

المحاضرة الثامنة

Web Services

د. رامز الخطيب

م. رفاه مختار

- تسمح خدمة الويب (Web Service) لجهاز كمبيوتر باستدعاء طريقة (method) موجودة على كمبيوتر آخر عبر صيغ بيانات مشتركة وبروتوكولات معيارية مثل: XML و HTTP.
- تُنفَّذ عمليات الاستدعاء عبر الشبكة للطرق في NET. باستخدام بروتوكول الوصول للأغراض البسيط (Simple Object Access Protocol) SOAP، وهو بروتوكول مبني على XML ويقوم بتوصيف كيفية تأشير mark up الطلبات requests والأجوبة responses وبحيث يتم تناقلها عبر بروتوكولات معيارية مثل HTTP.
- يُمكن استخدام SOAP التطبيقات من تمثيل وتناقل البيانات في قوالب معيارية XML.

- تُشجع شركة Microsoft مسوقي البرمجيات والأعمال الالكترونية على نشر وتشارك خدمات الويب. إذ طالما تتزايد يوماً بعد يوم أعداد المتصلين بالانترنت، فإن استدعاء التطبيقات للطرق عبر الانترنت يُصبح مرجحاً.

- ومن منظور البرمجة غرضية التوجه، تُمثل خدمات الويب الجيل الثاني من البرمجة غرضية التوجه. إذ عوضاً عن برمجة مجموعة قليلة من الصفوف في مكان واحد، سيُصبح المبرمجون قادرين على استخدام عدد هائل من الصفوف تتواجد في أماكن متفرقة متباعدة.

مثال ١ :

لنأخذ كمثال بسيط، حالة شركة تبيع على الويب أقراص موسيقية كما تُعطي معلومات عن الفنانين.

ولنفرض أن شركة أخرى تختص ببيع بطاقات الحفلات تُقدم خدمة ويب تسمح بمعرفة تواريخ الحفلات القادمة لمختلف الفنانين وتسمح للمستخدمين بشراء بطاقات هذه الحفلات.

يُمكن للشركتين أن تتعاونوا وبحيث يتمكن مستخدمي موقع الشركة الأولى من شراء بطاقات الحفلات مما سيزيد من عدد زوار موقع الشركة الأولى.

كذلك فإن الشركة الثانية ستزداد مبيعاتها إذ أنها ستستفيد من زبائن الشركة الأولى كما يُمكن لها أن تطلب عمولة من الشركة الأولى على البطاقات المباعة.

مثال ٢ :

• يُمثل الشكل التالي صفحة ويب لشركة سفر وسياحة تستخدم من عدة خدمات ويب لمساعدة زبونها:

- خدمة ويب لمعرفة حالة الطقس للوجهة المسافر إليها.
- خدمة ويب لمعرفة سعر الصرف للعملة المحلية.
- خدمة ويب لمعرفة ثمن التذكرة.

Northwind Traders Travel Site

Pick your destination: **Redmond**

The weather
Forecast calls for:



