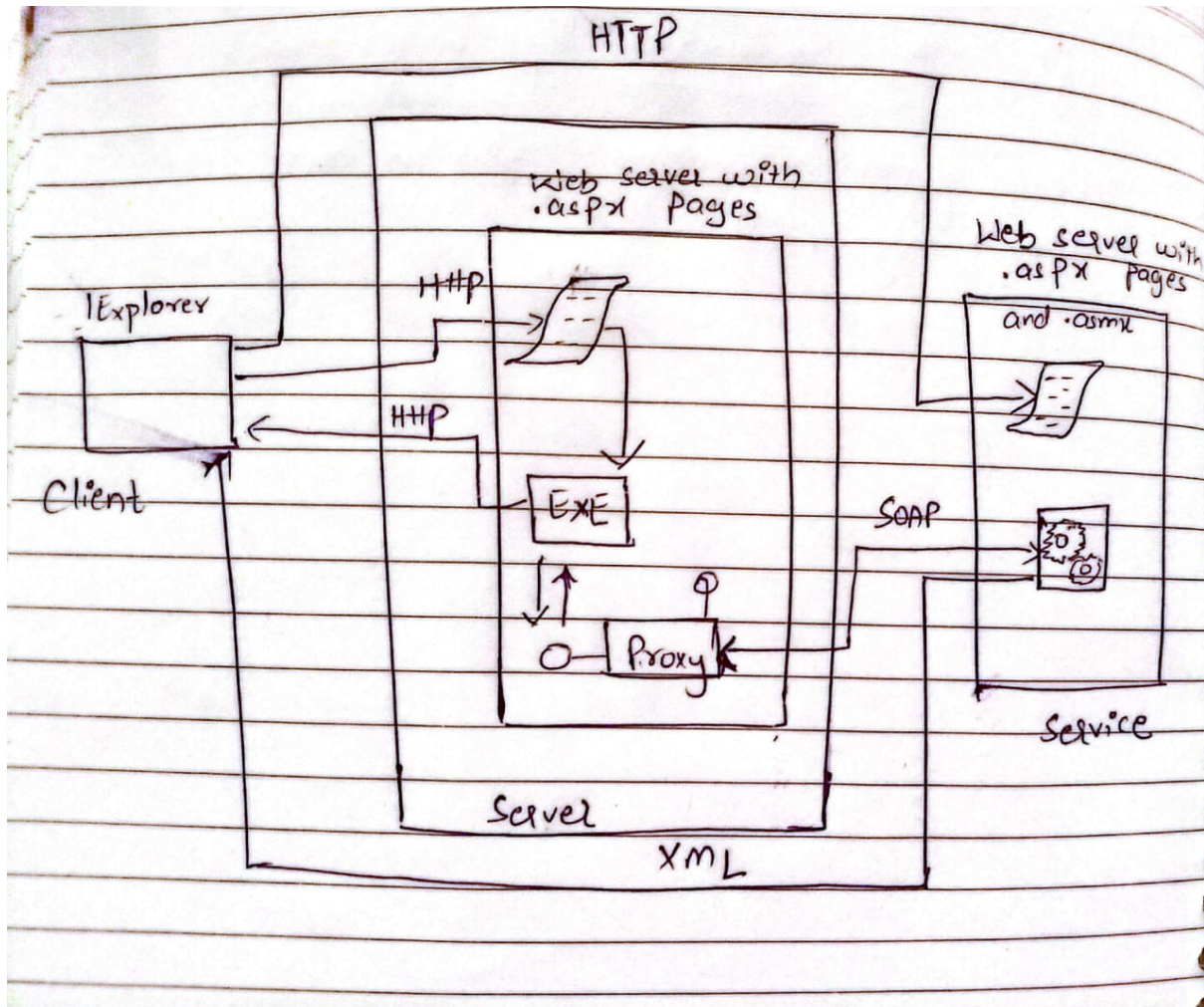


IAD Assignment Web Services

Web Services Execution Model:



MathLibServiceAlishbah.asmx:

```
<%@ WebService Language="VB" Class="MathLibService" %>
```

```
Imports System.Web.Services
```

```
<WebService(Namespace:="http://tempuri.org/")>
```

```
<WebServiceBinding(ConformsTo:=WsiProfiles.BasicProfile1_1)>
```

```
Public Class MathLibService
```

```
Inherits System.Web.Services.WebService
```

```
<WebMethod()>
```

```
Public Function Add(num1 As Double, num2 As Double) As Double
    Return num1+num2
End Function
```

