

WSD Lab-4: Creating basic WCF service and Implementing a client application to consume the service

Name : Lakum Vipul Dineshbhai
Roll No : CE066
Batch : A4
Student ID : 22CEUOF066

App.config

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

    <appSettings>
        <add key="aspnet:UseTaskFriendlySynchronizationContext" value="true" />
    </appSettings>
    <system.web>
        <compilation debug="true" />
    </system.web>
    <!-- When deploying the service library project, the content of the config file must be
added to the host's
app.config file. System.Configuration does not support config files for libraries. -->
    <system.serviceModel>
        <services>
            <service name="DemoWcfService.Service1">
                <host>
                    <baseAddresses>
                        <add baseAddress =
"http://localhost:8733/Design_Time_Addresses/DemoWcfService/Service1/" />
                    </baseAddresses>
                </host>
                <!-- Service Endpoints -->
```

```

        <!-- Unless fully qualified, address is relative to base address
supplied above -->
        <endpoint address="" binding="basicHttpBinding"
contract="DemoWcfService.IService1">
        <!--
        Upon deployment, the following identity element should be removed or replaced to
reflect the
        identity under which the deployed service runs. If removed, WCF will infer an
appropriate identity
        automatically.
        -->
        <identity>
            <dns value="localhost"/>
        </identity>
    </endpoint>
    <!-- Metadata Endpoints -->
    <!-- The Metadata Exchange endpoint is used by the service to
describe itself to clients. -->
    <!-- This endpoint does not use a secure binding and should be
secured or removed before deployment -->
    <endpoint address="mex" binding="mexHttpBinding"
contract="IMetadataExchange"/>
    </service>
</services>
<behaviors>
    <serviceBehaviors>
        <behavior>
            <!-- To avoid disclosing metadata information,
set the values below to false before deployment -->
            <serviceMetadata httpGetEnabled="True"
httpsGetEnabled="True"/>
            <!-- To receive exception details in faults for debugging
purposes,
set the value below to true. Set to false before deployment
to avoid disclosing exception information -->
            <serviceDebug includeExceptionDetailInFaults="False" />
        </behavior>
    </serviceBehaviors>
</behaviors>
</system.serviceModel>
</configuration>

```

IService1.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;
```

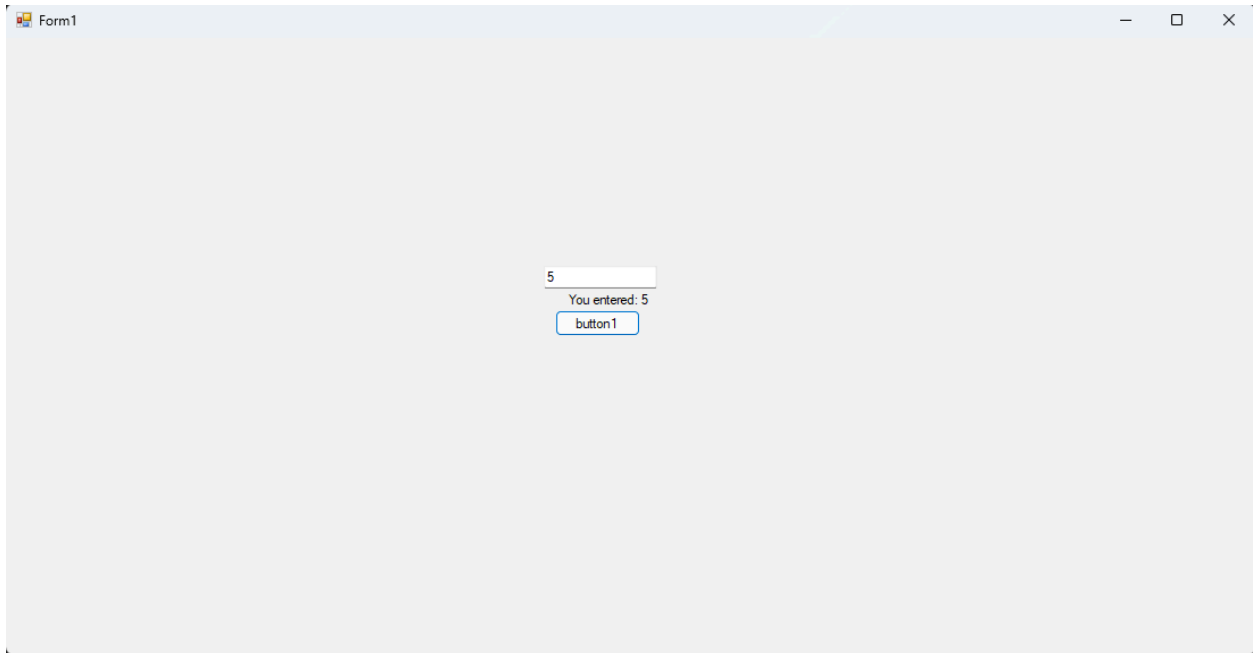
```
namespace DemoWcfService
{
    [ServiceContract]
    public interface IService1
    {
        [OperationContract]
        string GetData(int value);
    }
}
```

