This step-by-step article shows you how to write a simple Web service, called MathService, that exposes methods for adding, subtracting, dividing, and multiplying two numbers.
Requirements

The following items describe the recommended hardware, software, network infrastructure, skills and knowledge, and service packs that you need:

- Microsoft Internet Information Server 4.0 or Internet Information Server 5.0
- Microsoft Visual Studio .NET

This article assumes that you are familiar with the following topics:
- How to use the Visual Studio .NET integrated development environment

Write a Simple .asmx Web Service

1. Open Visual Studio .NET.

2. On the File menu, click New and then click Project. Under Project types click Visual C# Projects, then click ASP.NET Web Service under Templates. Type MathService in the Location text box to change the default name (WebService1) to MathService.

3. Change the name of the default Web service that is created from Service1.asmx to MathService.asmx.

4. Click Click here to switch to code view in the designer environment to switch to code view.

5. Define methods that encapsulate the functionality of your service. Each method that will be exposed from the service must be flagged with a WebMethod attribute in front of it. Without this attribute, the method will not be exposed from the service.

   NOTE: Not every method needs to have the WebMethod attribute. It is useful to hide some implementation details called by public Web service methods or for the case in which the WebService class is also used in local applications. A local application can use any public class, but only WebMethod methods will be remotely accessible as Web services.

Add the following method to the MathServices class that you just created:

```csharp
[WebMethod]
public int Add(int a, int b)
```
{ return(a + b);
}

[WebMethod]
{
    return (A - B);
}

[WebMethod]
{
    return A * B;
}

[WebMethod]
{
    if(B == 0)
    
    return -1;
    return Convert.ToSingle(A / B);
}

6. Click **Build** on the **Build** menu to build the Web service.

7. Browse to the MathService.asmx Web service page to test the Web service. If you set the local computer to host the page, the URL is http://localhost/MathService/MathService.asmx.

   The ASP.NET runtime returns a Web Service Help Page that describes the Web service. This page also enables you to test different Web service methods.

### Consume a Web Service

1. Open Visual Studio .NET.

2. Under **Project types** click **Visual C# Projects**, then click **Console Application** under **Templates**.

   **Add a reference for the MathService Web service to the new console application.**

   This step creates a proxy class on the client computer. After the proxy class exists, you can create objects based on the class. Each method call that is made with the object then goes
out to the uniform resource identifier (URI) of the Web service (usually as a SOAP request).

a. On the **Project** menu, click **Add Web Reference**.

b. In the **Add Web Reference** dialog box, type the URL for the Web service in the **Address** text box and press ENTER. If you set the local computer to host the Web service, the URL is http://localhost/MathService/MathService.asmx.

c. Click **Add Reference**. Alternatively, you can type the URL to the discovery file (MathService.vsdisco) or click **Web References on Local Web Server** in the left pane to select the MathService service from the list.

d. Expand the **Web References** section of Solution Explorer and note the namespace that was used.

4. Create an instance of the proxy object that was created. Place the following code in the function called **Main**:

   ```csharp
   localhost.Service1 myMathService = new localhost.Service1();
   ```

5. Invoke a method on the proxy object that you created in the previous step, as follows:

   ```csharp
   Console.Write("2 + 4 = {0}", myMathService.Add(2,4));
   ```

6. Click **Build** on the **Build** menu to build the console application.

7. Click **Start** on the **Debug** menu to test the application.

8. Close and save the project.

back to the top

References
For more information, see the "Programming the Web with Web Services" topic in the Visual Studio .NET Help, or the "ASP.NET Web Services and ASP.NET Web Service Clients" topic in the Microsoft .NET Framework Developer’s Guide.

For more information, visit the following Microsoft Web sites:

XML Web Services Developer Center
http://msdn.microsoft.com/webservices

Extreme XML: XML Web Service-Enabled Office Documents (MSDN Voices column):

Extreme XML: UDDI: An XML Web Service (MSDN Voices column):

DHTML Dude: Accessing Web Services From DHTML (MSDN Voices column):

For more information, see the Web Services Description Language Tool (Wsdl.exe) (Microsoft .NET Framework Tools).

Properties

Article ID: 308359 - Last Review: 08/16/2012 09:21:00 - Revision: 8.0

Applies to
Microsoft ASP.NET 1.1

Microsoft ASP.NET 1.0

Microsoft Web Services Enhancements for Microsoft .NET 1.1

Microsoft Visual C# .NET 2002 Standard Edition

Keywords:
kbhowtomaster kbsample KB308359