

# Création d'un service Web

Création d'un service Web XML qui convertit les températures mesurées en degrés Fahrenheit en degrés Celsius.

## Création du projet de service Web XML

### Pour créer un projet de service Web ASP.NET

1. Dans le menu **Fichier**, cliquez sur **Nouveau site Web**.
2. Dans la boîte de dialogue **Nouveau site Web**, sélectionnez la version du Framework : .Net Framework 3.5
3. sélectionnez l'icône **Service Web ASP.NET**.
4. Sélectionnez le langage C#
5. Entrez l'adresse du serveur Web sur lequel vous allez développer le service Web XML et spécifiez **TConversion** comme nom de répertoire.
6. Cliquez sur **OK** pour créer le projet.

## Implémentation du service Web XML

L'étape suivante consiste à écrire le code pour implémenter la fonctionnalité du service Web XML auquel accéderont les clients. Pour les services Web XML créés dans Visual Studio, ce code est placé dans un fichier code-behind masqué associé au fichier .asmx du service Web XML créé par Visual Studio.

### Pour ajouter une méthode de service Web XML

1. Dans le fichier de code Service.cs, recherchez le code de déclaration de la classe `Service1`. Remplacez le code d'attribut `[WebService(Namespace = "http://tempuri.org/")]` par le code suivant la déclaration de la classe :

```
[System.Web.Services.WebService(
    Namespace="http://erp.ateliers/XmlWebServices/",
    Description="Un service de conversion de température.")]
```

2. Si vous attachez l'attribut `WebService` à une classe **public**, vous pouvez inclure des informations supplémentaires sur le service Web XML, notamment un espace de noms et une description de ce service. La propriété de description de cet attribut est incluse dans la page d'aide du service.
3. Effacez la méthode `HelloWorld`
4. Dans la classe `Service`, ajoutez le code suivant pour déclarer la fonction `ConvertirTemperature` :

```
[WebMethod(Description="Cette méthode convertit une
température °F en °C")]
public double convertirTemperature(double dFahrenheit)
{
    return ((dFahrenheit - 32) * 5) / 9;
}
```

Si vous attachez l'attribut `WebMethod` à une méthode **Public**, cette dernière est exposée comme une partie du service Web XML. La propriété de description de cet attribut est incluse dans la page d'aide du service et dans la page d'aide de la méthode du service.

5. Cliquez avec le bouton droit sur `Service.asmx` dans l'**Explorateur de solutions**, puis cliquez sur **Définir comme page de démarrage** dans le menu contextuel.
6. Enregistrez la solution.

## Débugage du service Web XML

Visual Studio offre plusieurs méthodes pour générer et exécuter un service Web XML à partir de l'environnement de développement intégré (IDE, Integrated Development Environment), notamment :

- Exécuter (avec débogage)
- Exécuter sans débogage
- Afficher dans le navigateur

En tant que projet Visual Studio, ce service Web XML possède des configurations distinctes pour les versions de diffusion et de débogage. Comme vous avez créé ce projet en utilisant le modèle de projet de service Web ASP.NET, Visual Studio crée automatiquement ces configurations et définit les options par défaut et les autres paramètres appropriés.

Dans cette procédure pas à pas, vous allez placer un point d'arrêt dans le service Web XML et utiliser la méthode **Exécuter (avec débogage)**.

### Pour utiliser un point d'arrêt et démarrer le service Web XML avec débogage

1. Dans le menu **Déboguer**, choisissez **Nouveau point d'arrêt**, puis **Interrompre à la fonction**. Sous l'onglet **Fonction**, tapez **convertirTemperature** dans la zone **Fonction**, puis cliquez sur **OK** pour ajouter un point d'arrêt à la déclaration de méthode `convertirTemperature`.
2. Dans le menu **Déboguer**, cliquez sur **Démarrer**, puis sur **OK** dans la fenêtre **Débogage non activé** pour démarrer le débogage.

Cette commande demande à Visual Studio d'exécuter le service Web XML dans le débogueur. Visual Studio génère le projet et le déploie sur le serveur de développement spécifié. Au terme de l'exécution, le navigateur par défaut affiche le fichier `.asmx` du serveur de déploiement.