Service1.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

namespace DemoWcfService
{
    public class Service1 : IService1
    {
        public string GetData(int value)
        {
            return string.Format("You entered: {0}", value);
        }
    }
}
```

```
}  
}
```



Calculator

IService1.cs

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Runtime.Serialization;  
using System.ServiceModel;  
using System.Text;
```

```
namespace DemoWcfService  
{  
    [ServiceContract]  
    public interface IService1  
    {
```

```

        [OperationContract]
        string Calculate(int op1, int op2, string opr);

    }
}

```

Service1.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

namespace DemoWcfService
{
    public class Service1 : IService1
    {
        public string Calculate(int op1 , int op2 , string opr)
        {
            int ans = 0;
            switch (opr)
            {
                case "+":
                    ans = op1 + op2;
                    break;

                case "-":
                    ans = op1 - op2;
                    break;

                case "*":
                    ans = op1 * op2;
                    break;

                case "/":
                    ans = op1 / op2;
                    break;

                default:

```

```

        return "invalid operation";
    }
    return ans.ToString();
}
}
}

```

Client Side

Form1.cs

```

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;

namespace WindowsFormsApp2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            ServiceReference1.Service1Client sc = new ServiceReference1.Service1Client();
            tbans.Text = sc.Calculate(int.Parse(tbop1.Text), int.Parse(tbop2.Text), tbopr.Text);
        }
    }
}

```

Form1

5 / 5 1

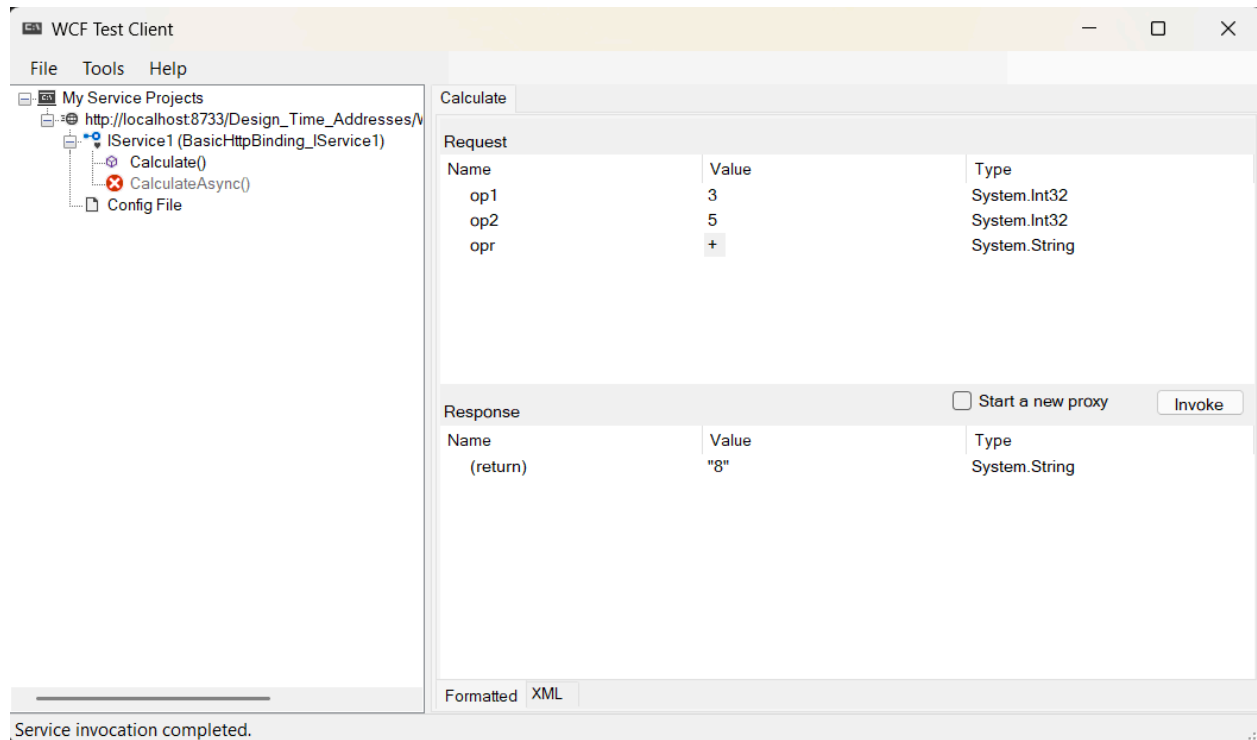
Calculate

Form1

5 add 5 invalid operation

Calculate

TEST Client



SOAP Message

Request Message

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <Action s:mustUnderstand="1"
xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://tempuri.org/IService1
/Calculate</Action>
  </s:Header>
  <s:Body>
    <Calculate xmlns="http://tempuri.org/">
      <op1>3</op1>
      <op2>5</op2>
      <opr>+</opr>
    </Calculate>
  </s:Body>
</s:Envelope>
```


Response message

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
    <CalculateResponse xmlns="http://tempuri.org/">
      <CalculateResult>8</CalculateResult>
    </CalculateResponse>
  </s:Body>
</s:Envelope>
```