

PROGRAMMER TO PROGRAMMER™



Beginning .NET Web Services using C#

.NET Web 服务入门经典

—— C# 编程篇

Joseph Bustos

Karli Watson

侯 彧

著
译



清华大学出版社

758

TP373
B11e

.NET Web 服务入门经典

—— C#编程篇

Joseph Bustos

Karli Watson

著

侯 戡 译

清华大学出版社

北 京

北京市版权局著作权合同登记号：01-2002-3197

内 容 简 介

本书讲述了如何使用 C#和.NET Framework 创建高质量的 Web 服务。本书以清晰的语言、深入的分析 and 真实环境中的应用程序示例详细阐述了 Web 服务各个方面的功能和实际应用。全书共分 9 章，分别讲解了 Web 服务和 XML 的基础知识，Web 服务中的数据表示、命名空间和模式，Web 服务的描述语言(WSDL)和 UDDI，Web 服务和客户端的通信，在.NET 中创建和使用 Web 服务的方法，使用 Web 服务访问和修改数据库的策略和技术，电子商务 Web 服务的设计和实现，Web 服务的部署和使用等主题。

本书适用于希望学习基于 Windows Forms 创建强大、健壮和用户友好的 Web 服务的开发人员阅读。

Beginning .NET Web Services using C#

Joseph Bustos, Karli Watson

EISBN: 1-86100-7574

Copyright©2002 by Wrox Press Ltd.

Original English language Edition Published by Wrox Press Ltd.

All Rights Reserved.

本书中文简体字版由英国乐思出版公司授权清华大学出版社出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，翻印必究。

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

图书在版编目(CIP)数据

.NET Web 服务入门经典——C#编程篇/(美)巴斯特斯，(美)沃森著；侯彧译。

—北京：清华大学出版社，2003

书名原文：Beginning .NET Web Services using C#

ISBN 7-302-06310-9

I. N... II. ①巴...②沃...③侯... III. ①计算机网络—程序设计②C 语言—程序设计 IV. TP393

中国版本图书馆 CIP 数据核字(2003)第 009047 号

出 版 者：清华大学出版社(北京清华大学学研大厦，邮编 100084)

<http://www.tup.com.cn>

责任编辑：胡辰浩

印 刷 者：北京大中印刷厂

发 行 者：新华书店总店北京发行所

开 本：787×1092 1/16 **印张：**21.5 **字数：**445 千字

版 次：2003 年 3 月第 1 版 2003 年 3 月第 1 次印刷

书 号：ISBN 7-302-06310-9/TP·4767

印 数：0001~4000

定 价：45.00 元

出版者的话

随着国际互联网的快速崛起和迅猛发展，计算机之间的互联需求越来越迫切，而目前计算机硬件设备的不兼容性严重束缚了互联网的发展，引发了新一轮的跨平台软件的开发浪潮。软件商纷纷推出新的战略规划和解决方案，Microsoft 提出的.NET 战略就是其中的经典之作。

在经历这场浪潮的洗涤和考验过程中，全球的软件开发人员都迫切需要了解新的软件技术和开发思路。为了满足国内 IT 从业人员的需要，清华大学出版社从 Wrox 出版公司引进了若干套编程系列丛书，“入门经典”系列是其中不可或缺的入门之作。作为世界著名的编程技术图书的出版公司，Wrox 推出的这套“入门经典”系列丛书主要面向编程的初学者、需要了解.NET 策略的程序员，以及需要迅速掌握多门编程工具的程序员。该丛书依旧秉承了 Wrox 公司“由程序员为程序员而著（Programmer to Programmer）”的创作理念，每本书均由世界顶级的编程高手执笔。他们站在资深程序员的高度，循序渐进地为初学者讲述了.NET 的理念和构架、编程基本思想、编程语言基础、程序的控制和调试、Windows 应用程序的开发、对象编程技术、数据库访问技术、Web 程序开发和.NET 构架应用等最新的软件开发知识，同时辅以大量操作性强的程序为示例，为读者提供了清晰的编程思路和宝贵的编程经验。