Lorsque vous ouvrez un fichier `.asmx` dans un navigateur, le service Web XML retourne une page d'assistance de service qui fournit des informations à son sujet. Le lien **Description du service** permet d'accéder à un document XML qui contient la description de service formelle du service Web XML. Pour plus d'informations, consultez Description de service Web XML.

3. Dans la page d'assistance du service, cliquez sur le lien **convertirTemperature**.
4. Dans la zone **dFahrenheit**, tapez le nombre **212**, puis cliquez sur le bouton **Invoke**. Lorsque le traitement atteint la fonction `convertirTemperature`, il s'arrête. Le débogueur Visual Studio met en surbrillance la ligne contenant le point d'arrêt et, lors de l'arrêt, vous permet d'effectuer plusieurs tâches.
5. Dans le menu **Déboguer**, cliquez sur **Continuer** pour poursuivre le traitement.
6. Le service Web XML répond en retournant la valeur convertie en un document XML qui ressemble à ce qui suit :

```
<?xml version="1.0" encoding="utf-8" ?>
<double
xmlns="http://erp.Ateliers/XmlWebServices/">100</double>
```

7. Pour mettre fin à l'exécution du service Web XML et retourner dans l'éditeur de code, dans le menu **Déboguer**, cliquez sur **Arrêter le débogage**.
8. Dans le menu **Déboguer**, cliquez sur **Supprimer tous les points d'arrêt**.

# Accès à un service Web

## Création d'un projet de client de service Web

### Pour créer une application Web ASP.NET

1. Dans le menu **Fichier**, cliquez sur **Nouveau site Web**.
2. Dans la boîte de dialogue **Nouveau site Web**, sélectionnez la version du framework : **.Net Framework 3.5**
3. sélectionnez l'icône **Site Web ASP.NET**, et entrez l'adresse du serveur Web sur lequel vous allez développer l'application Web et spécifiez **TempConvertClient1** comme nom de répertoire .
4. Cliquez sur **OK** pour créer le projet.
5. Dans l'**Explorateur de solutions**, cliquez avec le bouton droit sur **Default.aspx** et choisissez **Concepteur de vues** pour ouvrir le concepteur.
6. Sous l'onglet **Web Forms** de la **boîte à outils**, faites glisser une **Zone de texte**, une **Étiquette** et un **Bouton** jusqu'à l'aire de conception de **Default.aspx** et arrangez ces éléments à votre convenance.
7. Cliquez avec le bouton droit sur le bouton que vous avez ajouté, **Button1**, et cliquez sur **Propriétés** dans le menu contextuel. Dans la fenêtre **Propriétés**, attribuez à la propriété **Text** la valeur **Convertir**.
8. Cliquez avec le bouton droit sur l'étiquette que vous avez ajoutée, **Label1**, et cliquez sur **Propriétés** dans le menu contextuel. Dans la fenêtre **Propriétés**, effacez la propriété **Text** pour créer une étiquette vide.

### Ajout d'une référence Web

La découverte de service Web est le processus utilisé par un client pour rechercher un service Web et obtenir sa description de service. Dans Visual Studio, le processus de découverte de service Web implique l'interrogation d'un site Web en fonction d'un algorithme prédéterminé. L'objectif de ce processus est de rechercher la description de service, qui est un document XML utilisant le langage de description de services Web (WSDL, Web Services Description Language). La description de service décrit les services disponibles ainsi que les méthodes d'interaction avec ces services. Sans description de service, il est impossible d'interagir par programme avec un service Web.

Votre application doit avoir un moyen de communiquer avec le service Web et de le rechercher au moment de l'exécution. L'ajout d'une référence Web à votre projet pour le service Web permet d'effectuer ces opérations en générant une classe proxy qui se connecte au service Web et fournit une représentation locale de ce dernier.

