

Applications Réparties TD 3

Web Services et Windows Communication Foundation

Ce TD vous enseignera les bases de l'utilisation des Web Services en .NET avec Microsoft Visual Studio Express 2010 (<http://www.microsoft.com/visualstudio/en-us/products/2010-editions/express>) et Visual Web Developer Express 2010 (<http://www.microsoft.com/visualstudio/en-us/products/2010-editions/visual-web-developer-express>). Si besoin vous devrez mettre à jour votre .Net framework, la dernière version étant la 4.0 (<http://www.microsoft.com/downloads/fr-fr/details.aspx?FamilyID=9cfb2d51-5ff4-4491-boe5-b386f32c0992>) .

1 Création d'un client ASP.NET

Vous trouverez un grand nombre de Web Services disponibles sur le site <http://www.webservices.net/>. Nous allons utiliser le service « Currency Convertor » pour cet exercice.

Un fichier .asmx représente le point d'entrée adressable des services Web XML créés en code managé. La façon dont vous y accédez via le protocole HTTP détermine le type de réponse que vous obtenez.

Question 1 : Quelle est l'adresse du fichier wsdl du service « Currency Convertor » ? Pourquoi un Web service sous .Net correspond alors un fichier « .asmx » (Active Server Methods) ?

Question 2 : Dans le cadre d'un projet « Application Windows Forms », nous allons créer une simple application graphique qui permet d'afficher le taux de conversion à partir de deux monnaies. Une fois charger la « référence de service », quel namespace apparaît ? quelle classe est alors utilisable pour accéder aux méthodes offertes par le service ?

Question 3 : En lisant le fichier app.config, expliquez pourquoi l'exception suivante a été levée ?

<<

Une section de configuration du point de terminaison du contrat 'ServiceReference1.CurrencyConvertorSoap' n'a pas pu être chargée car plusieurs configurations de point de terminaison pour ce contrat ont été détectées. Indiquez la section de configuration du point de terminaison préférée par son nom.

>>

Quelles sont alors les deux solutions pour corriger votre projet ?

2 Développement de Services base sur Windows Communication Foundation (WCF)

2.1 Introduction

Windows Communication Foundation (WCF) est maintenant un composant du .Net framework depuis la version 3.0, permettant de réaliser des architectures orientées services (SOA). WCF est un tournant dans le développement de Services en .Net, rendant ainsi obsolète le développement direct de Services Web ASMX. Certains regretteront la simplicité du développement de Web Service ASMX. D'autres apprécieront l'abstraction logique fournie par WCF, liée à son abstraction.

WCF est l'un des trois piliers du .Net Framework 3.0 qui n'est autre qu'une « simple » extension du Framework .NET 2.0, auquel il ajoute principalement trois fonctionnalités :

- WWF (Windows Workflow Foundation) : apport de fonctionnalités concernant la gestion des Workflows avec une interface de création en mode Graphique dédiée.

Applications Réparties TD 3

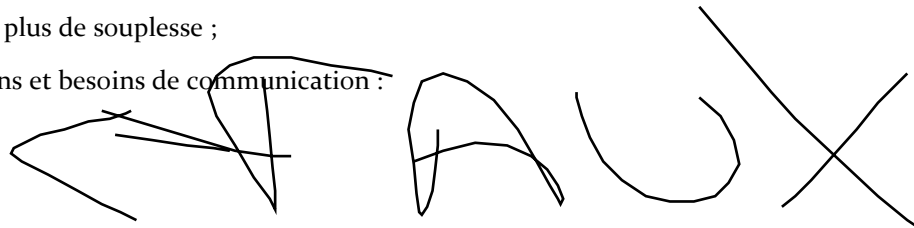
Web Services et Windows Communication Foundation

- WPF (Windows Présentation Foundation) : nouveau moteur de rendu pour les clients lourds Windows voué à remplacer les applications Winform.
- WCF (Windows Communication Foundation) présenté ici.
- WCS (Windows Card Space) : système de gestion d'identités par authentification unique.

