

XML We service完全实例详细解析 PDF转换可能丢失图片或格式, 建议阅读原文

https://www.100test.com/kao_ti2020/253/2021_2022_XMLWe_ervi_c97_253820.htm 首先,我们必须了解什么是webservice.就概念

上来说,可能比较复杂,不过我们可以有个宏观的了

解:webservice就是个对外的接口,里面有 函数可供外部客户调用(注意:里面同样有客户不可调用的函数).假若我们是服务端,我们写好了个webservice,然后把它给了客户(同时我们给了他们调用规则),客户就可以在从服务端获取信息时处于一个相对透明的状态.即是客户不了解(也不需要)其过程,他们只获取数据. webservice传递的数据只能是序列化的数据,典型的就是xml数据,这里我们也只讨论XML数据的传输. 有了一些对XML webservice的初步了解后,我们将切入正题,即是用一个具体的webservice事例的形式来讲解具体的webservice用法,用具体的事例来讲解一个概念我想怎么也要比单纯的说理能让人容易理解吧. 这里,我们将以一个简单的分布式课件搜索系统为例来讲解.使用VS2003为编译环境,C#为语言,SqlServer2000为数据库.(这个例子来源于一位网上朋友的文章的启发,觉得很能代表webservice的特点,就按那个想法做了这么个系统来示例了) 首先,明确我们要做什么.我们需要一个对客户的接口,也就是个站点,我们把它称做ServiceGatherSite,它是何种形式都无所谓,甚至它本身 并不需要数据库,它只是提供给用户一个查询的接口,真正的服务,普通用户是不接触到的.然后,这里我们还需要若干个提供服务的站点,我们可以称它们为资源站,这里为简单起见,假设有两个资源站,分别叫WebSiteA,WebSiteB,它们可以是不对外公布的,只是为了丰富查询数据而存在.最后,

是我们最需要关注的东西---资源站提供给ServiceGatherSite的服务.两个资源站,就有两个服务,我们称为SiteAService和SiteBService.两个服务间没有任何关系,内部提供的方法也完全没关联,只是需要把方法如何使用告诉ServiceGatherSite,意思是,服务只提供查询接口,返回的数据如何处理,服务本身并不管,全由使用服务的站点分配.写了这么多,算是简要的介绍了下有关XML webservice的概念和我们这个例子的结构,下篇文章,我们将开始真正进入代码的设计阶段. 上篇文章介绍了一些webservice的基本特性和我们例子的结构,这篇文章我们将开始具体的代码编写工作. 这个专题主要讲述的是webservice,因此这里我们的代码以Webservice相关为主,而其他工程,例如:ServiceGatherSite,WebSiteA等,只将简略介绍. 在VS2003中,开发一个webservice并不是件困难的事,首先,我们新建一个webservice项目(文件->新建->项目->C#->Web服务应用程序) 建完这个工程,我们将看到一个叫Service1.asmx的文件,这就是webservice的标准文件,它也有UI的概念,不过我们一般不关注,因此,我们查看其cs代码文件.如果你什么都还没做的话,将看见一个被注释掉的helloworld的WebMethod,把注释去掉,在运行,你就可以得到最简单的webservice运行实例了.点击"helloworld"将执行其方法.显然,这个函数对我们的意义只在于宏观的了解了下web服务的写法. 下面,我们将开始具体介绍webservice的写法.在代码文件里,如果我们写了一个函数后,希望此函数成为外部可调用的接口函数,我们必须在函数上面添上一行代码[WebMethod(Description="函数的描述信息")],如果你的函数没有这个申明,它将不能被用户引用.如:以下是引用片段: [WebMethod(Description="最简单的方法")] public

string HelloWorld() { return "Hello World". } 这个函数就是外部可调用的接口函数,对用户来说相当于一个API.如果某用户在引用了这个服务后,他调用HelloWorld()方法,他就将获得"HelloWorld"这个返回值.看到这里,我们是不是发现,其实webservice并不是那么的神秘,它也不过只是个接口,对我们而言,侧重点依然是接口函数的编写.下面,我将给出我们的例子所需要的接口函数.

```
[WebMethod(Description="查询以获取需要的课件信息")] public XMLDataDocument GetSiteAData(string AssignName) { XmlDocument xd=new XmlDocument(). // DataSet ds=new DataSet(). CStoreProc cp=new CStoreProc("SearchAssign"). cp.AddParIn("@keywords",SqlDbType.VarChar,30,AssignName). cp.AddParOut("@res",SqlDbType.Int). if(cp.SelectProc()) //如果执行成功,存储过程 { cp.myData.EnforceConstraints=false. //不进行格式严格检查 if((int)cp.GetReturnValue("@res")==-1) { string XML="". xd.LoadXml(XML). return xd. } xd=new XmlDocument(cp.myData). XmlNode root1=xd.DocumentElement. XmlNodeList roots=root1.SelectNodes("list"). foreach(XmlNode roota in roots) //为所有元素加上站点名称标记 { XElement Link=xd.CreateElement("SiteName"). Link.InnerText=ConfigurationSettings.AppSettings["SiteName"].ToString(). roota.AppendChild(Link). } return xd. } else return null. }
```

这是获取资源站点信息的一个接口函数.里面大部分的代码,我想对于有一定ASP.NET基础的朋友来说,都应该是一看就明白,这里只说明下CStoreProc,这是我封装的一个存储过程类,主要

功能是执行各种类型的存储过程. 细心的朋友可能会发现这个函数的返回类型似乎比较特殊,是个xml的文档.我们在前面已经说过,webservice只能传输序列化数据,xml显然满足条件,但比如hash表之类的非序列化数据,是不能传输的,xml使用最为广泛,而且考虑到跨平台应用,所以这里我们只以XML数据的传输来示例. 接上篇文章,我们先简单解释下GetSiteAData(string AssignName)函数. 函数功能很简单,只是要返回查询结果,其数据格式是XmlDataDocument.当查询失败时(无匹配查询结果),我们构造一个xml,返回一个空记录.否则,我们把查询后的dataset生成一个XmlDataDocument,接下来,由于该项目的需要,我加入了一个循环,添加dataset里所没有的节点,站点名称.在这之后,算是完成了一个符合我们期望格式的XML数据文档,我们把它返回.好了,webservice的方法函数介绍完了(这里还有个web服务方法,稍后介绍),接下来我们的任务是怎么调用它了.首先把webservice的项目编译完成,假定我们这个服务是针对资源站点A的,我们不妨称其为ServiceA.先单独运行asmx文件,执行GetSiteAData (string AssignName)方法,将提示你输入参数,你输入要搜索的内容,点确认,将返回给你一个XML数据,并在ie上显示出来,这就是你搜索到的内容拉. 100Test 下载频道开通, 各类考试题目直接下载。详细请访问 www.100test.com