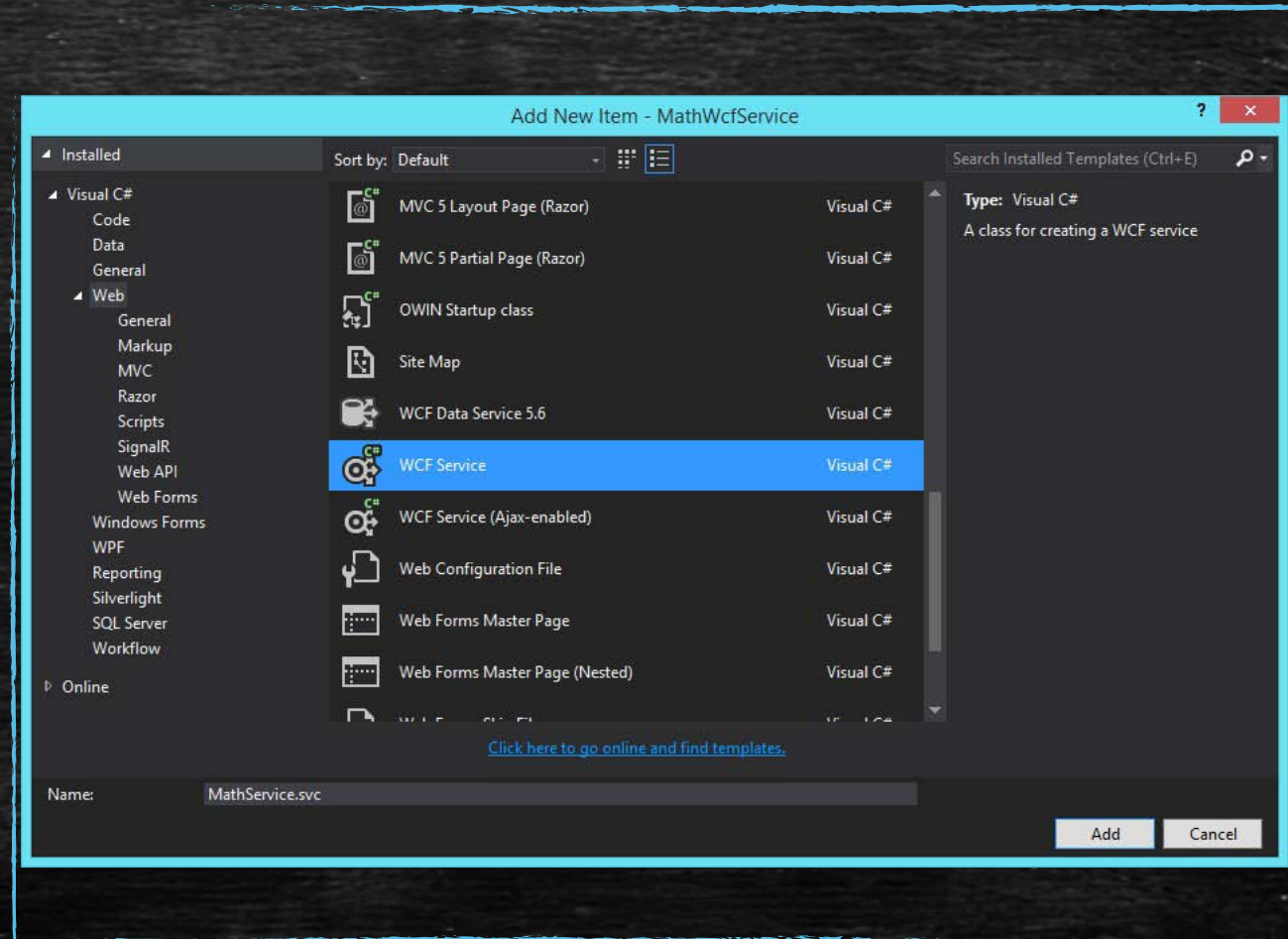


Exemple de création du projet
Service WCF « MathWcfService »

Création d'un Service WCF

- Création d'un nouveau fichier WCF Service :
 - Ajouter un nouvel élément au projet WCF Service.
 - Sélectionner le type d'éléments « Web ».
 - Sélectionner, dans la liste des « Templates », l'élément « WCF Service ».
 - Spécifier un nom pour le WCF Service.
 - Visual Studio crée en plus de la classe « WCF Service », une interface qui va contenir le contrat.

Création d'un Service WCF



Exemple d'ajout du WCF
Service « MathServices.svc ».

