1.1、Web Service基本概念

Web Service也叫XML Web Service WebService是一种可以接收从Internet或者Intranet上的其它系统中传递过来的请求，轻量级的独立的通讯技术。是:通过SOAP在Web上提供的软件服务，使用WSDL文件进行说明，并通过UDDI进行注册。

XML：(Extensible Markup Language)扩展型可标记语言。面向短期的临时数据处理、面向万维网络，是Soap的基础。

Soap：(Simple Object Access Protocol)简单对象存取协议。是XML Web Service 的通信协议。当用户通过UDDI找到你的WSDL描述文档后，他通过可以SOAP调用你建立的Web服务中的一个或多个操作。SOAP是XML文档形式的调用方法的规范，它可以支持不同的底层接口，像HTTP(S)或者SMTP。

WSDL：(Web Services Description Language) WSDL 文件是一个 XML 文档，用于说明一组 SOAP 消息以及如何交换这些消息。大多数情况下由软件自动生成和使用。

UDDI (Universal Description, Discovery, and Integration) 是一个主要针对Web服务供应商和使用者的新项目。在用户能够调用Web服务之前，必须确定这个服务内包含哪些商务方法，找到被调用的接口定义，还要在服务端来编制软件，UDDI是一种根据描述文档来引导系统查找相应服务的机制。UDDI利用SOAP消息机制（标准的XML/HTTP）来发布，编辑，浏览以及查找注册信息。它采用XML格式来封装各种不同类型的数据，并且发送到注册中心或者由注册中心来返回需要的数据。

1.2、XML Web Service的特点

Web Service的主要目标是跨平台的可互操作性。为了实现这一目标，Web Service 完全基于XML（可扩展标记语言）、XSD（XML Schema）等独立于平台、独立于软件供应商的标准，是创建可互操作的、分布式应用程序的新平台。因此使用Web Service有许多优点:

1、跨防火墙的通信

如果应用程序有成千上万的用户，而且分布在世界各地，那么客户端和服务器之间的通信将是一个棘手的问题。因为客户端和服务器之间通常会有防火墙或者代理服务器。传统的做法是，选择用浏览器作为客户端，写下一大堆ASP页面，把应用程序的中间层暴露给最终用户。这样做的结果是开发难度大，程序很难维护。 要是客户端代码不再如此依赖于HTML表单，客户端的编程就简单多了。如果中间层组件换成Web Service的话，就可以从用户界面直接调用中间层组件，从而省掉建立ASP页面的那一步。要调用Web Service，可以直接使用Microsoft SOAP Toolkit或.net这样的SOAP客户端，也可以使用自己开发的SOAP客户端，然后把它和应用程序连接起来。不仅缩短了开发周期，还减少了代码复杂度，并能够增强应用程序的可维护性。同时，应用程序也不再需要在每次调用中间层组件时，都跳转到相应的"结果页"。

2、应用程序集成

企业级的应用程序开发者都知道，企业里经常都要把用不同语言写成的、在不同平台上运行的各种程序集成起来，而这种集成将花费很大的开发力量。应用程序经常需要从运行的一台主机上的程序中获取数据；或者把数据发送到主机或其它平台应用程序中去。即使在同一个平台上，不同软件厂商生产的各种软件也常常需要集成起来。通过Web Service，应用程序可以用标准的方法把功能和数据"暴露"出来，供其它应用程序使用。

XML Web services 提供了在松耦合环境中使用标准协议（HTTP、XML、SOAP 和 WSDL）交换消息的能力。消息可以是结构化的、带类型的，也可以是松散定义的。

3、B2B的集成

B2B 指的是Business to Business，as in businesses doing business with other businesses,商家(泛指企业)对商家的电子商务，即企业与企业之间通过互联网进行产品、服务及信息的交换。通俗的说法是指进行电子商务交易的供需双方都是商家(或企业、公司)，她们使用了Internet的技术或各种商务网络平台，完成商务交易的过程。

Web Service是B2B集成成功的关键。通过Web Service，公司可以只需把关键的商务应用"暴露"给指定的供应商和客户，就可以了，Web Service运行在Internet上，在世界任何地方都可轻易实现，其运行成本就相对较低。Web Service只是B2B集成的一个关键部分，还需要许多其它的部分才能实现集成。 用Web Service来实现B2B集成的最大好处在于可以轻易实现互操作性。只要把商务逻辑"暴露"出来，成为Web Service，就可以让任何指定的合作伙伴调用这些商务逻辑，而不管他们的系统在什么平台上运行，使用什么开发语言。这样就大大减少了花在B2B集成上的时间和成本。