为了保证该系列丛书的质量，清华大学出版社迅速组织了一批位于 IT 开发领域前沿的专家学者进行翻译，经过编辑人员的进一步加工整理后，现陆续奉献给广大读者。

读者可以从 www.wrox.com 网站下载所需的源代码和获取相关的技术支持。同时，也欢迎广大读者参与 p2p.wrox.com 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

前 言

本书读者

本书适用于想学习基于 Windows Forms 创建强大、健壮和用户友好的 Web 服务的开发人员阅读。您需要了解一些基本的 C# 编程技巧，熟悉数据库的概念。如果您使用过 Visual Studio .NET IDE 将有助于理解本书的内容，但这不是必要条件。

本书内容

第 1 章——Web 服务的基础知识：本章将定义 Web 服务的概念。然后介绍一些 Web 服务的标准。最后学习 Web 服务的具体知识，创建并运行一个简单的 Web 服务。

第 2 章——Web 服务的数据：本章将讲解 XML 的基础知识，如何表示数据结构和表示数据结构的最佳方法。还将讲述命名空间和模式，使我们能够以标准的方式表示数据。最后介绍 .NET Framework 处理 XML 的方式。

第 3 章——描述 Web 服务：本章讲解如何描述 Web 服务，它交换什么数据，以什么格式交换。还将讲解 Web 服务描述语言(WSDL)的有关概念，WSDL 文档的结构以及如何生成 WSDL 文档。最后，我们将讨论通用描述、发现和集成(UDDI)协议，它允许用户在一个集中的目录中发布有关 Web 服务的信息。

第 4 章——Web 服务的使用和通信：本章讲解如何在 Web 服务和它们的客户端之间进行通信，还将讨论使用 HTTP 和 SOAP 访问 Web 服务。此外还介绍了 SOAP 的功能和语法。

第 5 章——.NET Web 服务：从本章开始我们将学习如何在 .NET 中创建和使用 Web 服务。首先介绍如何使用 ASP.NET 创建一个 Web 服务，然后介绍如何使用 VS.NET 创建 Web 服务。接下来将探讨 .NET Framework 中一些高级的 Web 服务技术，包括 HTTP、SOAP 头和 SOAP 扩展。最后讲解了 XML 串行化和如何自定义其行为的内容。

第 6 章——把数据发布为 Web 服务(I)：本章学习 Web 服务的主要用途之一——从数据库中提取数据，然后把它作为一个 Web 服务响应发送。

第 7 章——把数据发布为 Web 服务(II)：本章继续讲解第 6 章的示例，并讨论 DataSet 类和 DataAdapter 对象。

第 8 章——现实世界的 Web 服务(I)：本章讲述 Web 服务框架核心的创建。我们将



深入探讨一系列电子商务 Web 服务的设计和实现，逐步介绍设计的各个阶段，直到数据库和存储过程的创建。

第 9 章——现实世界的 Web 服务(II): 本章讲解 Web 服务的部署和分布，还将讨论 Web 服务的使用。

使用本书的前提

阅读本书需要满足一些软件 and 知识方面的要求。

1. 软件要求

- Microsoft Windows 2000 或 XP
- Visual Studio .NET 完全版或 Visual C# .NET 标准版
- MSDE 或 SQL Server 2000

2. 知识要求

- 了解 C# 语言
- 了解 SQL 语法

客户支持

我们一贯重视读者的意见，并想知道每位读者对本书的看法，包括读者喜欢和不喜欢的内容，以及读者希望我们下一次完善的地方。您可以通过返回本书后面的答复卡或发送电子邮件(地址为 feedback@wrox.com)来向我们反馈意见。请确保反馈信息提到本书的书名。

如何下载本书的示例代码

当您访问 Wrox 公司站点(地址为 <http://www.wrox.com/>)时，通过 Search 工具或书名列表，可以方便地定位需要的书目。然后，单击 Code 列中的 Download 超链接，或者单击本书的详细页面中的 Download Code 超链接，就可以下载相应的示例代码。

当您单击下载本书中的代码时，将会看到带有以下 3 个选项的 Web 页面：

- 如果您是 Wrox Developer Community 的成员(即如果您已经在 ASPToday、C#Today 或者 Wroxbase 上注册过)，就可以使用以往的用户名和密码进行登录