Création d'un Service WCF

- Ajout des « Web Methods » :
 - Lister les prototypes des méthodes du service dans l'interface du « WCF Service » en ajoutant l'attribut « ServiceContract » pour l'interface et «OperationContract » pour les méthodes.
 - Implémenter les méthodes de l'interface dans la classe « WCF Service ».

Création d'un Service WCF

```
IMathService.cs  X
MathWcfService.IMathService

using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

namespace MathWcfService
{
    [ServiceContract]
    1 reference
    public interface IMathService
    {
        [OperationContract]
        1 reference
        int Add(int a, int b);

        [OperationContract]
        1 reference
        int Subtract(int a, int b);

        [OperationContract]
        1 reference
        int Multiply(int a, int b);

        [OperationContract]
        1 reference
        int Divide(int a, int b);
    }
}
```

```
MathService.svc.cs  X
MathWcfService.MathService

using System;
using System.Runtime.Serialization;
using System.ServiceModel;

namespace MathWcfService
{
    0 references
    public class MathService : IMathService
    {
        1 reference
        public int Add(int a, int b)
        {
            return a + b;
        }

        1 reference
        public int Subtract(int a, int b)
        {
            return a - b;
        }

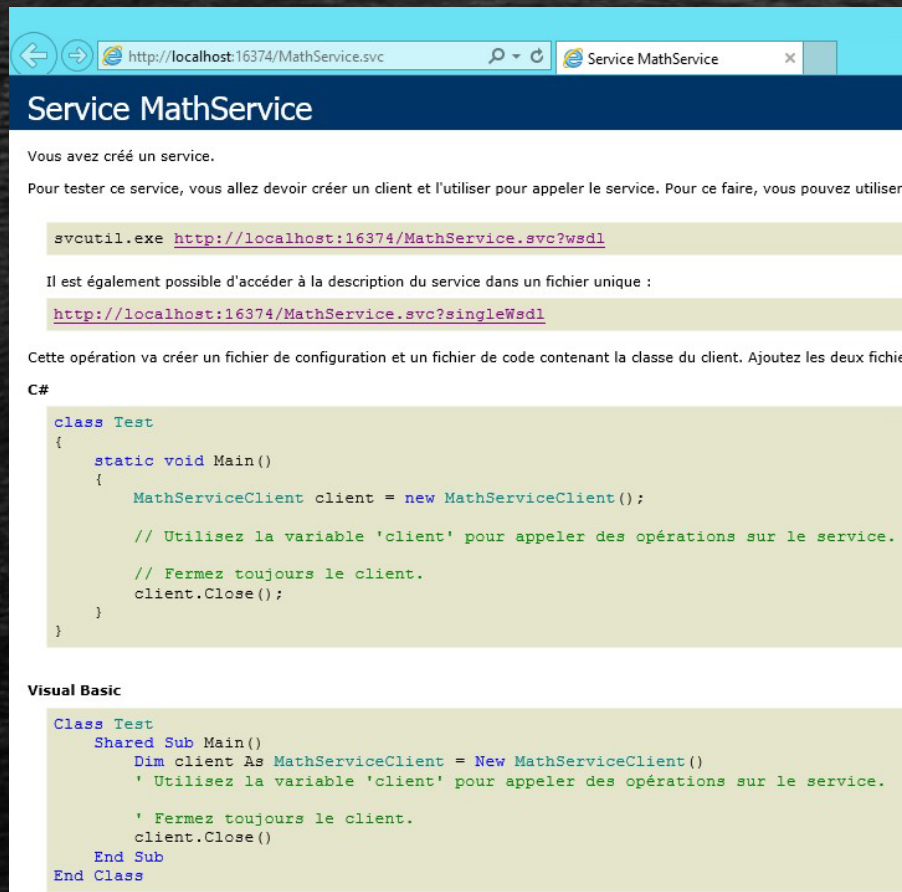
        1 reference
        public int Multiply(int a, int b)
        {
            return a * b;
        }

        1 reference
        public int Divide(int a, int b)
        {
            try
            {
                return a / b;
            }
            catch (DivideByZeroException)
            {
                return -1;
            }
        }
    }
}
```


Création d'un Service WCF

- Génération du Service WCF :
 - Compiler et générer le service WCF.
 - Explorer le service web :
 - Cliquer droit sur le fichier « .svc ».
 - Sélectionner l'option « Voir dans le navigateur (Internet Explorer) »
 - Examiner le service WCF.
 - Examiner la description WSDL du service.

