

STUDENT ACTIVITY 2.3

MTA Course: Web Development Fundamentals

Lesson name: Web Development Fundamentals 2.3

Topic: Call a service from a Web page

File name: WebDevFund_SA_2.3

Lesson Objective:

2.3: Call a service from a Web page. *This objective may include but is not limited to:* creating a basic WCF Service or Web Service so that it can be consumed; App_WebReferences; <system.serviceModel> configuration.

Resources, software, and additional files needed for this lesson:

- A Windows®-based PC with installed Web development software. Examples include Visual Studio 2008®, or
 - Microsoft® Visual Basic 2008®, Express Edition (<http://www.microsoft.com/express/downloads/#2008-Visual-Basic>)
 - Microsoft Visual C# 2008®, Express Edition (<http://www.microsoft.com/express/downloads/#2008-Visual-CS>)
 - Microsoft Visual Web Developer 2008, Express Edition (<http://www.microsoft.com/express/downloads/#2008-Visual-Web-Developer>)
- Use these sites for additional information:
 - Web services
 - <http://www.asp.net/learn/videos/video-280.aspx>
 - <http://www.asp.net/learn/3.5-videos/video-231.aspx>

Directions to the student:**Part 1**

Write a simple .asmx Web service

1. Open Visual Studio 2008.
2. Create a new ASP.NET Web service project. Name the Web service MathService and point the location to an appropriate Web server that is running ASP.NET if necessary.

3. Change the name of the Solution file to MathService for consistency.
4. Change the name of the default Web service that is created from Service1.asmx to MathService.asmx.
5. At this point, the code will be displayed. You can change the name of the class from Public Class Service1 to Public Class MathService, but you must also change the name of the class referenced in the asmx file in its markup view.
6. 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 situation in which the WebService class is also used in local applications. A local application can use any public class, but only WebMethod methods can be remotely accessed as Web services.

7. Add the following methods to the MathServices class that you just created. These methods must go inside the class body and replaced any placeholder method that already exists, such as Hello World.(This example is using Visual Basic.)

```
<WebMethod()> Public Function Add(a As Integer, b As Integer) As Integer
    Return(a + b)
End Function

<WebMethod()> Public Function Subtract(A As System.Single, B As
System.Single) As System.Single
    Return A - B
End Function

<WebMethod()> Public Function Multiply(A As System.Single, B As
System.Single) As System.Single
    Return A * B
End Function

<WebMethod()> Public Function Divide(A As System.Single, B As
System.Single) As System.Single
    If B = 0
        Return -1
    End If
    Return Convert.ToSingle(A / B)
End Function
```

8. Select Build on the Build menu to finalize the Web service.

9. 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.

Part 2

Consume a Web service

1. Start Visual Studio 2008.
2. Create a new Console application project. Select either VB or C#.
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).
 - a. In the Solution Explorer, right click on References, choose 'Add Service Reference'. From the dialog box, choose 'Advanced', and then click the 'Add Web Reference' button at the bottom of the dialog box. In the Add Service 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*.
 - b. Click Add Reference.
 - c. 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 this code in the Main procedure of the Module1 module:


```
Dim myMathService As localhost.MathService = New
localhost.MathService()
```
5. In the main method of your console application code, invoke a method on the proxy object created in the previous step, such as:


```
Console.WriteLine("2 + 4 = {0}", myMathService.Add(2,4))
```
6. Save the project. Compile and run the application.

Note: Be prepared to discuss what you learned to the class.