The exchange rate is: **\$1.56**

We can fly you there for only:
\$1,999.98



**Weather
XML Web Service**



**Exchange Rate
XML Web Service**



**Airfare
XML Web Service**



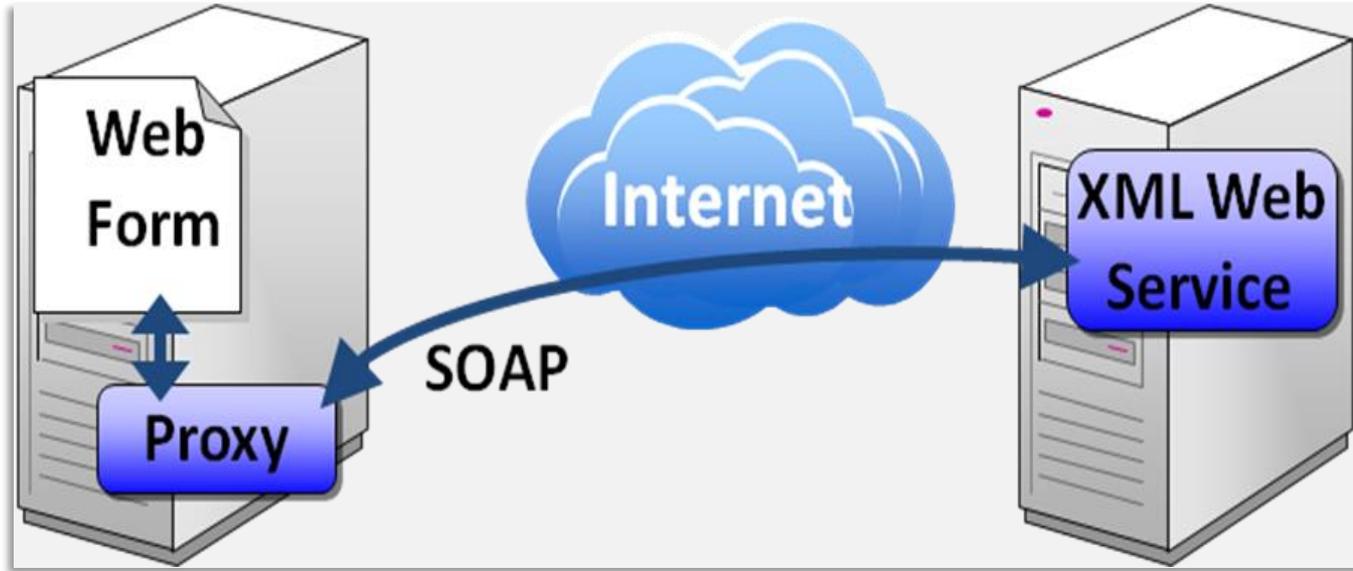
**Airfare
Database**

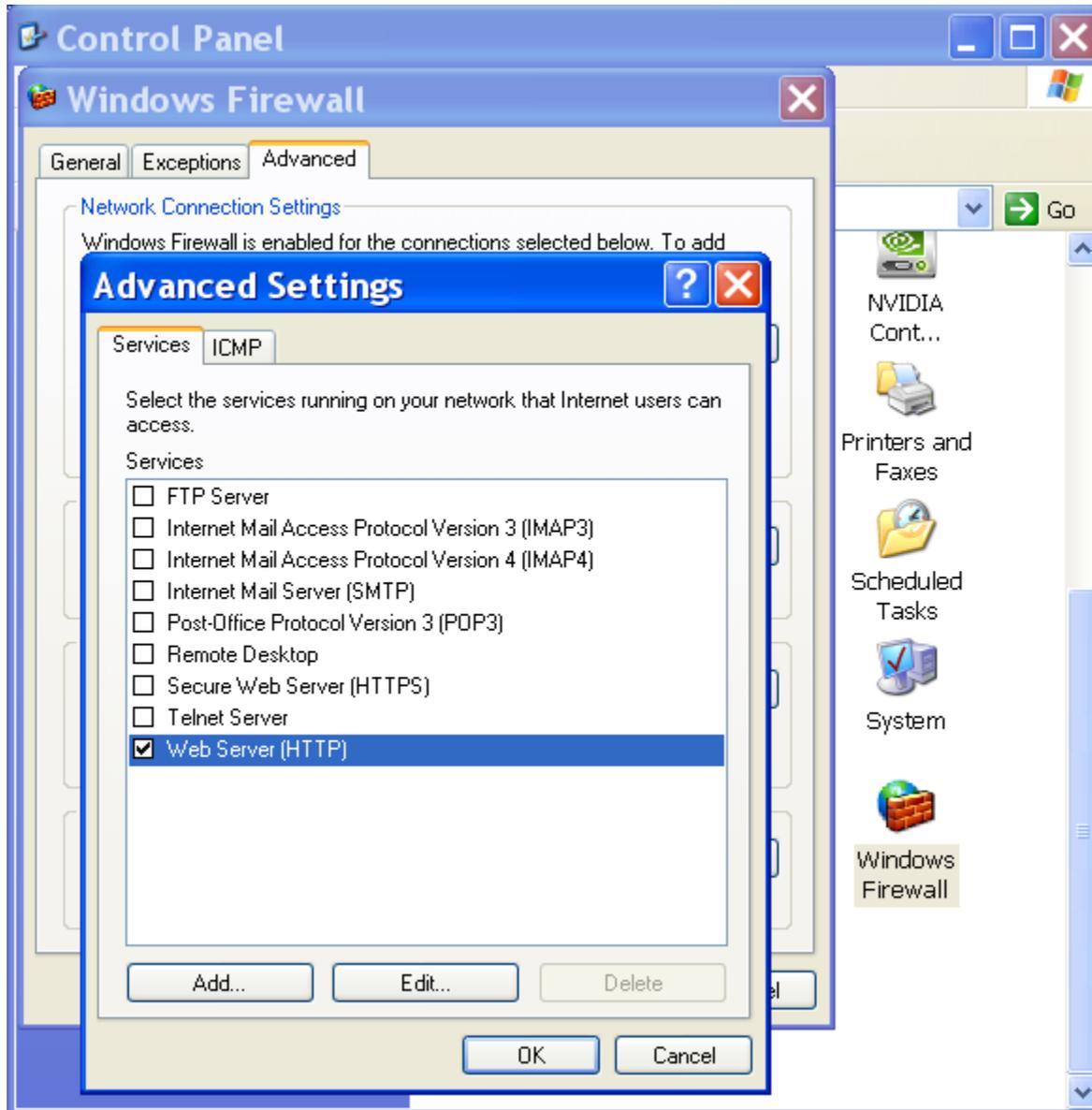


ماهي خدمة الويب (Web Service) :

- إن خدمة الويب عبارة عن تطبيق موجود على جهاز كمبيوتر، بحيث يُمكن لأي جهاز كمبيوتر آخر على الشبكة الوصول إلى هذا التطبيق.
- يُدعى عادة جهاز الكمبيوتر الذي تتواجد عليه خدمة الويب بـ `remote machine`.
- يقوم الموقع الذي يريد الاستفادة من خدمة الويب بإرسال طلب استدعاء لخدمة الويب الموجودة على `remote machine`، فتم معالجة هذا الطلب وإرسال الجواب إلى الموقع المستدعي.
- سنرى أن خدمة الويب هي في النهاية صف `Class` يحوي مجموعة من الطرق `Methods`.

