Tutorial ASP.NET: Trabajando con XML y Web Services

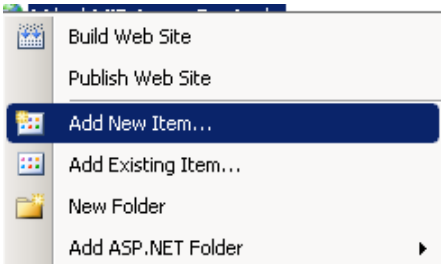
Los servicios web XML permiten intercambiar datos en forma de mensajes XML entre sistemas heterogéneos. Los servicios web se crean utilizando el entorno de aplicación ASP.NET, lo que permite a estos tener acceso a numerosas características de .NET Framework.

Fernando Giardina enero 20 2011

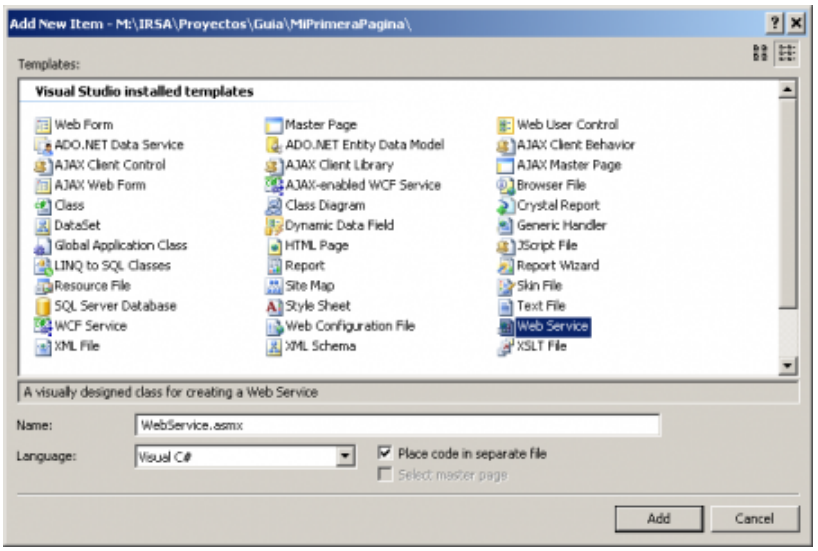
¿Cómo crear un web services usando el IDE de ASP.NET?

1.- Lo primero que debemos hacer es agregar un nuevo ítem de tipo "WebService" a la solución.

Hacemos botón derecho sobre el proyecto, "**Add new ítem**".



De la lista de ítems seleccionamos Web Service que por default nos propone un nombre. Lo vamos a cambiar para no confundir los conceptos. Nuestro Web Services se llamará `wsMDW.asmx`



Ahora vamos a tener en nuestro proyecto dos archivos nuevos.

- `wsMDW.asmx`
- `wsMDW.cs`: este archivo va a contener los métodos y objetos que necesitamos publicar.

wsMDW.cs

Esta es la estructura base de la clase `wsMDW`. Esta clase hereda de `System.Web.Services.WebService` y de esta forma la identificamos como Web Services.

```
01 using System;
02 using System.Collections.Generic;
03 using System.Web;
04 using System.Web.Services;
05
06 [WebService(Namespace = "http://tempuri.org/")]
07 [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfil
08 public class wsMDW : System.Web.Services.WebService {
09 public wsMDW ()
10 { } AQUÍ VAN LOS METODOS WEB QUE VAMOS A PUBLICAR
11 }
```

Nuestro primer método web:

Vamos a agrega un método llamado “FechaYHora” que nos va a devolver la fecha y hora del servidor.

```
1 [WebMethod] public string FechaYHora() {
2 return DateTime.Now.ToString();
3 }
```

Vemos que tiene la misma apariencia de un método común y corriente.

Las dos grandes diferencias son que este método está dentro de una clase que Hereda de `System.Web.Services.WebService` y tiene el Decorate [WebMethod] que lo identifica como Método web.

Perfecto, ahora nos preguntamos qué tiene que ver XML en todo eso. Sencillamente lo que responde un Web Services es en formato XML. Vamos a ejecutarlo presionando F5 desde el IDE de ASP.NET

Ahora clickeamos sobre el link “FechaYHora” lo invocamos y veremos en pantalla la respuesta que nos da el web services.

El resultado es este XML con el tag string cuyo valor es la fecha y hora del momento en que se ejecuto el método web.