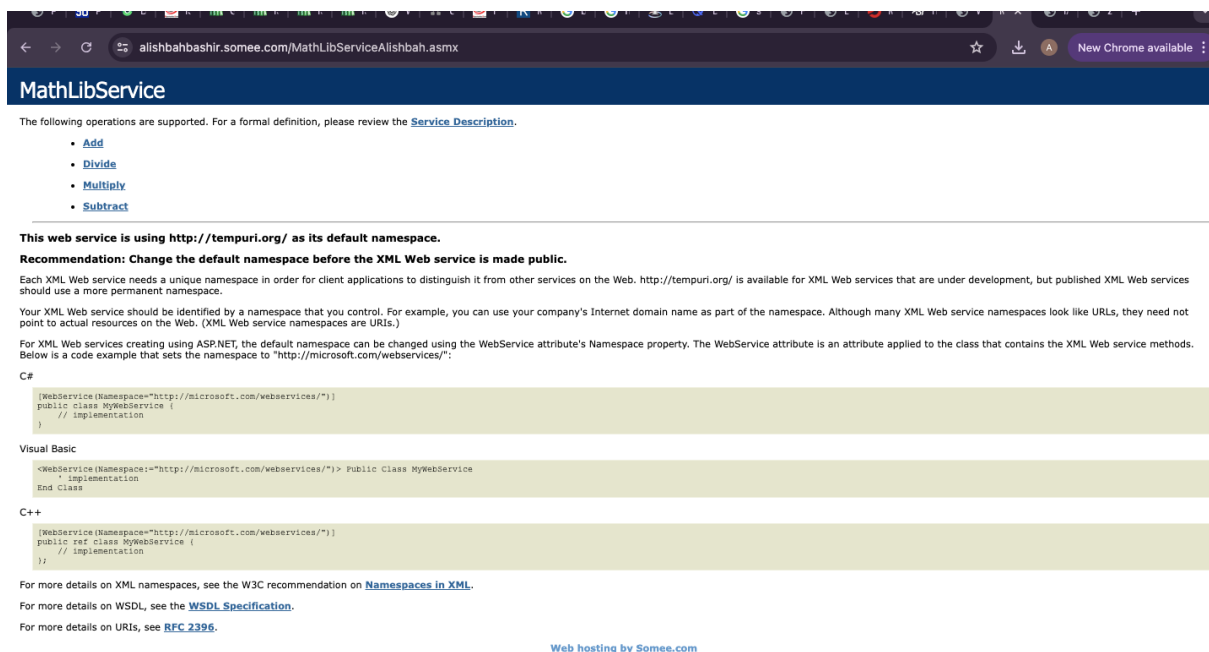
```
<WebMethod()>
Public Function Subtract(num1 As Double, num2 As Double) As Double
    Return num1-num2
End Function
```

```
<WebMethod()>
Public Function Multiply(num1 As Double, num2 As Double) As Double
    Return num1*num2
End Function
```

```
<WebMethod()>
Public Function Divide(num1 As Double, num2 As Double) As Double
    If num2=0 Then
        Return "Error: Division by zero"
    End If
    Return num1/num2
End Function
```

End Class

Testing Web Service:



The following operations are supported. For a formal definition, please review the [Service Description](#).

- [Add](#)
- [Divide](#)
- [Multiply](#)
- [Subtract](#)

This web service is using <http://tempuri.org/> as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. <http://tempuri.org/> is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services created using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "<http://microsoft.com/webservices/>".

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementation
}
```

Visual Basic

```
<WebService(Namespace="http://microsoft.com/webservices/")> Public Class MyWebService
    ' implementation
End Class
```

C++

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public ref class MyWebService {
    // implementation
};
```

For more details on XML namespaces, see the W3C recommendation on [Namespaces in XML](#).

For more details on WSDL, see the [WSDL Specification](#).

For more details on URIs, see [RFC 2396](#).