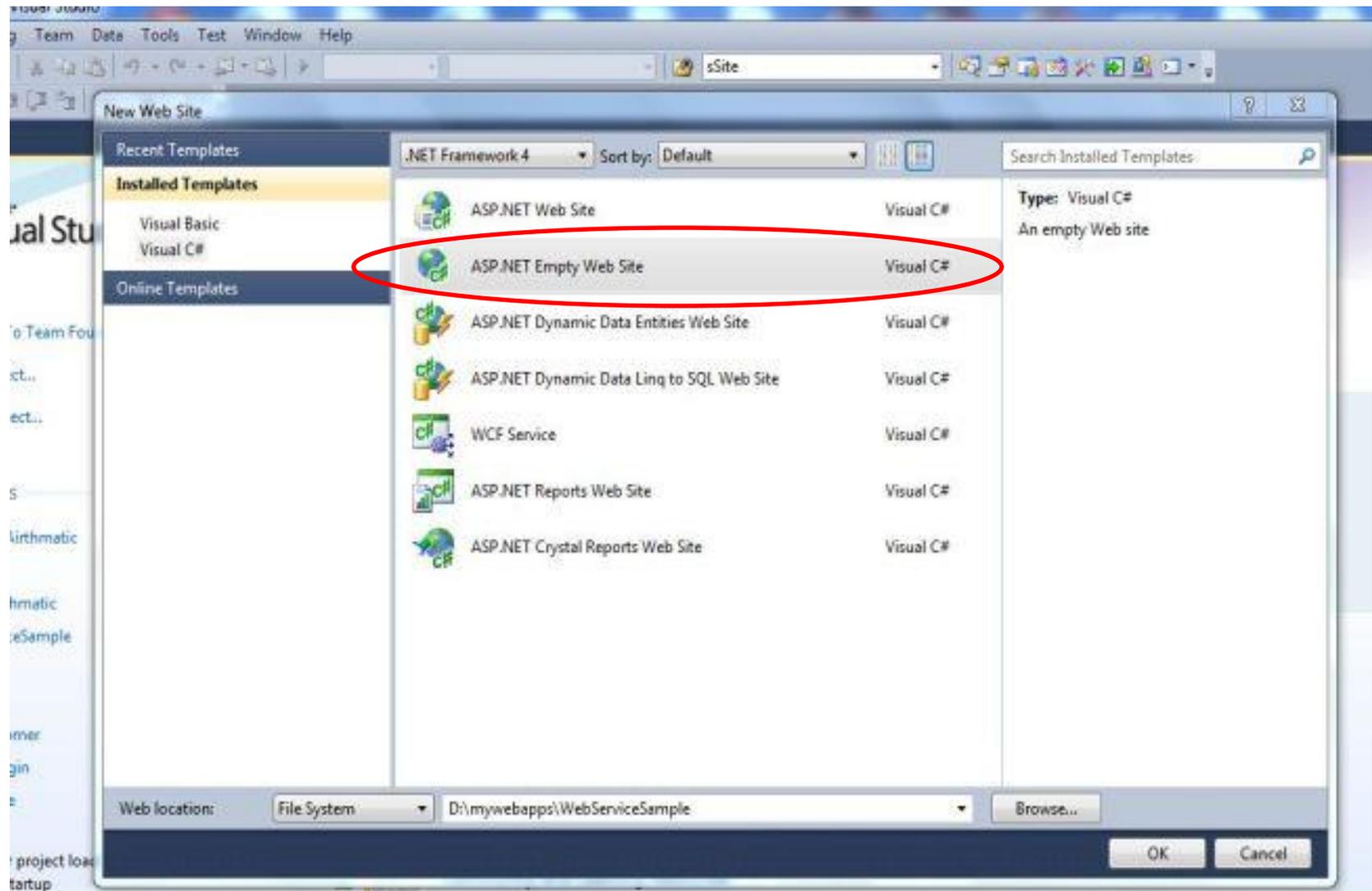
- يتم نقل الطلبات والأجوبة باستخدام البروتوكول **SOAP**. مما يعني أن أي زبون قادر على توليد ومعالجة رسائل **SOAP** يستطيع استخدام خدمة الويب، وذلك بغض النظر عن اللغة التي كُتبت بها هذه الخدمة .