以下载代码。

- 如果不是它们的成员，则会询问你是否愿意注册为会员，以便可以免费下载代码。此外，也可以从 Wrox Press 下载免费的文章。注册为会员后，可以得到本书升级版本和新版本的有关信息。
- 第三个选项是完全绕过注册过程，直接下载代码。

对于本书而言，不注册也能下载代码，但是，如果您愿意注册后下载代码，您的注册信息不会泄漏给第三方。关于这方面的详细条款和情况，可以通过单击下载页面上的相关链接来查看。

从我们的站点上下载的文件都是使用 WinZip 压缩过的文档。保存文件到本地磁盘上的文件夹中后，需要使用一个解压缩程序(例如 WinZip 或 PKUnzip)来解压缩文件。在解压缩文件时，通常将代码解压缩到每一章所在的文件夹中。在解压缩的过程中，应确保解压缩程序(WinZip、PKUnzip 等)被设置为使用原有文件夹名。

勘误表

我们已经尽最大努力确保本书中的文本和代码没有错误，但是错误仍然在所难免。如果您发现本书存在错误，例如拼写错误或不正确的代码段，请反馈信息给我们，我们将不胜感激。勘误表的发送可以节约其他读者学习本书的时间，而且能够帮助我们提供更高质量的信息。您的反馈信息将被检查，如果正确，将被粘贴到本书的勘误页面上，或者在本书的后续版本中使用。

要在我们的站点上找到勘误表，请访问 <http://www.wrox.com/>，并通过 Advanced Search 或者书名列表轻松定位本书页面。然后，单击 Book Errata 超链接即可，该链接位于本书的详细页面中的封面图解下面。

E-Mail 支持

如果您希望直接向详细了解本书的专家咨询本书的问题，可以发送电子邮件到 support@wrox.com，要求在邮件的主题栏中带上本书的书名和 ISBN(国际标准图书编号)的后 4 位数字。一个典型的电子邮件应包括下面的内容：

- 在主题栏中必须有本书的书名、ISBN 的后 4 位数字和问题所在的页码。
- 邮件正文中应包括读者的名字、联系信息和问题。

我们不会给您发送垃圾邮件。我们仅仅需要有用的详细资料，以便节约您和我们的时间。当您发送一封电子邮件时，它将经过下面的一系列支持：

- 客户支持：首先，您的邮件将被递送到我们的客户支持人员手中，并由他们阅



读。对于一些被频繁提到的问题将被归档，并将立即回答有关本书或者 Web 站点的任何常见问题。

- 编辑支持：接着，一些深层次的问题将被提交给对本书负责的技术编辑手中，他们在程序设计语言或者特定的产品上有着丰富的经验，能够回答相关主题的详细技术问题。
- 作者支持：最后，如果编辑不能回答您的问题(这种情况很少发生)，他们将请求本书的作者。我们将尽量保护作者免受干扰，以便不影响其写作。然而，我们也非常高兴转寄给他们一些特殊的问题。所有 Wrox 公司的作者都为他们的书籍提供技术支持。作为回应，他们将发送电子邮件给用户和编辑，进而使所有的读者受益。

Wrox 公司的支持过程仅仅对那些与我们出版的书目内容直接相关的问题提供支持，对于超出常规书目支持的问题，您可以从 <http://p2p.wrox.com>/论坛中的公共列表中获得支持信息。

p2p.wrox.com 站点

为了便于作者和其他人讨论，特将讨论内容加入到 P2P 站点的邮件列表中，而且我们独特的系统将 programmer to programmer™(由程序员为程序员而著)的编程理念与邮件列表、论坛、新闻组以及所有其他服务内容(一对一的邮件支持系统除外)相联系。如果您向 P2P 发送一个问题，应该相信它一定会被登录邮件列表的 Wrox 公司作者和其他相关专家所检查到。无论您是在阅读本书，还是在开发自己的应用程序，都可以在 p2p.wrox.com 站点中找到许多对自己有帮助的邮件列表。