Création d'un Service WCF



Service MathService

Vous avez créé un service.

Pour tester ce service, vous allez devoir créer un client et l'utiliser pour appeler le service. Pour ce faire, vous pouvez utiliser

```
svcutil.exe http://localhost:16374/MathService.svc?wsdl
```

Il est également possible d'accéder à la description du service dans un fichier unique :

```
http://localhost:16374/MathService.svc?singleWsdl
```

Cette opération va créer un fichier de configuration et un fichier de code contenant la classe du client. Ajoutez les deux fichiers à votre projet.

C#

```
class Test
{
    static void Main()
    {
        MathServiceClient client = new MathServiceClient();

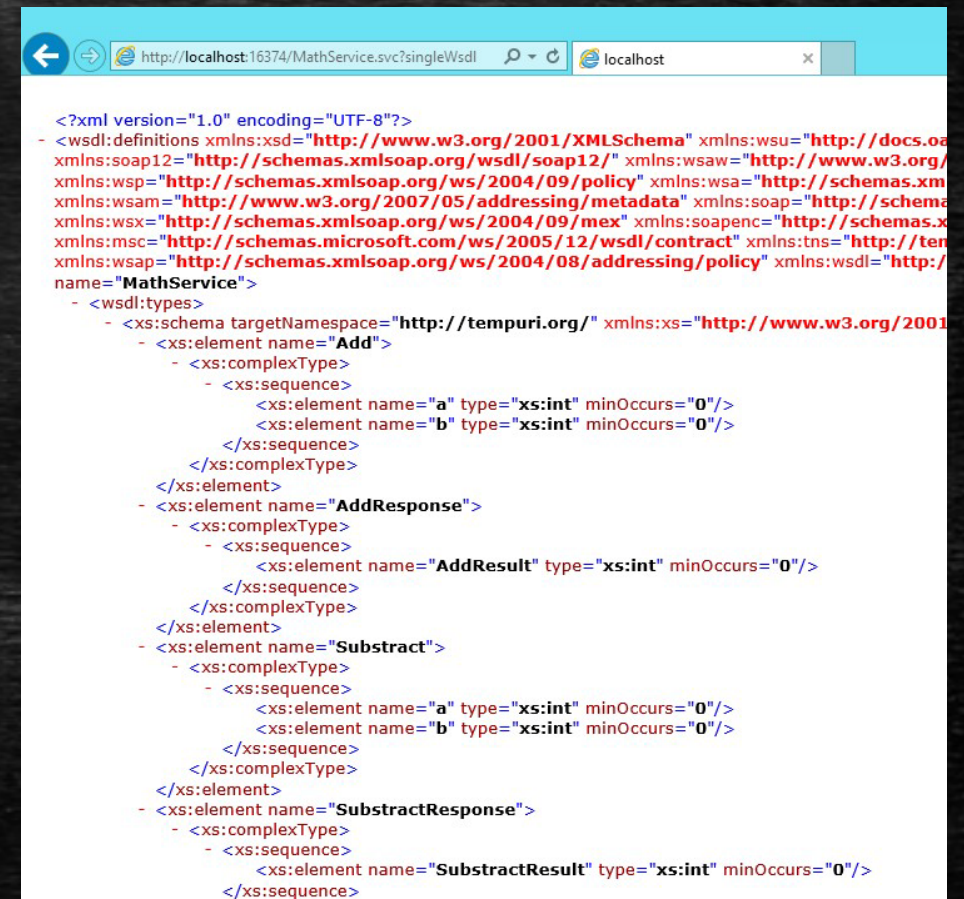
        // Utilisez la variable 'client' pour appeler des opérations sur le service.

        // Fermez toujours le client.
        client.Close();
    }
}
```

Visual Basic

```
Class Test
Shared Sub Main()
    Dim client As MathServiceClient = New MathServiceClient()
    ' Utilisez la variable 'client' pour appeler des opérations sur le service.

    ' Fermez toujours le client.
    client.Close()
End Sub
End Class
```