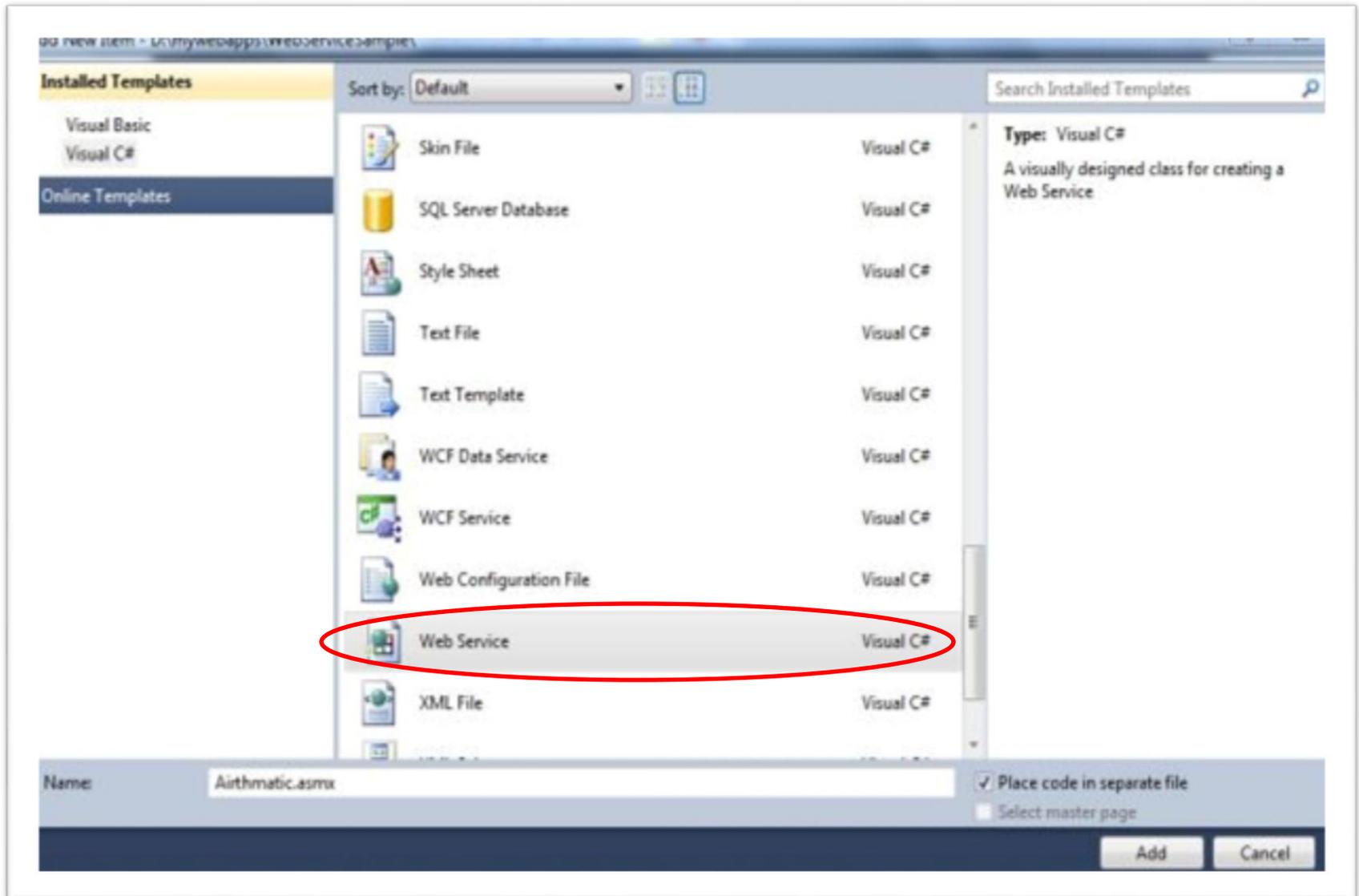


مثال تطبيقي :