### Pour ajouter une référence Web

1. Dans le menu **Site Web**, cliquez sur **Ajouter une référence Web**.
2. Dans la zone **URL** de la boîte de dialogue **Ajouter une référence Web**, tapez l'URL permettant d'obtenir la description du service Web auquel vous voulez accéder, par exemple **http://localhost:numPort/TConversion/Service1.asmx**. Cliquez ensuite sur le bouton **Aller à** pour récupérer des informations sur le service Web.  
- ou -  
Si le service Web existe sur l'ordinateur local, cliquez sur le lien **Services Web sur l'ordinateur local** dans le volet de navigation. Cliquez ensuite sur le lien du service Web TempConvert1 dans la liste fournie pour récupérer des informations sur ce service Web.
3. Dans la zone **Espace de noms**, renommez l'espace de noms en **ConvertSvc**, qui est l'espace de noms que vous utiliserez pour accéder à cette référence Web.
4. Cliquez sur **Ajouter une référence** pour ajouter une référence Web au service Web cible.

## 5. Accès au service Web

Après avoir ajouté une référence pour le service Web à votre projet, vous devez créer une instance de la classe proxy du service Web. Vous pouvez alors accéder aux méthodes du service Web de la même manière qu'aux méthodes de tous les autres objets, en appelant les méthodes de la classe proxy. Lorsque votre application appelle ces méthodes, le code de classe proxy généré par Visual Studio gère les communications entre votre application et le service Web.

Dans un premier temps, vous allez créer une instance de la classe proxy du service Web. Vous allez ensuite prendre une valeur, fournie dans **TextBox1**, et effectuer un appel à la méthode **convertirTemperature** du service Web en utilisant la classe proxy. Vous afficherez ensuite la valeur retournée par le service Web dans **Label1**.

### Pour accéder au service Web

1. Double-cliquez sur le bouton **Convertir** dans **WebForm1.aspx** pour créer une méthode de gestion d'événements pour ce bouton et pour afficher le fichier code-behind.
2. Entrez le code suivant :

```
protected void Button1_Click(object sender, EventArgs e)
{
    try
    {
        ServiceSoapClient ws = new ServiceSoapClient();
        double dFahrenheit = Convert.ToDouble(TextBox1.Text);
        double dCelsius = ws.convertirTemperature(dFahrenheit);
        Label1.Text = dCelsius.ToString();
    }
    catch
    {
        Label1.Text = "La conversion a échoué.";
    }
}
```

3. Sélectionnez **Default.aspx** dans l'Explorateur de solutions.
4. Dans le menu **Site Web**, cliquez sur **Définir comme page de démarrage**.
5. Enregistrez la solution.

### Débogage du client de service Web

#### Pour utiliser un point d'arrêt et démarrer l'application Web avec débogage

1. Dans le menu **Débuguer**, cliquez sur **Nouveau point d'arrêt**, puis sélectionnez **Interrompre à la fonction**. Tapez **Button1\_Click** dans la zone **Fonction**, puis cliquez sur **OK** pour placer un point d'arrêt sur le gestionnaire d'événements **Button1\_Click**. Pour plus d'informations, consultez Vue d'ensemble des points d'arrêt.
2. Dans le menu **Débuguer**, cliquez sur **Démarrer**, puis sur **OK** dans la fenêtre **Débogage non activé** pour démarrer le débogage. Cette commande ordonne à Visual Studio d'exécuter l'application Web dans le débogueur. Visual Studio génère le projet et le déploie sur le serveur de développement spécifié. Au terme de l'exécution, le navigateur par défaut est lancé et navigue jusqu'au fichier .aspx sur le serveur de déploiement.
3. Dès que la page s'affiche dans le navigateur, tapez le nombre **212** dans la zone de texte, puis cliquez sur le bouton **Convertir**. Lorsque le traitement atteint le gestionnaire d'événements **Button1\_Click**, il s'arrête. Le débogueur Visual Studio met en surbrillance la ligne contenant le point d'arrêt et, lors de l'arrêt, vous permet d'effectuer plusieurs tâches. Pour plus d'informations, consultez Présentation du débogueur et Affichage des données dans le débogueur.
4. Dans le menu **Débuguer**, cliquez sur **Continuer** pour poursuivre le traitement.
5. Le service Web répond en retournant la valeur convertie, et l'application attribut au texte de **Label1** la valeur **100**. Pour mettre fin à l'exécution de l'application Web et retourner dans l'éditeur de code, dans le menu **Débuguer**, cliquez sur **Arrêter le débogage**.
6. Dans le menu **Débuguer**, cliquez sur **Effacer tous les points d'arrêt**.