按照下面的步骤可以预订一个邮件列表：

- (1) 登录 <http://p2p.wrox.com>/站点。
- (2) 从左边的主菜单栏选择一个适当的类别。
- (3) 单击希望加入的邮件列表。
- (4) 按照说明订阅并填写自己的邮件地址和密码。
- (5) 回复您收到的确认邮件。
- (6) 使用订阅管理器加入更多的邮件列表并设置自己的邮件首选参数。

本系统能提供最佳支持的原因

您可以加入整个邮件列表，也可以只接收每周的邮件摘要。如果您没有时间和工具来接收邮件列表，可以直接查找我们的在线文档。独特的 Lyris 系统可以将一些没有用的垃圾邮件删除，并保护您的电子邮件地址不被侵扰。当存在加入和离开列表、以及任何有关列表的其他常见问题时，请发送邮件到 listsupport@p2p.wrox.com。

目 录

第 1 章 Web 服务的基础知识	1
1.1 Web 服务的概念	1
1.2 需要 Web 服务的原因	2
1.3 使用 Web 服务的时机	3
1.4 Web 服务的工作原理	5
1.5 .NET 中的 Web 服务	9
1.6 小结	11
第 2 章 Web 服务的数据	12
2.1 XML 的基本知识	12
2.2 存储数据的 XML	19
2.3 模式	22
2.3.1 模式基础	22
2.3.2 模式类型定义	33
2.4 命名空间	39
2.5 处理 XML 数据	44
2.5.1 文档对象模型(DOM)	44
2.5.2 SAX	45
2.6 .NET 中的 XML	46
2.6.1 System.Xml 命名空间	46
2.6.2 操作 XML	56
2.6.3 加载和保存 XML 文档	58
2.7 小结	69
第 3 章 描述 Web 服务	70
3.1 Web 服务描述	71
3.1.1 Web 服务描述语言	71
3.1.2 WSDL 扩展	84
3.1.3 WSDL 示例	92
3.1.4 WSDL 的使用	96
3.2 Web 服务的发现	103
3.2.1 UDDI	103



3.2.2	WS-Inspection	104
3.3	小结	104
第 4 章	Web 服务的使用和通信	106
4.1	简单对象访问协议(SOAP)	106
4.1.1	SOAP 的结构	107
4.1.2	通过 HTTP 使用 SOAP	114
4.1.3	多部分消息	115
4.1.4	SOAP 错误	117
4.1.5	定义 SOAP 消息的结构	119
4.2	Web 服务通信示例	125
4.3	小结	134
第 5 章	.NET Web 服务	135
5.1	ASP.NET Web 服务	135
5.2	Visual Studio .NET Web 服务	151
5.2.1	在 VS .NET 中创建 Web 服务	151
5.2.2	在 VS .NET 中使用 Web 服务	157
5.3	Web 服务的技巧	164
5.3.1	使用消息头	164
5.3.2	SOAP 扩展	180
5.3.3	控制 SOAP 文档格式	196
5.4	.NET 中的 XML 串行化	202
5.4.1	XmlSerializer 类	203
5.4.2	XML 串行化属性	207
5.5	模式的生成	211
5.6	Web 服务的 XML 串行化	212
5.7	小结	217
第 6 章	发布数据为 Web 服务(I)	218
6.1	ADO.NET 概述	218
6.1.1	ADO.NET 对象模型	219
6.1.2	使用 ADO.NET DataReader 类	220
6.2	用于数据检索的 Web 服务	228
6.2.1	WroxEvents 数据库	228
6.2.2	维护静态数据的本地副本	242
6.3	小结	244

第 7 章 发布数据为 Web 服务(II)	245
7.1 简单的数据添加	245
7.2 DataSet	252
7.2.1 DataAdapter 对象	252
7.2.2 更新数据	259
7.2.3 非连接客户端	262
7.3 并发性	268
7.4 小结	273
第 8 章 现实世界的 Web 服务(I)	274
8.1 实现一个电子商务 Web 服务	275
8.2 构建 Web 服务	278
8.3 小结	315
第 9 章 现实世界的 Web 服务(II)	316
9.1 有关安全性的问题	316
9.2 如何部署 Web 服务	319
9.3 使用 Web 服务	324
9.4 小结	328