4、软件和数据重用

Web Service在允许重用代码的同时，可以重用代码背后的数据。使用Web Service，再也不必像以前那样，要先从第三方购买、安装软件组件，再从应用程序中调用这些组件；只需要直接调用远端的Web Service就可以了。另一种软件重用的情况是，把好几个应用程序的功能集成起来，通过Web Service "暴露"出来，就可以非常容易地把所有这些功能都集成到你的门户站点中，为用户提供一个统一的、友好的界面。 可以在应用程序中使用第三方的Web Service 提供的功能，也可以把自己的应用程序功能通过Web Service 提供给别人。两种情况下，都可以重用代码和代码背后的数据。

从以上论述可以看出，Web Service 在通过Web进行互操作或远程调用的时候是最有用的。不过，也有一些情况，Web Service根本不能带来任何好处，Web Service有一下缺点：

1、 单机应用程序

目前，企业和个人还使用着很多桌面应用程序。其中一些只需要与本机上的其它程序通信。在这种情况下，最好就不要用Web Service，只要用本地的API就可以了。COM非常适合于在这种情况下工作，因为它既小又快。运行在同一台服务器上的服务器软件也是这样。当然Web Service 也能用在这些场合，但那样不仅消耗太大，而且不会带来任何好处。

2、 局域网的一些应用程序

在许多应用中，所有的程序都是在Windows平台下使用COM，都运行在同一个局域网上。在这些程序里，使用DCOM会比SOAP/HTTP有效得多。与此相类似，如果一个.net程序要连接到局域网上的另一个.net程序，应该使用.net Remoting。其实在.net Remoting中，也可以指定使用SOAP/HTTP来进行Web Service 调用。不过最好还是直接通过TCP进行RPC调用，那样会有效得多。

1.3、XML Web Service的应用

1.最初的 XML Web Service 通常是可以方便地并入应用程序的信息来源，如股票价格、天气预报、体育成绩等等。

2.以 XML Web Service 方式提供现有应用程序，可以构建新的、更强大的应用程序，并利用 XML Web Service 作为构造块。

例如，用户可以开发一个采购应用程序，以自动获取来自不同供应商的价格信息，从而使用户可以选择供应商，提交订单，然后跟踪货物的运输，直至收到货物。而供应商的应用程序除了在Web上提供服务外，还可以使用XML Web Service检查客户的信用、收取货款，并与货运公司办理货运手续。

二、Web Service开发

.net平台内建了对Web Service的支持，包括Web Service的构建和使用。与其它开发平台不同，使用.net平台，你不需要其他的工具或者SDK就可以完成Web Service的开发了。.net Framework本身就全面支持Web Service，包括服务器端的请求处理器和对客户端发送和接受SOAP消息的支持。下来我们就一步一步的用Microsoft Visual Studio .net 20058（后面简称VS.Net 2008）创建和使用一个简单的Web Service。

2.1首选建立 一个webservice 服务实例 ：

VS2008 版本

建好之后的页面目录为如下：

打开Service.cs文件  修改成如下图所示：

下面我们开始运行Service.asmx 文件

如果出现的是上面的效果 说明你的WEBSERVICE运行正确，现在记住这个服务地址（即浏览器中的地址），其实个人理解webservice相当于一个接口，这个接口当 然是有地址的，往往在开发我们做好的接口发送给客户（附带接口集成文件等），通过客户端调用接口，在服务器端处理客户的数据 再返回给客户。所以这个地址是有用的，很显示这个WEB服务的地址必须要一直开着，如果断掉的话 会出现什么情况呢？？我们接下来继续在ASP.NET网站如何调用这个服务呢？

2.2 ASP.NET网站调用WEB服务接口（前提你的WEB服务一直开着，也就是上面建好的ASP.NET WEB服务网站是运行着的）

建立一个ASP.NET网站，这个我就不多说了。建立一个Add.aspx页面 拉来三个控件：textbox1,textbox2,label分别用来显示 整数a,整数b,计算的和用label显示。

最关键的一步是如何添加WEB服务呢。

看下图：

选中你项目 右键添加WEB此用，

上面的地址 就是刚才我们运行WEBservice 的地址 输入上去 点击前进 右面就会出现 服务名了。再添加引用OK。我们的WEB服务就添加到我们的网站上来了。

如果出现这个图 就说明你的添加是正确的。

接下来就是后台代码如何调用这个WEB服务了。。。。。。

当然在后台代码中我们事先实例化创建一个服务对象。

代码如下

|  |
| --- |
| protected void Button1\_Click(object sender, EventArgs e)     {         localhost.Service aa = new localhost.Service();         Label1.Text =Convert.ToString( aa.GetSum(Convert.ToInt32(TextBox1.Text.Trim()), Convert.ToInt32(TextBox2.Text.Trim())));     } |

