

# Applications Réparties TD 3

## Clients de Web Services (sous .NET)

---

Ce TD vous enseignera les bases de l'utilisation des Web Services en .NET avec Visual Studio .NET.

La configuration requise utilise le framework .Net 4.0

### 1 Introduction au développement sous Visual Studio

Pour les étudiants qui n'auraient pas développé en C#/.Net sous Visual Studio, l'enseignant présentera les bases de l'utilisation de l'IDE pour créer des applications « console » et graphiques.

### 2 Introduction aux Web Services

Vos applications .NET, quelles soient Windows Forms ou Web Forms, peuvent avoir recours, dans le but de vous simplifier la vie, à des **Web Services**.

Pour faire simple, vous pouvez imaginer ces Web Services comme des fonctions, stockées sur un serveur Web, et mise à la disposition de tout le monde. Vous pouvez donc, à partir de n'importe quel ordinateur disposant d'une connexion Internet, avoir accès à ces méthodes, mais vous ne pouvez pas les modifier ou bien même les visualiser : vous passer un ou plusieurs paramètres au Web Service, celui-ci effectue le traitement et vous renvoi ce qu'il faut.

En pratique, cela peut s'avérer un peu complexe d'où ce TD.

### 3 Création d'un Client à un Web Service existant

#### 3.1 Ajout de la référence Web

Avant de pouvoir travailler avec un Web Service, il faut ajouter sa référence Web dans votre projet. Pour cela, vous devez connaître l'adresse Web du Web Service.

Nous utiliserons le site <http://www.webservicex.net> qui contient un bon nombre de Web Services et qui vous offre la possibilité de pouvoir les tester en ligne.

Une fois l'adresse du Web Service trouvée, vous pourrez l'ajouter à votre projet comme une « Web References ».

A partir du fichier de description WSDL du service sélectionné, des classes proxy seront créées pour vous donner accès depuis votre application C# .Net à ce service.

#### 3.2 Développement d'une application cliente du Service Web

En premier lieu il vous faut parcourir l'espace de solution à la recherche du namespace de la référence Web importée.

Vous pourrez alors utiliser la Classe Proxy SOAP correspondante pour instancier un proxy vers le service Web dans votre programme et appeler de manière transparente les méthodes du service Web.

Exercice : Développez une application graphique de conversion de Devises à partir du service Web « Currency Convertor » de <http://www.webservicex.net>.

Attention à la notion de Binding. Comme expliqué en cours les Service Web peuvent s'appuyer sur tout type de protocole réseau couche Session ou Transport au niveau de la pile OSI/ISO. En pratique il peut s'agir de http bien sûr mais aussi SMTP, TCP/IP ... C'est ce choix qui vous permet d'introduire une politique de sécurité dans les accès aux services Web que vous concevrez.

## Applications Réparties TD 3

### Clients de Web Services (sous .NET)

---

En cas d'import d'un service vous trouverez ces informations dans le fichier App.Config.

Exercice : Trouvez alors 2 solutions pour choisir le binding qui vous intéresse.

### 3.3 Clients Web Services avancés

Les étudiants les plus avancés pourront développer une application qui consommera plus d'un Web Service, en particulier ceux de <http://www.websvcx.net> ou autres Web services de Google <http://msdn.microsoft.com/fr-fr/coding4fun/cc148273.aspx>.