第1章 Web服务的基础知识

欢迎阅读本书！本书将讲解如何使用 C# 和 .NET Framework 创建 Web 服务，并介绍如何使用 Web 服务执行特定的任务。为了达到这个目标，我们还将讲述其他几种支持 Web 服务的技术。在这些基本知识方面所打下的牢固基础(和对“幕后”所发生事情的了解)能使我们简化所创建的 Web 服务，使之更有效率。这种简化不仅能够改进性能，也能让我们创建更好地与自己的或合作组织的现有代码集成的 Web 服务。在本书的最后，我们将创建一些实际的 Web 服务实现示例，引导您延伸和扩展这些代码，创建自己强大的 Web 服务。

但本章我们不介绍细节方面的内容，只讲解基础知识。首先要讲的自然就是“Web 服务”的确切含义。在定义了这个术语之后，还将介绍 Web 服务的前景为何如此光明，以及何时使用 Web 服务。

接着，将介绍 Web 服务的一些标准，所有这些标准都独立于 .NET(本书讲述的大部分信息适用于利用任何技术开发的 Web 服务)。这些标准将在后面的章节中详细讲解。最后，我们将介绍 .NET 的具体知识；并介绍 Web 服务在 .NET Framework 中的位置，我们将看到使用 C# 创建并运行一个基本的 Web 服务是多么地容易。

在开始之前，有几句忠告。创建 Web 服务根本不像一些外行人想像的那么困难。这些标准的似乎无穷无尽的复杂性，或者为有效使用 Web 服务而需要在设计角度方面作出的转换都很容易让人灰心丧气。然而，一旦您进入了正常轨道，您几乎不会注意到这些问题的存在。当然，标准多种多样，很多标准的说明文档从哪方面来衡量都说不上太好，但对于有能力的程序员来说，没有哪个标准是无法理解的。事实上，如果不是这些标准的表达方法太过愚蠢，领悟它们是件轻而易举的事情。

所以，不要对摆在我们面前的任务感到气馁。虽然为简化应用程序的创建而花些时间来充分理解 Web 服务的标准非常重要，但应记住在很多情况下这些标准的细节是不重要的。一旦闯过了这个基础关，余下的大部分编程工作和标准的 .NET 开发并不难。

1.1 Web服务的概念

“Web 服务”在本书中是指为一些功能提供接口，让它可以被 Web 上的其他程序访问的一种技术。这里所说的功能可能是从数据库获取数据，执行一个数学运算，或者更复杂的任务，比如远程控制哈勃太空望远镜。与本书所谈的 Web 服务进行通信需



要一套具体的标准和协议。这些标准和协议使我们能够创建可以被任何接入 Internet 的用户使用的 Web 服务(虽然在访问服务时可能要求相应的授权),也能使我们使用第三方创建的任何数量的 Web 服务。这些标准指定了下面的内容:

- 一种独立于平台的数据交换格式
- 一种描述数据结构的方法
- 打包数据的标准方法,可能用于跨 Internet 的传输
- Web 服务描述自身、它们能进行的操作和客户程序所需的数据结构的一种方法
- 通过 Web 服务的功能或描述以编程方式定位 Web 服务的一个框架

从非技术的角度来说,大致了解这些标准以及这些标准对我们意味着什么就足够了。关于标准本身更详细的信息,可以参考后面“Web 服务标准”一节中的内容。

1.2 需要 Web 服务的原因

随着 Internet 和网络化的日渐成熟,我们看到了电子 B2B(企业对企业)集成的激增。这里集成的意思是指属于不同企业的计算机通过 Web 直接通信,一般不需要人的干预或需要得很少。对大多数组织来说,这种集成是直到最近一段时间才成为现实的,随着技术一直在提升,成本一直在下降,这使它变得越来越具有吸引力。集成的好处是显而易见的,包括生产率的提高,和人为错误所造成的延迟的减少——这两种收益会直接影响任何公司的底线。

