

How-To Write a Web Service with Visual C# and ASP.NET

This article helps you write a web service, called *MathService*, that exposes methods for adding, subtracting, dividing, and multiplying two numbers.

Requirements

The following list describes the recommended hardware, software, skills and knowledge that you need:

- Microsoft Windows
- Internet Information Server
- Visual Studio .NET

This article assumes that you're familiar with the topic: How to use the Visual Studio .NET integrated development environment.

Write an .asmx web service

1. Open Visual Studio .NET.
2. On the **File** menu, select **New** and select **Project**. Under **Project types**, select **Visual C# Projects**. Then select **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. Select **Click** to switch to code view in the designer environment.
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's 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 created:

C#

```
[WebMethod]
public int Add(int a, int b)
{
    return(a + b);
}
[WebMethod]
public System.Single Subtract(System.Single A, System.Single B)
{
    return (A - B);
}
[WebMethod]
public System.Single Multiply(System.Single A, System.Single B)
{
    return A * B;
}
[WebMethod]
public System.Single Divide(System.Single A, System.Single B)
{
    if(B == 0) return -1;
    return Convert.ToSingle(A / B);
}
```

6. Select **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**, select **Visual C# Projects**, then select **Console Application under Templates**.
3. 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).

1. On the **Project** menu, select **Add Web Reference**.

2. 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`.
3. Select **Add Reference**. Alternatively, you can type the URL to the discovery file (*MathService.vsdisco*) or select **Web References** on Local Web Server in the left pane to select the MathService service from the list.
4. 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:

C#

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

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

```
C#Copy  
Console.WriteLine("2 + 4 = {0}", myMathService.Add(2,4));
```

6. Select **Build** on the **Build** menu to build the console application.
7. Select **Start** on the **Debug** menu to test the application.
8. Close and save the project.

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 .NET Framework Developer's Guide.

Courtesy: <https://docs.microsoft.com/en-us/troubleshoot/dotnet/csharp/write-web-service>

Modified: 2021.10.14.7.00.AM
Dököll Solutions, Inc.