

WCF SOAP: Exercice

Objectifs

1. Découvrir WCF
2. Créer un Web Service et le consommer (client)
3. Manipuler les contrats (ServiceContract, DataContract)

Remarque importante

Si vous utilisez les machines de l'institut, créer vos projets sur un disque non réseau (pas U:\, Z:\). Vous pouvez utiliser MesDocuments ou encore votre Bureau. N'oubliez cependant pas de copier votre solution en fin de séance sur votre U:\ afin de pouvoir la récupérer. Si vous ne faites pas ceci, vous aurez une erreur « default proxy » → le proxy de l'école et WCF ne se comprennent pas bien.

Exercice 1 : Crise bancaire (Découverte WCF)

1. Créez un projet WCF Application Service. Renommez l'interface du service en « Ibanque » et l'implémentation en « BanqueImpl ».
2. L'interface du service définit les méthodes suivantes :

// renvoie la valeur augmentée de la TVA suivant le pays

// Ex : calculTVA(100,"BE") renvoie 121

// FR -> 19.06 %

// BE -> 21 %

// DE -> 19 %

double calculTVA(double valeur, string pays);

// renvoie la valeur multipliée par un nombre aléatoire compris entre 0 et 1.

// nombre aléatoire -> aller voir la classe Random

double criseBancaire(double valeur);

3. Une fois votre service implémenté, exécutez-le. Remarquez que votre web service est déployé dans IIS Express (barre des tâches de Windows).

Un navigateur s'ouvrira, cliquez sur le lien BanqueImpl.svc dans le navigateur. Si votre Web Service est correct, vous aurez l'image suivante qui indique que vous pouvez utiliser votre Web Service.

Service BanqueImpl

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 l'outil svcutil.exe à part

```
svcutil.exe http://localhost:1740/BanqueImpl.svc?wsdl
```

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

```
http://localhost:1740/BanqueImpl.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 application cli

C#

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

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

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

4. Créez ensuite une application cliente (ConsoleApplication) qui consommera le service c'est-à-dire qui appellera les 2 méthodes du service. Pour ce faire, créez un projet de type « Application Console » dans la même solution. Définissez ce projet comme projet de démarrage. Ajoutez une référence de service via l'explorateur de solutions, utiliser la fonction « découvrir » afin de trouver la référence vers le service.

Si la référence vers le service ne veut pas s'ajouter, vérifiez ceci :

Allez voir le fichier d'implémentation du service, faites un clic droit, affichez le balisage et vérifiez l'hébergement du service en particulier la balise « Service ».

Vous verrez que la référence s'appellera par défaut « ServiceReference1 ».

Dans votre programme principal « client », instanciez le proxy vers le service via un new ServiceReference1.BanqueClient.

Vérifiez ensuite l'appel aux méthodes du service.

Rmq : lorsque vous modifiez le contrat de service, pensez à régénérer votre solution et à mettre à jour votre référence de service dans le client.

Exercice 2 : Student (Prise en main + DataContract + InstanceObject)

Utilisez tout ce que vous avez mis en œuvre à l'exercice 1 afin de créer le service « Student » .

Le service mettra à disposition une liste d'étudiants ainsi que les opérations d'ajout, de modification, suppression, nombre d'étudiants et existence d'un étudiant dans cette liste.

Voici les méthodes de l'interface :

```
List<Student> lister();  
  
bool addStud(string name, string firstname, DateTime datenaiss);  
  
bool deleteStud(string name, string firstname, DateTime datenaiss);  
  
int nbStud();
```

La liste contiendra des **objets** de type « Student » contenant un nom, prénom et une date de naissance. Comme vous le voyez l'opération lister renverra ces objets « Student » , pensez donc à la sérialisation (DataContract).

Pensez également au fait que le comportement d'hébergement par défaut d'un web Service est d'être recyclé à chaque appel. Quelle conséquence ?