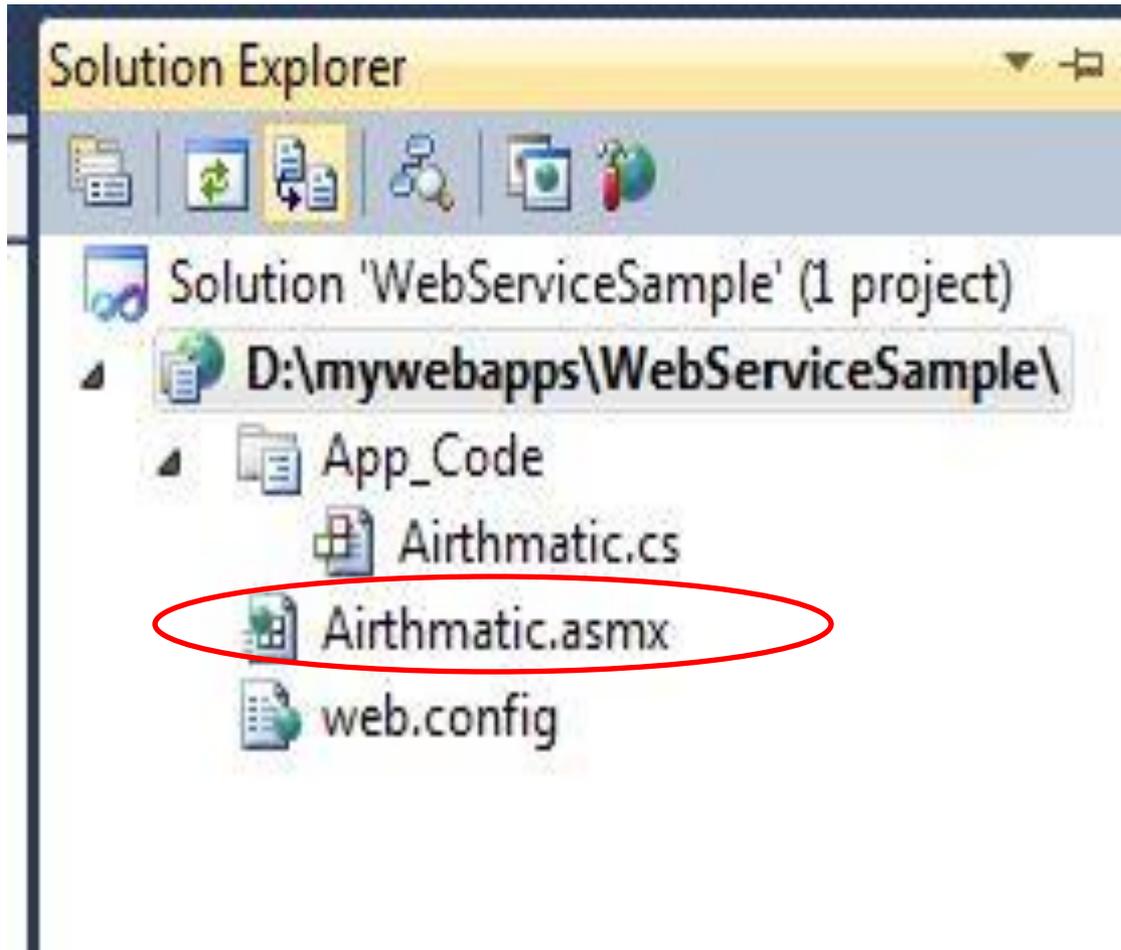
"File" -> "Website" -> "ASP.NET empty website".



"Add new item" → Web Service



نلاحظ أنه تم إنشاء خدمة الويب



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



Arithmetic.cs

```
/// <summary>
/// Summary description for arithmetic
/// </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]
```

1 reference

```
public class arithmetic : System.Web.Services.WebService {
```

0 references

```
public arithmetic () {
```

```
    //Uncomment the following line if using designed components
    //InitializeComponent();
}
```

[WebMethod]

0 references

```
public string HelloWorld() {
    return "Hello World";
}
```

```
}
```