WCF est une 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). WCF se veut être un modèle uniforme de distribution de services :

- Faiblement couplé pour plus de souplesse ;
- Adapté à toutes situations et besoins de communication :

- Disponibilité ;
- Interopérabilité ;
- Sécurité.



Les étapes de mise en œuvre de création d'un service WCF sont les suivantes :

- Première étape : définition du contrat. On définit les signatures des méthodes composant notre service WCF, et du format des données à échanger.
- Seconde étape : implémentation du contrat. On implémente les services.
- Troisième étape : configuration du service. On définit les endPoints, autrement dit les points d'accès au service.
- Quatrième étape : hébergement des services dans une application .NET.

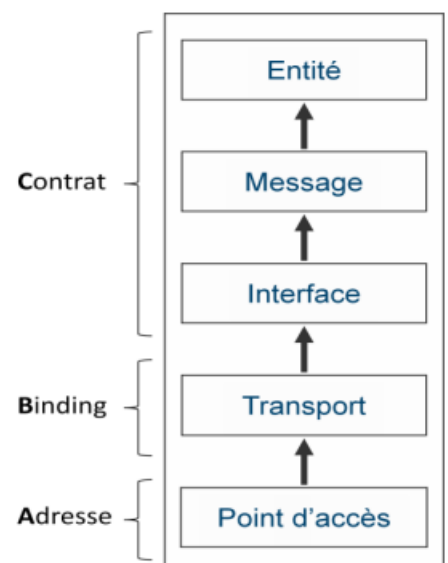
Puis, il ne reste plus qu'à créer une application cliente (quelle que soit sa forme, Windows, Web, ou tout autre composant) qui pourra consommer le service WCF.

WCF repose sur trois éléments :

- Une Adresse : adresse à laquelle le client doit se connecter pour utiliser le service.
- Un Binding : protocole à utiliser par le client pour communiquer avec le service.
- Un Contrat : infos échangées entre le serveur et le client afin que ce dernier sache comment utiliser le service.

La mise en œuvre de WCF repose principalement sur 2 espaces de nom du Framework .NET :

- System.ServiceModel
- System.Runtime.Serialization



Applications Réparties TD 3

Web Services et Windows Communication Foundation

Comme nous l'avons vu précédemment, le contrat de 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
- DataMember

La métadonnée « ServiceContract » 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.

La métadonnée « OperationContract » 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.

L'attribut DataMember 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 devoir nous retourner.

2.2 Création d'un Web Service avec WCF

En pratique, Visual Web Developer Express 2010 permet de développer un projet « Application du service WCF ». Les contrats sont alors définis dans le fichier « I<nom de service> ».cs et l'implémentation associée dans le fichier <nom de service>.svc.cs .

Question 4 : Créez un simple web service contenant une méthode Hello() retournant une chaîne « Hello ».

Question 5 : En vous positionnant sur le nom du projet dans l'explorateur de solution et en lançant le debugger, que voyez-vous ? que contient le fichier <nom du service>.svc ? Quel est l'url du fichier wsdl pour ce service ?

Question 6 : En vous positionnant sur le fichier <nom du service>.svc dans l'explorateur de solution et en lançant le debugger, que voyez-vous ?

2.3 Création d'un Client de test

Question 7 : A l'aide d'une application console développée sous Visual Studio C# 2010 Express, testez le service que vous avez développé précédemment.

2.4 Application

Question 8 : Développez un service web fournissant les fonctionnalités classiques d'une calculatrice, puis créez un client graphique utilisant ces fonctionnalités à distance.

Applications Réparties TD 3

Web Services et Windows Communication Foundation

3 Références

- [1] Demystifying Windows Communication Foundation, Keith Elder, Microsoft MVP, INETA Speaker.
- [2] Création et consommation de services WCF, Version 1.1, Association .Net France, Jean DALAT, James RAVAILLE.