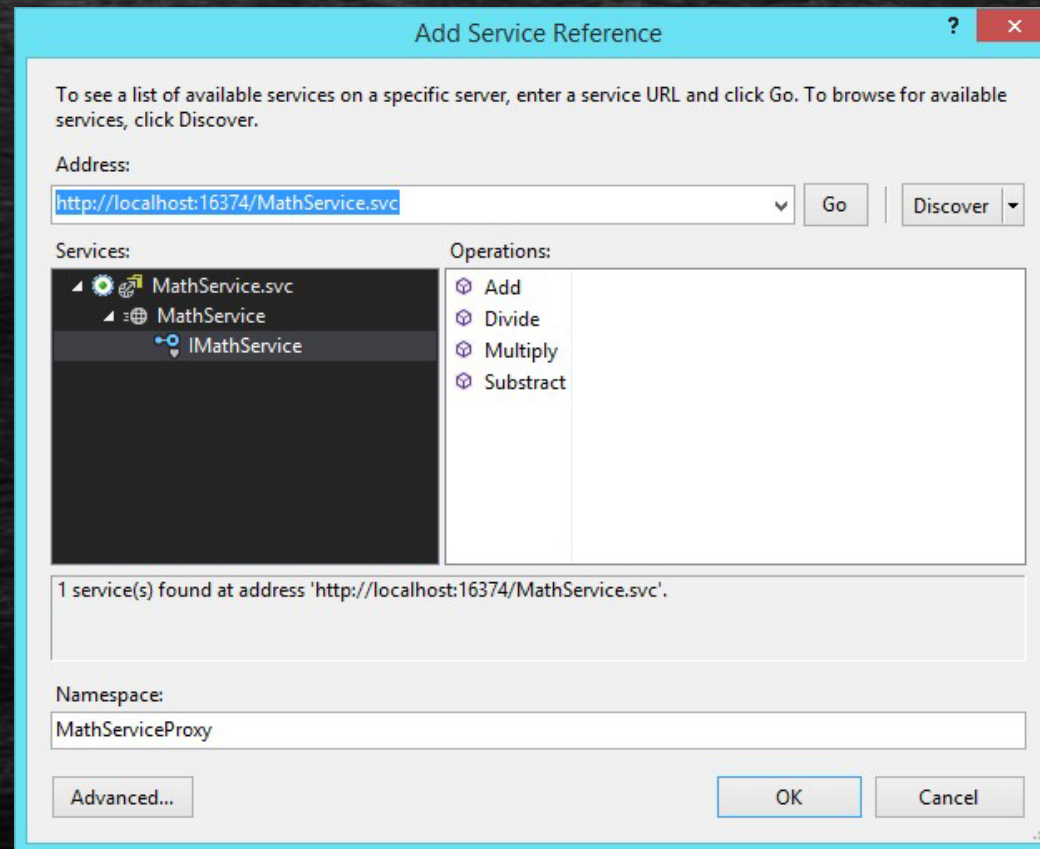
```
<?xml version="1.0" encoding="UTF-8"?>
- <wsdl:definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsu="http://docs.oa
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:wsaw="http://www.w3.org/
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsa="http://schemas.xml
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas
xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex" xmlns:soapenc="http://schemas.x
xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract" xmlns:tns="http://te
xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy" xmlns:wsdl="http://
name="MathService">
- <wsdl:types>
- <xs:schema targetNamespace="http://tempuri.org/" xmlns:xs="http://www.w3.org/2001
- <xs:element name="Add">
- <xs:complexType>
- <xs:sequence>
<xs:element name="a" type="xs:int" minOccurs="0"/>
<xs:element name="b" type="xs:int" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
- <xs:element name="AddResponse">
- <xs:complexType>
- <xs:sequence>
<xs:element name="AddResult" type="xs:int" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
- <xs:element name="Subtract">
- <xs:complexType>
- <xs:sequence>
<xs:element name="a" type="xs:int" minOccurs="0"/>
<xs:element name="b" type="xs:int" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
- <xs:element name="SubtractResponse">
- <xs:complexType>
- <xs:sequence>
<xs:element name="SubtractResult" type="xs:int" minOccurs="0"/>
</xs:sequence>
```


Utilisation du Service WCF

- Ajouter un « Service Reference » au projet qui va utiliser le service web.
 - Entrer l'adresse du Service Web ou utiliser l'option « Discover ».
 - Cliquer sur le bouton « Go ».
 - Spécifier un « Namespace » pour le « Service Reference ».

Utilisation du Service WCF

Exemple d'ajout de « Service Reference ».



Utilisation du Service WCF

- Utiliser les « Web Methods » du service Web :
 - Instancier le proxy du service web.
 - Utiliser les « Web Methods » du service.

Utilisation du Service WCF

```
MathServiceProxy.MathServiceClient proxy =  
    new MathServiceProxy.MathServiceClient();  
  
Console.Write(proxy.Add(10, 5));
```

Exemple d'instanciation du
proxy du service web et
utilisation de la « Web Method »