تقوم بإضافة الطرق التالية إلى خدمة الويب:

```
1 reference
public class arithmetic : System.Web.Services.WebService {

    0 references
    public arithmetic () {

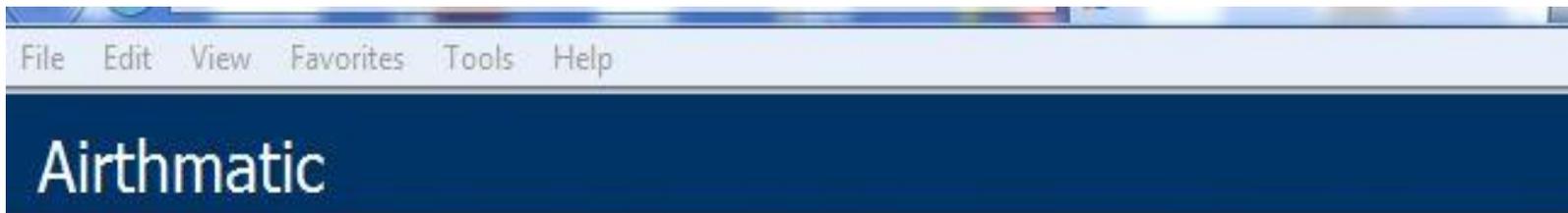
        //Uncomment the following line if using designed components
        //InitializeComponent();
    }

    [WebMethod]
    0 references
    public int Add(int x, int y)
    {
        return x + y;
    }

    [WebMethod]
    0 references
    public int Sub(int x, int y)
    {
        return x - y;
    }

    [WebMethod]
    0 references
    public int Mul(int x, int y)
    {
        return x * y;
    }

    [WebMethod]
    0 references
    public int Div(int x, int y)
    {
        return x / y;
    }
}
```



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

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

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. 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 own domain name. 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 `WebServiceNameSpace` property of the `WebService` class. Below is a code example that sets the namespace to "`http://microsoft.com/webservices/`".

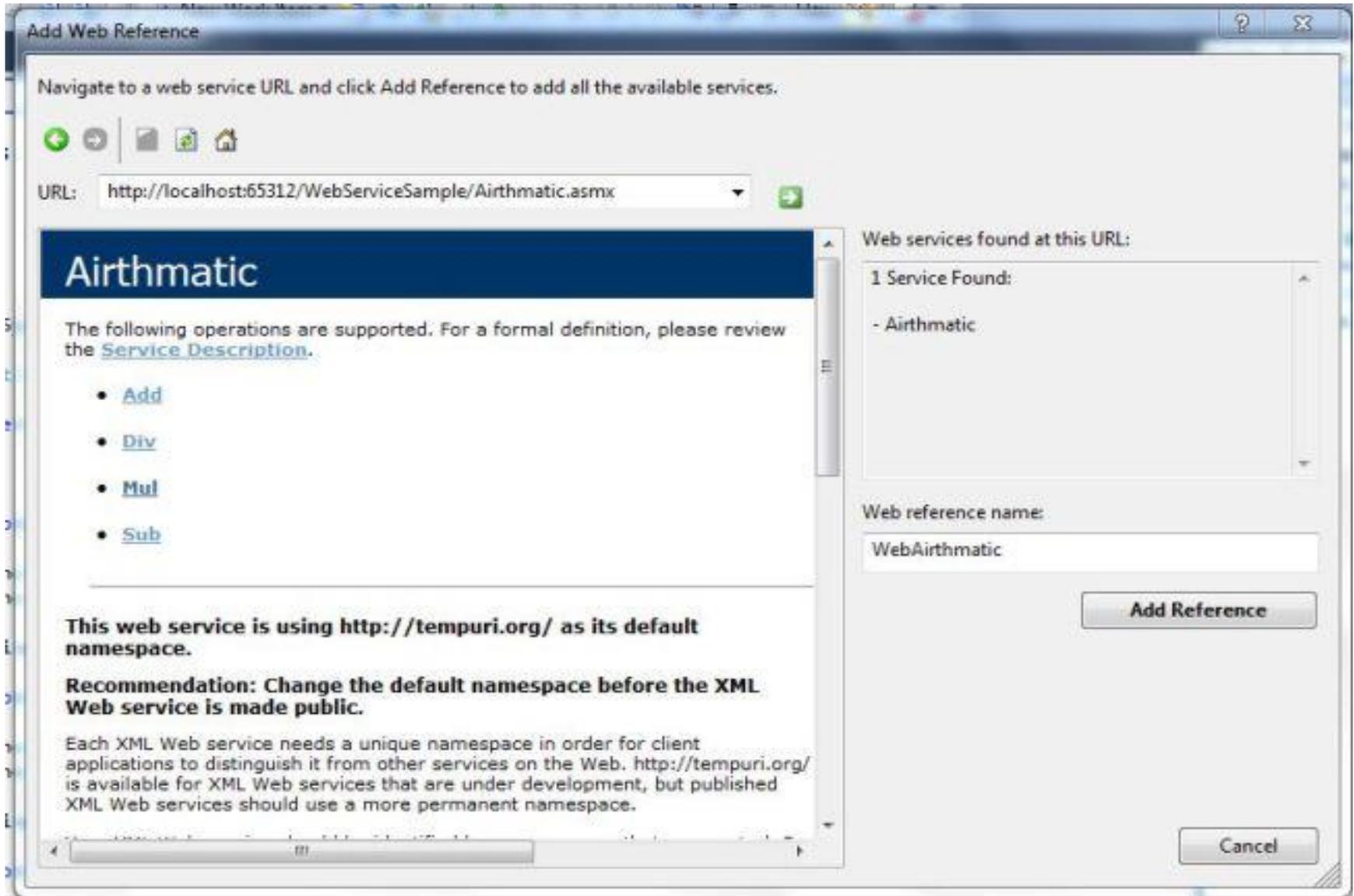
```
<wsdl:definitions targetNamespace="http://tempuri.org/">
- <wsdl:types>
  - <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
    - <s:element name="Add">
      - <s:complexType>
        - <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="x" type="s:int"/>
          <s:element minOccurs="1" maxOccurs="1" name="y" type="s:int"/>
        </s:sequence>
      </s:complexType>
    </s:element>
  - <s:element name="AddResponse">
    - <s:complexType>
      - <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="AddResult" type="s:int"/>
      </s:sequence>
    </s:complexType>
  </s:element>
- <s:element name="Sub">
  - <s:complexType>
    - <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="x" type="s:int"/>
      <s:element minOccurs="1" maxOccurs="1" name="y" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

استخدام خدمة الويب السابقة

body	
Enter 1st Number	<input type="text"/>
Enter 2nd Number	<input type="text"/>
Result	[lblResult]
<input type="button" value="Add(+)"/>	<input type="button" value="Sub(-)"/>
<input type="button" value="Mul(*)"/>	<input type="button" value="Div(/)"/>