[Web hosting by Somee.com](#)

Add

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
a:	21
b:	71

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<string xmlns="http://tempuri.org/">92</string>
```

Subtract

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
num1:	2.7
num2:	3.4

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<string xmlns="http://tempuri.org/">-0.7</string>
```

ServiceClient.aspx:

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="ServiceClient.aspx.vb"
Inherits="ServiceClient" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
  <title>Math Web Service Client</title>
```

```
</head>
```

```
<body>
```

```
  <form id="form1" runat="server">
```

```
    <div>
```

```
      <h2>Math Web Service Client</h2>
```

```
      <asp:Label ID="lblNum1" runat="server" Text="Number 1: "></asp:Label>
```

```
      <asp:TextBox ID="txtNum1" runat="server"></asp:TextBox><br /><br />
```

```

<asp:Label ID="lblNum2" runat="server" Text="Number 2: "></asp:Label>
<asp:TextBox ID="txtNum2" runat="server"></asp:TextBox><br /><br />

<asp:Button ID="btnAdd" runat="server" Text="Add" OnClick="btnAdd_Click" />
<asp:Button ID="btnSub" runat="server" Text="Subtract" OnClick="btnSub_Click" />
<asp:Button ID="btnMul" runat="server" Text="Multiply" OnClick="btnMul_Click" />
<asp:Button ID="btnDiv" runat="server" Text="Divide" OnClick="btnDiv_Click" /><br
/><br />

<asp:Label ID="lblResult" runat="server" Font-Bold="True" Font-
Size="Large"></asp:Label>
</div>
</form>
</body>
</html>

```

ServiceClient.aspx.vb:

(I hadn't had access to lab PC during holidays, so this code is used instead, that directly calls the service using raw SOAP via HttpWebRequest.

First of all,

My code builds a raw SOAP envelope from scratch:

```

Dim soapEnvelope As String = String.Format( _
    "<?xml version=""1.0"" encoding=""utf-8""?>" & _
    "<soap:Envelope xmlns:xsi=""..."">" & _
    "<soap:Body>" & _
    "<{0} xmlns=""http://tempuri.org/"">" & _ // Method name (e.g., "Add")
    "<num1>{1}</num1>" & _ // First parameter (a)
    "<num2>{2}</num2>" & _ // Second parameter (b)
    "</{0}></soap:Body></soap:Envelope>", _
    methodName, a, b)

```

This creates a SOAP XML message like:

```

<soap:Envelope>
  <soap:Body>
    <Add xmlns="http://tempuri.org/">
      <num1>5</num1>
      <num2>3</num2>
    </Add>
  </soap:Body>
</soap:Envelope>

```

The code sends this XML via HTTP POST to:

`http://alishbahbashir.somee.com/MathLibServiceAlishbah.asmx`

Critical headers are set:

```
request.Headers.Add("SOAPAction", ""http://tempuri.org/" & methodName & "")
```

```
request.ContentType = "text/xml; charset=utf-8"
```

The service returns a SOAP response like:

```
<soap:Envelope>
```

```
<soap:Body>
```

```
<AddResponse>
```

```
<AddResult>8</AddResult>
```

```
</AddResponse>
```

```
</soap:Body>
```

```
</soap:Envelope>
```

My code extracts the value between <AddResult> and </AddResult>.)

Here's the complete code:

```
Imports System.Net
```

```
Imports System.IO
```

```
Imports System.Text
```

```
Partial Class ServiceClient
```

```
Inherits System.Web.UI.Page
```

```
Private Function CallWebService(methodName As String, a As Integer, b As Integer) As String
```

```
Dim url As String = "http://alishbahbashir.somee.com/MathLibServiceAlishbah.asmx"
```

```
Dim soapEnvelope As String = String.Format( _
```

```
"<?xml version=""1.0"" encoding=""utf-8""?>" & _
```

```
"<soap:Envelope xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance"" " & _
```

```
"xmlns:xsd=""http://www.w3.org/2001/XMLSchema"" " & _
```

```
"xmlns:soap=""http://schemas.xmlsoap.org/soap/envelope/"">" & _
```

```
"<soap:Body>" & _
```

```
"<{0} xmlns=""http://tempuri.org/"">" & _
```

```
"<num1>{1}</num1>" & _
```

```
"<num2>{2}</num2>" & _
```

```
"</{0}>" & _
```

```
"</soap:Body></soap:Envelope>", _
```

```
methodName, a, b)
```

```
Dim request As HttpWebRequest = CType(WebRequest.Create(url), HttpWebRequest)
```

```
request.Headers.Add("SOAPAction", ""http://tempuri.org/" & methodName & "")
```

```
request.ContentType = "text/xml; charset=""utf-8""
```

```
request.Accept = "text/xml"
```

```
request.Method = "POST"
```

```
Using stream As Stream = request.GetRequestStream()
```

```
Dim data As Byte() = Encoding.UTF8.GetBytes(soapEnvelope)
stream.Write(data, 0, data.Length)
End Using