 现在我们运行我的add.aspx 页面

运行后 输入两个数 点击调用 输出结果 为：32 说明我们调用的WEB服务成功了。。。。

2.2 现在如果WEB服务关闭了 再运行这个网站会出现什么情况呢？

出现上面的情况，说明我们的调用WEB服务失败了，所以我们在调用WEB服务时 要使WEB服务处理运行状态。。

一些复杂的接口和这个原理差不多，根据我们系统的实际需求完善WEB服务的功能。

关注.NET开发技术,网站开发,应用系统开发http://www.hnhqwl.com

## **[Webservice服务创建、调用笔记](http://www.cnblogs.com/ysyn/p/4265870.html)**

**引言**

　　以前使用windows服务，于是学习并记录下来：[windows服务的创建、安装、调试全过程及引发的后续学习](http://www.cnblogs.com/ysyn/p/3470826.html" \t "http://www.cnblogs.com/ysyn/p/_blank)。现如今需要用到webservice，对此感觉到很困惑。经过几天的学习、查阅资料，终于大致搞清如何运用，至于它的根本原理，暂时还不能完全理解，以后有机会再详细研究吧。

**定义：**

　　webservice是一个平台独立的，低耦合，自包含的、基于可编程可编程的应用程序，可使用开放的xml标准来描述、发布、发现、协调和配置这些应用程序，用于开发分布式的互操作的应用程序。（百度定义）

　　Web Service技术， 能使得运行在不同机器上的不同应用无须借助附加的、专门的第三方软件或硬件， 就可相互交换数据或集成。依据Web Service规范实施的应用之间， 无论它们所使用的语言、 平台或内部协议是什么， 都可以相互交换数据。Web Service是自描述、 自包含的可用网络模块， 可以执行具体的业务功能。Web Service也很容易部署， 因为它们基于一些常规的产业标准以及已有的一些技术，诸如的xml、HTTP。Web Service减少了应用接口的花费。Web Service为整个企业甚至多个组织之间的业务流程的集成提供了一个通用机制。

 　　个人理解：webservice是一个独立的平台，是用xml来描述的，可以帮组不同平台下的系统建立数据交互机制。比如：在一个很庞大的asp.net网站上创建一个webservice，公司内部的其他部门开发的系统需要引用网站的一些数据，那么在网站上面创建一个webservice，其他系统直接添加web服务即可调用网站上的一个数据或者是方法等等。

**创建一个简单的webservice：**

讲述碰到的第一个小问题：本人使用的visual studio2010，在新建项目时找不到web服务？

直接选择.net4.0，可以选择创建asp.net空模板，然后添加新项选择web服务模板即可。此方法最简单。还有：选择.net2.0,，3.0,3.5，就可以选择webservice模板了，之后需要.net4.0新特性，在项目属性窗口或者页窗口的Build选项卡中选择.net4.0即可。

新建一个空的asp.net项目，添加webservice页面，编写：

 View Code

至此创建了一个webservice，里面包含加减乘除4个方法。如下图显示：

**Asp.net调用webservice：**

下面创建一个asp.net项目调用webservice：

首先引用webservice，将服务的网站网址加入服务引用即可。添加服务引用界面如图：

编写调用的方法：

protected void Button1\_Click(object sender, EventArgs e)

{

string selectFlag = selectOper.Value;

ServiceReference1.WebService1SoapClient web = new ServiceReference1.WebService1SoapClient();

if (selectFlag.Equals("+"))

{

Result.Text = (web.addition(double.Parse(Num1.Text), double.Parse(Num2.Text))).ToString();

}

else if (selectFlag.Equals("-"))

{

Result.Text = (web.subtract(double.Parse(Num1.Text), double.Parse(Num2.Text))).ToString();

}

else if (selectFlag.Equals("\*"))

{

Result.Text = (web.multiplication(double.Parse(Num1.Text), double.Parse(Num2.Text))).ToString();

}

else if (selectFlag.Equals("/"))

{

Result.Text = (web.division(double.Parse(Num1.Text), double.Parse(Num2.Text))).ToString();

}

}

在运行界面内，随即选择“-”号，最后得到的运算如下图：

**Winform调用webservice：**

既然上面在介绍webservice时说了其是夸平台的，那么创建一个winform系统调用webservice。同样先添加webservice，

private void button1\_Click(object sender, EventArgs e)

{

ServiceReference1.WebService1SoapClient web = new ServiceReference1.WebService1SoapClient();

this.txt3.Text = (web.multiplication(double.Parse(this.txt1.Text), double.Parse(this.txt2.Text))).ToString();

}

可以看出winform的调用和asp.net调用基本是相同的。在此也就没有一一验证，只是简单的选择“\*”作为运算符，运行结果如下：

**总结：**

　　上文简单描述了webservice的基本认识，接着创建了一个相当简单的webservice，然后创建一个asp.net项目和winform系统对它进行调用，所有一切基本上完成了webservice的使用过程。大家平时看到的更高级的webservice，或者可以说是更为复杂的webservice吧，这些都是在最简单的webservice基础上增加的一些很复杂的业务逻辑，抽丝剥茧后还是基本的webservice应用。