```
<table class="auto-style1">
  <tr>
    <td class="auto-style2">
      <asp:Label ID="Label1" runat="server" Text="enter 1st number"></asp:Label>
    </td>
    <td>
      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    </td>
  </tr>
  <tr>
    <td class="auto-style2">
      <asp:Label ID="Label2" runat="server" Text="enter 2nd number"></asp:Label>
    </td>
    <td>
      <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
    </td>
  </tr>
  <tr>
    <td class="auto-style2">
      <asp:Label ID="Label3" runat="server" Text="Result"></asp:Label>
    </td>
    <td>
      <asp:Label ID="Label4" runat="server" Text="Label"></asp:Label>
    </td>
  </tr>
  <tr>
    <td class="auto-style2">
      <asp:Button ID="Button1" runat="server" Text="ADD(+)" />
    </td>
    <td>
      <asp:Button ID="Button2" runat="server" Text="Sub(-)" />
    </td>
  </tr>
  <tr>
    <td class="auto-style2">
      <asp:Button ID="Button3" runat="server" Text="Mul(*)" />
    </td>
    <td>
      <asp:Button ID="Button4" runat="server" Text="Div(/)" />
    </td>
  </tr>
</table>
```

Add a web reference



using WebArithmetic in Client

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using WebArithmetic;
public partial class _Default : System.Web.UI.Page
{
    Arithmetic obj = new Arithmetic();
    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();
    }
}
```

Out put

