

This topic is specific to a legacy technology. XML Web services and XML Web service clients should now be created using [Windows Communication Foundation](#).

After you publish a Web service created using ASP.NET, you can test its functionality by using a browser to call it via either HTTP-GET or HTTP-POST. Access its .asmx file in a browser and then click hyperlinks to the Web service methods, or access individual methods directly by appending a query string to the .asmx URL.

Note:

By default, a Web service created using ASP.NET is able to support multiple protocols, including SOAP over HTTP and implementations of HTTP-GET and HTTP-POST where non-SOAP XML is returned in the response.

To test a Web service in a browser using HTTP-GET

1. Deploy your Web service to a Web server. For more information, see [XML Web Service Publishing and Deployment](#).
2. Access a Web browser and enter the URL for your Web service in the address bar, using the following format:

```
http://servername/appath/webservicename.asmx
```

Path portion	Value
<i>servername</i>	The name of the server on which your Web service was deployed.
<i>Appath</i>	The name of your virtual directory and the rest of the Web application path.
<i>webservicename.asmx</i>	The name of the Web service .asmx file.

For example, suppose you have published a Web service named `StockServices`. When published, the base URL for this service is `http://<servername>/appath/StockServices.asmx`. You could test this service by entering this HTTP-GET request in the browser's address bar:

```
http://<servername>/appath/StockServices.asmx
```

The server, in response to this request, displays the Web service's HTML description page.

3. The Web service's HTML description page shows you all the Web service methods supported by a particular Web service. Link to the desired Web service method and enter the necessary parameters to test the method and see the XML response.

To directly test a Web service method in a browser using HTTP-GET

1. Deploy your Web service to a Web server. For more information, see [XML Web Service Publishing and Deployment](#).

2. Access a Web browser and enter the URL for your Web service method in the address bar, using this format:

```
http://servername/vdir/webservicename.asmx/Methodname?parameter=value
```

Parameter	Value
<i>servername</i>	The name of the server on which your Web service is deployed.
<i>Apppath</i>	The name of your virtual directory and the rest of the Web application path.
<i>webservicename.asmx</i>	The name of the Web service .asmx file.
<i>Methodname</i>	The name of a public method that is exposed by your Web service. If left blank, the Web service's description page is shown, listing each public method available in the .asmx file. (Optional)
<i>parameter</i>	The appropriate parameter name and value for any parameters required by your method. If left blank, the Web service's description page is shown, listing each public method available in the .asmx file. (Optional)

Note:

The Web service method name in this syntax is case sensitive, but the server, project, and Web service names are not.

For example, suppose the `StockServices` Web service from the preceding procedure contains a Web service method called `GetQuote`; the Web service method accepts a stock symbol as a parameter, returning the price as a double-precision floating-point number. Enter the following HTTP-GET request in the browser's address bar to test this method:

```
http://<servername>/appath/StockServices.asmx/GetStockQuote?tickerName=MSFT
```

3. The server sends a response containing an XML document, which is displayed in the browser. For the `GetQuote` example, the XML has the current price of the stock you request. The result might look like the following:

```
<?xml version="1.0" ?>  
  
<double>74.5</double>
```

To test a Web service in a browser using HTTP-POST

1. Deploy your Web service to a Web server. For more information, see [XML Web Service Publishing and Deployment](#). This procedure uses as an example the following Web service, deployed as a file `math.asmx` that is accessible from the virtual root of a site `http://www.contoso.com`:

VB

This language is not supported or no code example is available.

VB

```

<%@ WebService Language="VB" Class="Math" %>
Imports System.Web.Services

Public Class Math
    Inherits WebService

    <WebMethod> _
    Public Function Add(num1 As Integer, num2 As Integer) As Integer
        Return num1 + num2
    End Function

    <WebMethod> _
    Public Function Subtract(num1 As Integer, num2 As Integer) As Integer
        Return num1 - num2
    End Function
End Class

```

2. Create an HTML page with a form that has its **method** attribute set to **POST**. Use the following format:

```

<form method=POST action='http://www.contoso.com/math.asmx/Subtract'>
    <input type="text" size="5" name='num1\' "></td> -
    <input type="text" size="5" name='num2\' "></td> =
    <input type=submit value="Subtract"> </td>
</form>

```

Parameter	Value
<i>method</i>	POST . If you want to test your Web service using HTTP-POST, use POST.
<i>action</i>	URL to the Web service method. In the previous example, math.asmx is the Web service and Subtract is the Web service method.
<i>type="text"</i>	For each parameter of the Web service method, create input tags with the type attribute set to "text" . This allows you to type a parameter value in the text input control.
<i>name='num1'</i>	The name of the Web service method parameter. Add as many text input controls on the form as there are parameters in the Web service method. For instance, if a Web service method has three parameters, three text input controls are needed that each have their name attribute set to the name of the parameter.
<i>type=submit</i>	Add a submit button so you can post the data back to the Web service method.

3. Access a Web browser and enter the URL for the HTML document you created in the previous step.

The HTML document created in the previous step is displayed.

4. Enter the appropriate values for the Web service method in the text boxes and click the **submit** button.

For example, if you entered **6** and then **3** into the two text boxes for the example's Subtract Web service method, the following result is returned:

```
<?xml version="1.0" ?>  
<int xmlns="http://tempuri.org/">3</int>
```

See Also

Tasks

[How to: Explore Existing XML Web Services Created Using ASP.NET](#)

[How to: Access XML Web Services from a Browser](#)

Concepts

[Building XML Web Service Clients](#)

[Web Services Discovery](#)

Other Resources

[Creating Clients for XML Web Services](#)

Copyright © 2007 by Microsoft Corporation. All rights reserved.
Last Published: 2010-05-07

Community Content