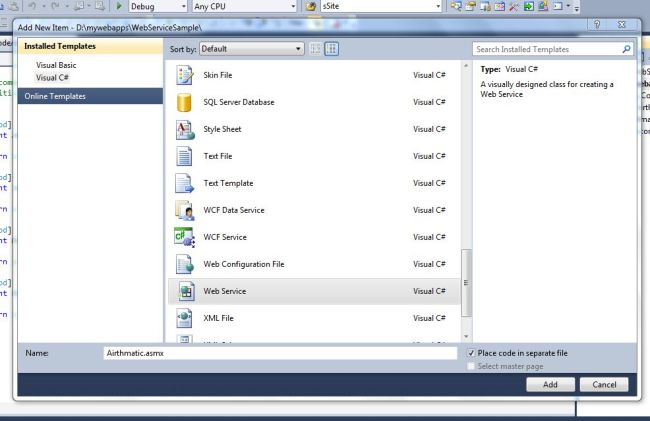
**Create webservices for using calculator funtions using url such.as (Add ◊ Multiply ◊ Divide ◊ Subtract … etc)**

Code:-



using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Services;

/// <summary>

/// used for Airthmatic calculation

/// </summary>

[WebService(Namespace = "http://tempuri.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.

// [System.Web.Script.Services.ScriptService]

public class Airthmatic : System.Web.Services.WebService

{

public Airthmatic() {

//Uncomment the following line if using designed components

//InitializeComponent();

}

[WebMethod]

public int Add(int x, int y)

{

return x + y;

}

[WebMethod]

public int Sub(int x, int y)

{

return x - y;

}

[WebMethod]

public int Mul(int x, int y)

{

return x \* y;

}

[WebMethod]

public int Div(int x, int y)

{

return x / y;

}

}



using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using WebAirthmatic;

public partial class \_Default : System.Web.UI.Page

{

Airthmatic obj = new Airthmatic();

int a, b, c;

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void btnAdd\_Click(object sender, EventArgs e)

{

a = Convert.ToInt32(txtFno.Text);

b = Convert.ToInt32(txtSno.Text);

c = obj.Add(a, b);

lblResult.Text = c.ToString();

}

protected void btnSub\_Click(object sender, EventArgs e)

{

a = Convert.ToInt32(txtFno.Text);

b = Convert.ToInt32(txtSno.Text);

c = obj.Sub(a, b);

lblResult.Text = c.ToString();

}

protected void BtnMul\_Click(object sender, EventArgs e)

{

a = Convert.ToInt32(txtFno.Text);

b = Convert.ToInt32(txtSno.Text);

c = obj.Mul(a, b);

lblResult.Text = c.ToString();

}

protected void btnDiv\_Click(object sender, EventArgs e)

{

a = Convert.ToInt32(txtFno.Text);

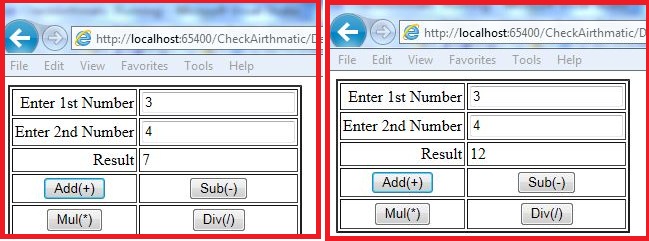
b = Convert.ToInt32(txtSno.Text);

c = obj.Div(a, b);

lblResult.Text = c.ToString();

}

}



**Create webservice for converting xml to json ◊ the service take input xml as a string and return json converted output of that xml.**

Code:-

public HttpResponseMessage Get()

{

string path = HostingEnvironment

.MapPath("~/Employees.xml");

XmlDocument doc = new XmlDocument();

doc.Load(path);

HttpResponseMessage response = this.Request.

CreateResponse(HttpStatusCode.OK);

response.Content = new StringContent

(doc.OuterXml, Encoding.UTF8, "application/xml");

return response;

}

The XML data is assumed to be residing in Employees.xml file. In order to send this data to the client we need to load the XML document. This requires physical path of the XML file. Notice the use of HostingEnvironment class from System.Web.Hosting namespace and its MapPath() method.

Once we get the physical path we load it in an XmlDocument object using its Load() method. The Employees.xml contains XML markup as shown below :

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

<employees>

<employee employeeid="1">

<firstname>Nancy</firstname>

<lastname>Davolio</lastname>

<homephone>(206) 555-9857</homephone>

<notes>

<![CDATA[Education includes a BA in psychology

from Colorado State University in 1970.

She also completed "The Art of the Cold Call."

Nancy is a member of Toastmasters International.]]>

</notes>

</employee>

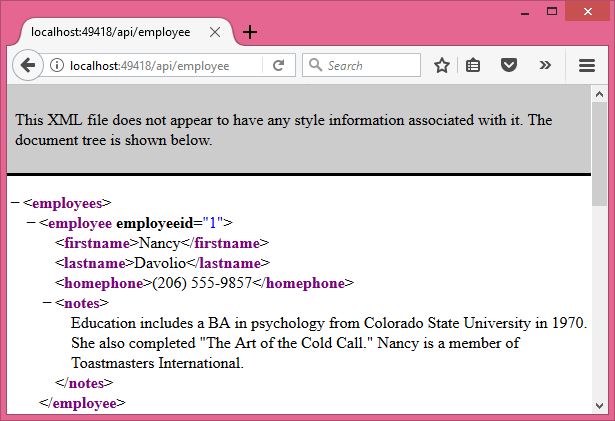
...

...

</employee>

</employees>

Output:-



public HttpResponseMessage Get()

{

string path = HostingEnvironment.MapPath

("~/Employees.xml");

XmlDocument doc = new XmlDocument();

doc.Load(path);

HttpResponseMessage response = this.Request.

CreateResponse(HttpStatusCode.OK);

string json = JsonConvert.SerializeXmlNode(doc);

response.Content = new StringContent(json,

Encoding.UTF8, "application/json");

return response;

}

