

Day 21 Assignment
By Mary Margarette
On 21-02-2022

Create a web service for Mathematical Operations.Example: Factorial, Add, Mul, Div

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

namespace WebApplication1
{
    /// <summary>
    /// Summary description for Web1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // 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 Web1 : System.Web.Services.WebService
    {

        [WebMethod]
        public int Add(int a, int b)
        {
            return a+b;
        }

        [WebMethod]
        public int Mul(int c, int d)
        {
            return c*d;
        }

        [WebMethod]
        public int Div(int e, int f)
        {
```

```
        return e / f;
    }

    [WebMethod]
    public int Factorial(int n)
    {
        int fact = 1;
        for(int i = 1; i <= n; i++)
        {
            fact = fact*i;
        }
        return fact;
    }
}
```

Create a Console Application and consume the webservice

Code:

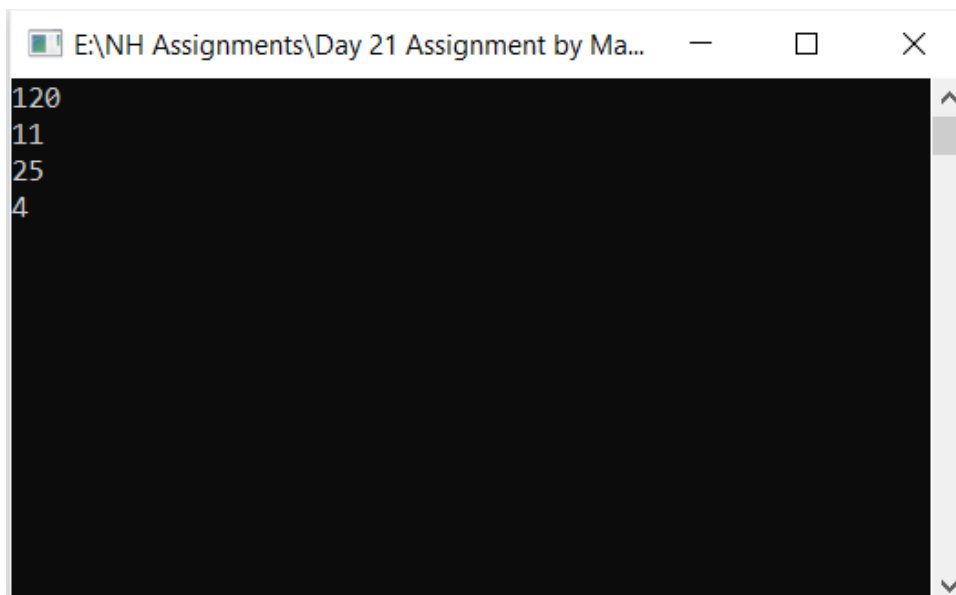
--

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using WebC0.ServiceReference1;

namespace WebC0
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            Web1SoapClient wb = new Web1SoapClient();
            Console.WriteLine(wb.Factorial(5));
            Console.WriteLine(wb.Add(5,6));
            Console.WriteLine(wb.Mul(5,5));
            Console.WriteLine(wb.Div(8,2));

            Console.ReadLine();
        }
    }
}
```

Output:



A screenshot of a Windows console window. The title bar shows the file path "E:\NH Assignments\Day 21 Assignment by Ma...". The console output displays four lines of text: "120", "11", "25", and "4", which correspond to the results of the Factorial, Add, Mul, and Div operations respectively.

```
E:\NH Assignments\Day 21 Assignment by Ma...
120
11
25
4
```

Create a Windows Forms application and consume the webservice[for finding factorial of the number]

Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using WebWin1.ServiceReference1;

namespace WebWin1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int n=Convert.ToInt32(textBox1.Text);
            Web1SoapClient obj = new Web1SoapClient();
            textBox2.Text=obj.Factorial(n).ToString();
        }
    }
}
```

Output:

Form1

input

5

Factorial

120

Put the screen shots of webservice running.

Web1 Web Service

localhost:52159/Web1.asmx

Web1

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

- [Add](#)
- [Div](#)
- [Factorial](#)
- [Mul](#)

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 creating 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](#).