今天,很少有应用程序要求所有资源都位于一个本地服务器上,甚至这些资源也不会只位于一个平台上。相反,应用程序和资源可能分布在多台计算机上,有些用局域网连接起来,有些没有。很多时候,应用程序的两个模块之间仅用 Internet 连接。这是一种分布式的架构。公司应用程序的日趋复杂产生了对分布式应用程序架的更大需求。这意味着即使是单个的公司也会从 B2B 交易要求的电子通信中受益。

所以无论是属于单个组织的计算机还是属于多个组织的计算机都想通过 Internet 彼此交流。借助更基本的 Internet 通信协议,您可以用各种各样的方法达到这个目的,但您可以花几个月的时间和大量的资本开发自己的协议。如果努力了,您最后得到的可能是一个非常高效和创新的结果。但是,当您想使用您的系统和一个不同组织的系统进行通信时,该怎么办呢?两个系统之间的转换也许并不容易,甚至是不可能的,最后您所得到的只是一团糟。

过去人们做过很多努力,引入分布式架构系统以解决这个问题。大多数系统本身的缺点甚至妨碍了它们在企业范围内的部署,更别说在全球的部署。有各种各样的问题,从只能部署到特定的平台,到安全问题,例如要求在公司防火墙中开放端口(这个要求肯定会让您的网络管理员火冒三丈)。所有这些问题的主要原因在于这样一个事实:那

些大的厂商从来没有就用一种办法做事达成过协议，所有的厂商都有目光短浅的时候。但是，这些厂商现在面临着一个无可更改的事实：他们必须共存才能在全球市场中保持竞争力——这也是标准化的一个强大的推动力量！

这个事实要求行业开发新的标准，因此为我们带来了 Web 服务。这些标准可以让我们快速廉价地创建和使用 Web 服务。这既适用于我们自己的 Web 服务，又适用于其他组织提供的 Web 服务。弄清了这个事实，我们接下来考虑 Web 服务在企业应用中的各种可能性，以及企业因此而获得的收益。

首先是通过提供扩展的功能增加利润的能力。即使用 Web 服务，我们可以开放一个业务层供外界使用，有效地开发一条新的销售渠道。例如，有一个组织主要投资于编写代码把内部的 HTML 文档转换为 PDF 格式的文件。Web 服务可以轻松地开放这些代码，允许外部组织访问代码——是以一定的费用！另外，还有简化内部用户，尤其是在家里工作的用户访问这些代码的优势。进一步来说，能够更容易地访问 Web 服务代码也意味着没有必要在不同的位置复制代码，因为每个人都可以访问位于一个中央位置的代码。

Web 服务的另一个主要的优势是，我们可以很轻松地适应它的变化和更新。有一个标准可以让 Web 服务描述它们自己，所以 Web 服务的变化可以很容易地传播到 Web 服务客户端。例如，有一个 Web 服务对应于股市标识符提供股价。一旦找到这个服务，它很容易使用，因为它对使用它和表示结果所要求的数据格式进行了完整的描述。内容格式的变化可以轻松地适应，在很多情况下除了改变数据串行化为 Web 服务格式的方法之外，不需要对客户端代码作什么修改。实际上，客户端代码的修改工作可以交给把从 Web 服务获得的数据开放给我们的应用程序的一个工具去做，这个工具就是代理。这种代理能够响应数据格式的变化对自己进行更新，我们就可以去处理更重要的事情。

然而，Web 服务最强大的特性也许是使用 XML 支持的跨平台的兼容性。即使我们乐观一些，一个公司内部使用的都是相同的操作系统，软件等，我们也不可能期望其他的组织使用相同的设备。因为 Web 服务不作此类的要求，所以从任何类型的设备，例如从移动电话到大型机，只要能连入 Internet，访问给定的 Web 服务都一样容易。

1.3 使用 Web 服务的时机

分布式架构的世界可以分为两个主要的区域：内部世界和外部世界。内部世界关系到一个组织如何在它的内部网络中部署应用程序。外部世界关系到一个企业如何将它的应用程序和业务过程与外部组织集成。