Dim response As WebResponse = request.GetResponse()
Using reader As New StreamReader(response.GetResponseStream())
    Dim result As String = reader.ReadToEnd()

    ' Extract the value inside the result tag (e.g., <AddResult>5</AddResult>)
    Dim startTag As String = "<" & methodName & "Result>"
    Dim endTag As String = "</" & methodName & "Result>"
    Dim startIndex As Integer = result.IndexOf(startTag) + startTag.Length
    Dim endIndex As Integer = result.IndexOf(endTag)

    Return result.Substring(startIndex, endIndex - startIndex)
End Using
End Function

Protected Sub btnAdd_Click(sender As Object, e As EventArgs)
    Dim a As Integer = CInt(txtNum1.Text)
    Dim b As Integer = CInt(txtNum2.Text)
    lblResult.Text = "Result: " & CallWebService("Add", a, b)
End Sub

Protected Sub btnSub_Click(sender As Object, e As EventArgs)
    Dim a As Integer = CInt(txtNum1.Text)
    Dim b As Integer = CInt(txtNum2.Text)
    lblResult.Text = "Result: " & CallWebService("Subtract", a, b)
End Sub

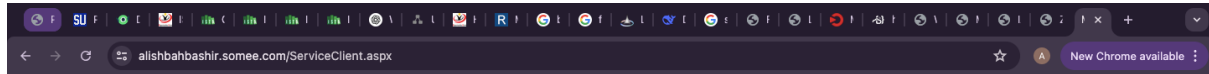
Protected Sub btnMul_Click(sender As Object, e As EventArgs)
    Dim a As Integer = CInt(txtNum1.Text)
    Dim b As Integer = CInt(txtNum2.Text)
    lblResult.Text = "Result: " & CallWebService("Multiply", a, b)
End Sub

Protected Sub btnDiv_Click(sender As Object, e As EventArgs)
    Dim a As Integer = CInt(txtNum1.Text)
    Dim b As Integer = CInt(txtNum2.Text)

    If b = 0 Then
        lblResult.Text = "Error: Cannot divide by zero."
        Return
    End If

    Try
        lblResult.Text = "Result: " & CallWebService("Divide", a, b)
    Catch ex As Exception
```

```
        lblResult.Text = "Error: " & ex.Message  
    End Try  
End Sub  
End Class
```



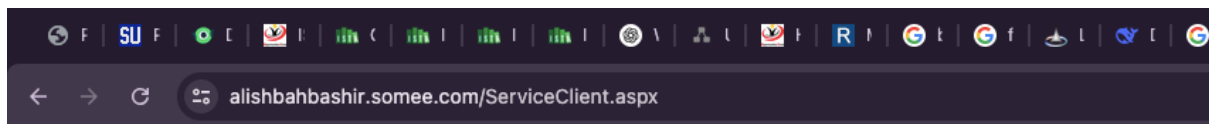
Math Web Service Client

Number 1:

Number 2:

Result: 7

[Web hosting by Somee.com](#)



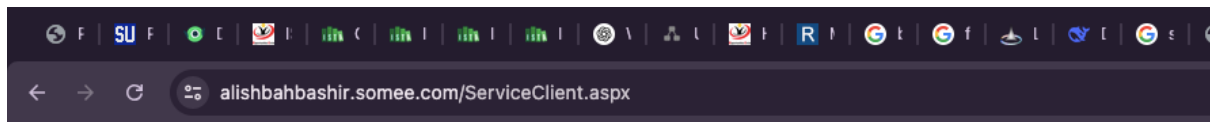
Math Web Service Client

Number 1:

Number 2:

Result: -1

[Web hosting by Somee.com](#)



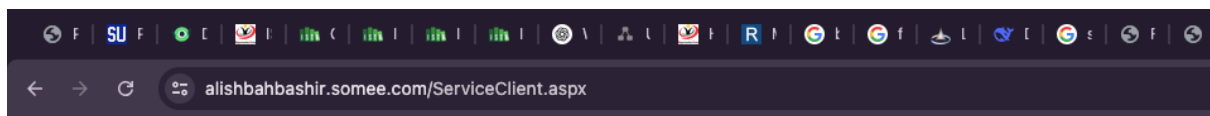
Math Web Service Client

Number 1:

Number 2:

Result: 12

[Web hosting by Somee.com](http://alishbahbashir.somee.com)



Math Web Service Client

Number 1:

Number 2:

Result: 0.75

[Web hosting by Somee.com](http://alishbahbashir.somee.com)