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <string xmlns="http://tempuri.org/">29/10/2010 11:18:26
```

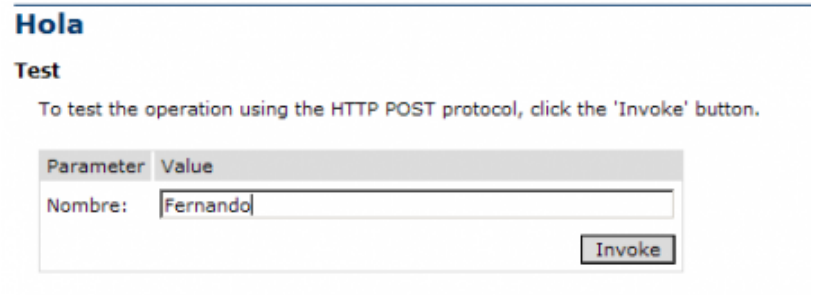
De la misma forma se crean y prueba otros métodos web que reciban o no parámetros. Ahora veamos un ejemplo simple de un método web que recibe parámetros.

Podemos probar con el clásico HOLA.

```
[WebMethod]
public string Hola(string Nombre)
{
return "Hola " + Nombre; }
```

- FechaYHora
- Hola

Presionamos sobre el link Hola y nos va a pedir el parámetro NOMBRE que definimos en nuestro método web.

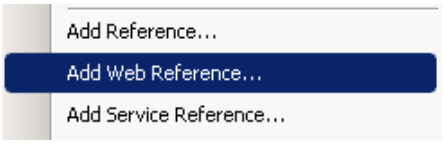


Agregamos un nombre, invocamos y veamos la respuesta.

```
<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="http://tempuri.org/">Hola Fernando</strin
```

De esta forma testeamos web services. Ahora nos interesa ver como los podemos utilizar dentro de nuestra aplicación. Lo primero que vamos a hacer es generar una referencia web de la siguiente forma.

1.- Hacemos botón derecho sobre el proyecto y agregamos una referencia web.

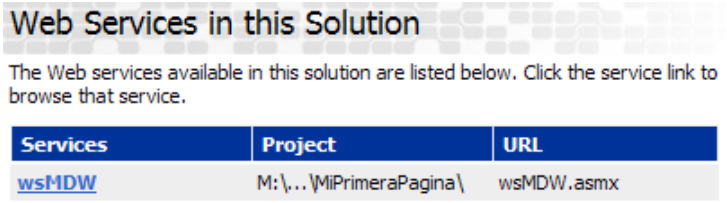


En esta ventana nos da 3 opciones.

Browse to:

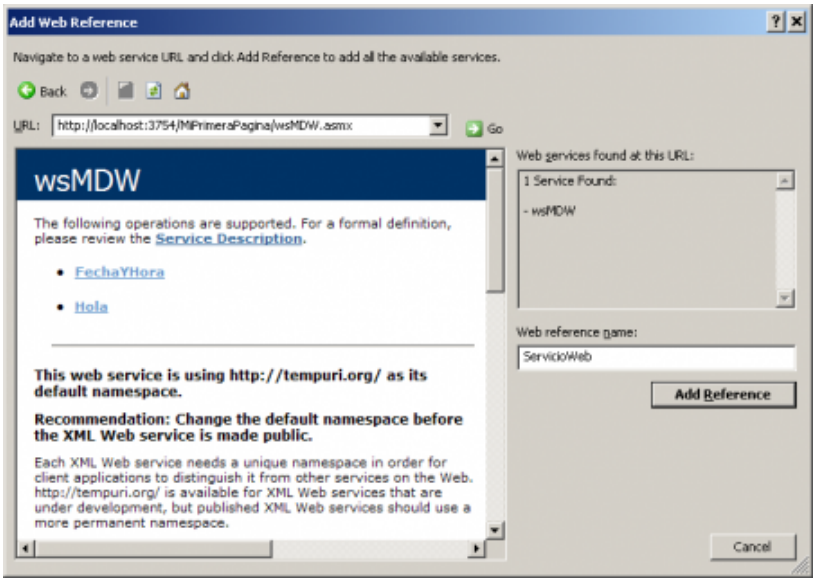
- Web services in this solution
- Web services on the local machine
- Browse UDDI Servers on the local network
  Query your local network for UDDI servers.

2.- Como el web services está dentro del proyecto vamos a seleccionar la primera opción.



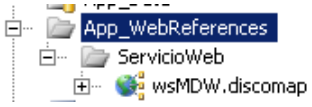
Ahí se van a listar todos los web services que tengamos en nuestro proyecto.

3.- Seleccionamos el Servicio wsMDW y nos va a mostrar un preview y nos solicita un nombre.
Pondremos ServicioWeb



4.- Por último, agregamos el servicio presionando el botón “Add reference” y se agrega la referencia a nuestro proyecto.

Como vemos se creo una nueva carpeta “App_WebReferences”, ServicioWeb y el archivo wsMDW.discomap



Ahora que ya tenemos referenciado nuestro web services en el proyecto podemos consumirlo.

Supongamos que queremos invocar el método FechaYHora y mostrarlo en pantalla. Para eso, vamos a crear un nuevo webform llamado ConsumoWS.aspx, y agregamos un Label donde vamos a mostrar la respuesta del método.

ConsumoWS.aspx