内部世界带来了和新购并的组织集成这样的挑战,这个新购并的组织完全可能使用迥然不同的技术解决方案。Web 服务可以简化这种集成,因为我们不必强迫任何人使用任何特定的技术;只需开放并使用相关的信息,除非必要的情况,否则无需打断正常的操作过程。另一个挑战可能是扩展现在的应用程序,也许是另外添加资源,如新的硬件。当 Web 服务介入时,我们可以在某种程度上打消对新添加的设备在现有的基础结构中融合程度的忧虑,因为不论是添加什么,Web 服务都能够简化异构设备之间的通信。

同样地,在分布式计算的外部世界中的业务关系也可以得到简化。例如,您是一家船运公司,想和一个叫 Sports Warehouse 的公司做业务。Sports Warehouse 通过 Web 服务开放一整套订购系统和产品数据库。您的船运公司可以开放一个 Web 服务让 Sports Warehouse 在您的系统中查询运输成本信息,提交装运要求,或者了解已装运产品的当前状态。把这些 Web 服务提供的功能综合在一起保证了您的公司和 Sports Warehouse 之间的无缝集成,客户可以高高兴兴地购物,不用担心细节方面的因素。自然,这不是什么新概念,但只在几分钟的时间内就做好这种集成,而不管实际的实现使用的是什么技术的能力正是 Web 服务的长处所在。只要两个组织都使用 Web 服务,集成的时间就可以大幅度地缩短。

在阅读了上面的内容之后,您很可能会问什么时候不该使用 Web 服务。在分布式架构的内部世界中,其主要的问题就是性能。通过超文本传输协议(HTTP)进行通信,即使是在局域网中也要慢于更直接的通信。而且,还要在 Web 服务使用的平台中立的格式之间进行数据转换,这更加剧了通信的缓慢。虽然使用 Web 服务可以很快地完成很多任务,但不要忘了考虑其他的要求。在设备兼容性高的场合,使用现有的技术如 Microsoft 的 DCOM 或 Java 的同等技术,可以获得更好的结果。由于本地通信是在已知的端点之间进行的,没有必要使用 Web 服务平台中立的优点。我们不得不偶尔选择牺牲可用性来换取性能的提高。

相同的担心(和解决方案)也适用于分布式架构的外部世界,虽然在外部世界中关心的主要是安全问题。和 Web 服务竞争的技术,如 Microsoft 的 DCOM 或者 Java 的同等技术可以提供更安全的框架,即使使用这些技术会把我们限制在一个具体的平台上。这些竞争技术的另一个优势是,它们大部分都在这个领域沿用已久,有很多省力的工具和一大堆文档,供您在遇到困难时使用。

和很多诸如此类的难题一样,最后的决定往往取决于进行分布式应用开发的人的经验。毕竟,如果一个公司雇佣了一个世界上最好的 DCOM 程序员,那为什么还要强迫他改变做事的方法呢?自然,这个公司面临的也可能是另一种情况,它的程序员恰恰是一个推动使用 Web 服务技术的人……但这是一个完全不同的问题了!

1.4 Web 服务的工作原理

现在我们已经大致描述了使用 Web 服务的动机和 Web 服务在企业中的应用,下面该来认真了解一下 Web 服务的标准了。支持 Web 服务的是 Internet 通信使用的主要协议:超文本传输协议(HTTP)。在大多数情况下,HTTP 将用作 Web 服务通信的传输协议(虽然 Web 服务的规范没有要求)。基本的情况如图 1-1 所示。

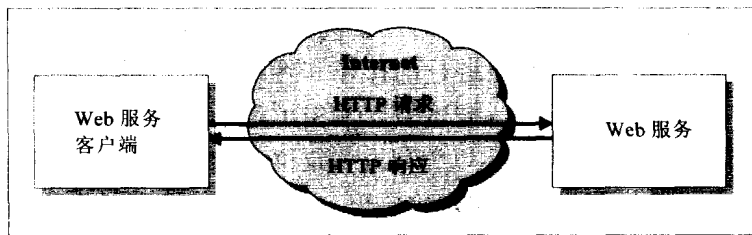


图 1-1

此图显示的是客户端以请求—响应的模式通过 HTTP 与 Web 服务进行通信。实际情况并不局限于此——Web 服务通信也可以是单向的,客户端发出请求但不等待响应。Web 服务也可以启动对话,这种对话可以是单向的,也可以是双向的。