```
1 <%@ Page Language="C#" AutoEventWireup="true" CodeFile=
2 <html> <head runat="server">
3 <title></title> </head>
4 <body> <form id="form1" runat="server"> <div>
5 <asp:Label ID="Label1" runat="server" Text="Label"></asp
6 </form> </body>
7 </html>
```

Ahora la llamada al método web la hacemos desde el CodeBehind.

ConsumoWS.aspx.cs

Agregamos estas líneas al método `Page_Load`

```
1 protected void Page_Load(object sender, EventArgs e) { 
2 ServicioWeb.wsMDW();
3 Label1.Text = miws.FechaYHora();
4 }
```

Así de simple se crea y consume un web services realizado en ASP.NET

Capítulos del Tutorial

- Capítulo 1: Tutorial de desarrollo web con ASP.NET
- Capítulo 2: Primera aplicación web en ASP.NET
- Capítulo 3: Ejecutar código JavaScript en ASPNET
- Capítulo 4: Archivos Web.Config y Global.asax
- Capítulo 5: Utilización de Código detrás del modelo o código en línea

- Capítulo 6: Controles de servidor y eventos
- Capítulo 7: Controles de usuarios reutilizables
- Capítulo 8: Crear una página de login, autenticación y seguridad.
- Capítulo 9: Utilización de estilos en ASP.NET (CSS)
- Capítulo 10: Trabajando con XML y Web Services

1 Compartir 2 Twittear 0 g+1 Compartir

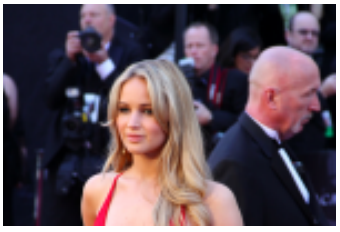
Fernando Giardina

En 1997 comienzo a desarrollar en lenguaje C y Oracle programando IVRs, automatización de procesos y sistemas de CRM. Continuo con JAVA, Perl y PHP hasta que en el año 2002 me inicio en la tecnología .NET desarrollando aplicaciones web, desktop y mobile.

http://www.metalstudio.com.ar

Otros artículos que podrían interesarte

Lo peor y lo ... peor de trabajar para una startup



Fotos de famosos desnudos: un modelo de negocio emergente en Internet

Drones que podrías estar volando ahora mismo

Google Wing hace ver los drones de Amazon como juguetes

Añade un comentario...
Publicar también en Facebook
Publicar como **Javier J Solis Flores** ▾ Comentar
Plug-in social de Facebook

Antares

Una pregunta Fernando, sin ningún tipo de malicia. En una ocación recuerdo haber consumido un webservice en ASP.NET usando Visual Basic. En esa oportunidad necesitabamos agregar un namespace al archivo de código subyacente utilizando el palabra reservada “Imports” para tener acceso a la clase del Web Service. En C# no es necesario hacer eso para acceder a la clase “wsMDW”. Atentamente Antares.

Fernando Serapio

Cuando usas el VS o VWD y agregas un webservice a la solucion automaticamente te crea las referencias para las ensamblados que necesites usar, en este caso, al agregar el webservice te agrega:

using System.Web.Services;

Michi

Claro, eso lo resuelve .net al agregar el web service.

Luis

Excelente articulo! He estado buscando en internet articulos sobre WebServices en .NET y no habia podido lograr encontrar nada que me ayudara a entender y practicarlo. Esto esta muy bien explicado.
Gracias