访问 Web 服务和访问 Internet 上的其他内容是非常相像的,它也要求一个 URL。我们在 Web 上冲浪时,使用一个浏览器,在“地址”栏中输入 URL 地址,系统把一个 HTTP 请求发送到和该 URL 关联的 Web 服务器,然后得到一个包含要显示的 HTML 的 HTTP 响应。返回的 HTML 可能是某些处理的结果,如输入 Web 窗体然后提交给 Web 服务器的信息,在这种情况下,发往存放 Web 页的 Web 服务器的 HTTP 请求将包含额外的信息。当查询 Web 服务时,我们以相同的方式访问 URL 和 Web 服务器。Web 服务和相关的协议规范只是告诉我们在 HTTP 请求中应该放置什么信息,以及能够检索到什么内容。

如上所述,Web 服务并不强制使用 HTTP。但是,使用 HTTP 确实有道理。首先,HTTP Web 服务器很容易获得,使 Web 服务的部署不成问题。另外,HTTP 还有一个近亲,即 HTTPS,它可帮助您实现安全的数据传输。使用 HTTP 作为主要传输机制的系统转换为使用 HTTPS 并不困难,因为这两个协议的非安全特性是完全相同的。对于使用 HTTP 通信的 Web 服务也是如此,所以我们可以有一个选择,为我们的 Web 服务提供安全性,同时又不必做任何额外的工作。



Web 服务的标准

在本章的开头对 Web 服务的定义中曾指出,赋予 Web 服务强大力量的标准和协议涵盖了 5 个领域。和这些领域相关的真实技术如下所示:

- 独立于平台的数据交换格式——使用可扩展标记语言(XML)来完成
- 描述数据结构的一种方法——用 XML 模式指定 XML 数据结构
- 打包数据的一个标准方法,也许用于跨 Internet 的传输——XML 数据可能被直接打包成 HTTP 消息,但更常见的是包装在 SOAP 文档中
- Web 服务描述自身、它们能进行的操作和客户端所需的数据结构的一种方法——Web 服务使用 Web 服务描述语言(WSDL)进行描述
- 通过 Web 服务的功能或描述以编程方式定位 Web 服务的一个框架——可以使用通用描述、发现和集成(UDDI)来实现 Web 服务的发现

这些技术和它们在 Web 服务中的应用简述如下:

1. XML

XML 是允许我们把数据编码为纯文本文件的一个标准。这有一个直接的好处:任何设备都可以读取它,因为文本文件在不同平台上都是一样的。自然也有标准要遵守,比如使用什么语言编码,每个字符用多少字节表示,但在可能对 Web 服务感兴趣的所有设备上,这些问题都是可以解决的。

在 XML 和 HTML 之间有着数不清的相似性,包括它们都是标记语言这个事实。标记语言把纯文本分为几个部分,给每一部分的文本加注标记,让这些部分具有定义良好的含义,从而在纯文本中加入结构和含义。例如,使用 HTML 可以把文本的一个区域标记为粗体或斜体,或者使用标记指定插入一个位图图像。XML 的工作原理是相同的,只不过它使用的标记不是专门用于 Web 页的,而是可以表示我们想要的任何内容的标记,例如我们可以把文本的一个区域标记为包含具体特性的特定数据块。

读取一个 XML 文档时,我们对标记进行解释,以把文档中存储的数据转换为我们想要的任何特定于平台的数据格式。在 .NET Framework 中,可以把存储在对象中的数据直接串行化为 XML,然后再轻松地把它反串行化回对象。但串行化为 XML 文档的 .NET 对象不一定要反串行化回对象;任何可以读取 XML 的设备(上面说过,几乎所有设备都可以读取 XML)都可以解释数据,如果必要,把它转换为一个不同的格式。

XML 的灵活性使它可以用于所有的 Web 服务通信。如果需要,甚至可以在 XML 中存储二进制数据。它惟一个真正的缺点是,XML 文档存储数据不像纯二进制表示那样效率高,它必须进行解释(分析)。这在交换数据和使用数据时都会造成性能上的问题,虽然有 XML 分析模型帮助我们进行分析。

我们将在第 2 章讲解 XML 和它的操作。