

Programming .NET Web Services



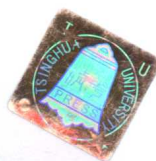
.NET Web服务

编程

Alex Ferrara & Matthew MacDonald 著

天宏工作室 译

O'REILLY®



清华大学出版社

.NET Web 服务编程

Alex Ferrara & Matthew MacDonald 著

天宏工作室 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

O'Reilly & Associates, Inc. 授权清华大学出版社出版

清华大学出版社

图书在版编目 (CIP) 数据

.NET Web 服务编程 / (美) 费莱拉萨 (Ferrara, A.), 麦克唐那 (MacDonald, M.) 著;
天宏工作室译. —北京: 清华大学出版社, 2003.10

书名原文: Programming .NET Web Services

ISBN 7-302-07030-X

I. N… II. ①费… ②麦… ③天… III. 计算机网络—程序设计 IV. TP393

中国版本图书馆 CIP 数据核字 (2003) 第 070654 号

北京市版权局著作权合同登记

图字: 01-2003-0824 号

Copyright ©2002 by O'Reilly & Associates, Inc.

Authorized Simplified Chinese translation edition, by O'Reilly & Associates, Inc., is published by Tsinghua University Press, 2003. Authorized translation of the original English edition, 2002 O'Reilly & Associates, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

本书之英文原版由 O'Reilly & Associates, Inc. 于 2002 年出版。

本中文简体翻译版由 O'Reilly & Associates, Inc. 授权清华大学出版社于 2003 年出版。此翻译版的出版和销售得到出版权和销售权的所有者——O'Reilly & Associates, Inc. 的许可。

版权所有, 未经书面许可, 本书的任何部分和全部不得以任何形式复制。

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。

书 名 / .NET Web 服务编程

书 号 / ISBN 7-302-07030-X/TP · 5168

责任编辑 / 冯志强

封面设计 / Ellie Volckhausen, 张健

出版发行 / 清华大学出版社 (www.tup.tsinghua.edu.cn)

地 址 / 北京清华大学学研大厦 (邮政编码 100084)

经 销 / 各地新华书店

印 刷 / 北京艺辉印刷有限公司

开 本 / 178 毫米 × 233 毫米 27.75 印张 414 千字

版 次 / 2003 年 9 月第一版 2003 年 9 月第一次印刷

印 数 / 0001-4000 册

定 价 / 52.00 元 (册)

O'Reilly & Associates 公司介绍

为了满足读者对网络 and 软件技术知识的迫切需求,世界著名计算机图书出版机构 O'Reilly & Associates 公司授权清华大学出版社,翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly & Associates 公司是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司,同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》(被纽约公共图书馆评为二十世纪最重要的 50 本书之一)到 GNN (最早的 Internet 门户和商业网站),再到 WebSite (第一个桌面 PC 的 Web 服务器软件),O'Reilly & Associates 一直处于 Internet 发展的最前沿。

许多书店的反馈表明,O'Reilly & Associates 是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比,O'Reilly & Associates 公司具有深厚的计算机专业背景,这使得 O'Reilly & Associates 形成了一个非常不同于其他出版商的出版方针。O'Reilly & Associates 所有的编辑人员以前都是程序员,或者是顶尖级的技术专家。O'Reilly & Associates 还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家,而现在编写著作,O'Reilly & Associates 依靠他们及时地推出图书。因为 O'Reilly & Associates 紧密地与计算机业界联系着,所以 O'Reilly & Associates 知道市场上真正需要什么图书。

前言

在过去几年里，我们一直生活在与Web有关的无休止的大肆宣传之中。一些技术专家、专栏作家和CEO们不断宣称Internet将要彻底改变一切，从整个社会的社会结构到购买小商品的方式。虽然已经发生了一些变化，但是其他根本性的预言和雄心勃勃的新技术现在看来似乎遇到了一点麻烦。

人们经常想起的一种说法是“可编程的Internet”，在这种Internet上，各种类型的应用程序和设备不断通信，开放而轻松地共享特性和功能。而事实是，虽然我们现有有了一个连接全球计算机的空前的网络，但是我们还没有开始意识到它的全部潜力。虽然出现了像COM/DCOM、CORBA/IIOP、RMI和XML-RPC这样雄心勃勃的技术，但是目前Internet还是主要用于发送电子邮件和检索HTML页面。如果没有费用高昂的定制工作，那么专用平台上的程序仍然无法使用其他平台上的设备。

现在，Microsoft（以及像IBM和Sun这样的传统竞争对手）已经提议了一个新的“Web服务平台”，该平台基于XML和相关的技术，目标是让应用程序在Internet上共享功能就像在本地计算机上一样容易。其潜力惊人，实现方式简单而优美，并且令人惊诧的是，它是开放的。这种服务从一开始就合并了XML，跨平台支持是它固有的特性，并且可以从参与此平台的开发商那里获得开发工具来处理所有繁重的工作。在Internet上已经散布了Web服务和简单的Web服务目录，从占星术和翻译服务到Microsoft自己的TerraService（提供对1.5兆字节的卫星图像SQL Server数据库的访问）。像eBay和McAfee这样的公司努力使用.NET开发一些创新的Web服

目录

前言	1
第一章 理解 Web 服务	11
为什么使用 Web 服务	11
Web 服务的起源	14
.NET Web 服务体系结构	21
相互竞争的 Web 服务技术	28
其他技术	28
第二章 创建 ASP.NET Web 服务	32
创建 Web 服务 “Hello, World”	32
WebService 属性	52
WebMethod 属性	54
部署 Web 服务	60
使用 .NET 编译器	66
第三章 使用 Web 服务	70
Web 服务的使用者模式	70

访问 Web 服务: HTTP	73
表述 Web 服务: WSDL	86
调用 Web 服务: 客户代理	96
创建客户代理	97
使用代理类	106
通过 Screen Scraper 服务使用 Web 站点	114
 第四章 处理数据类型	122
处理简单数据类型	122
处理复杂数据类型	132
使用复杂数据类型作为参数	142
使用 DataSet 传递数据	148
使用二进制数据	156
 第五章 管理状态	159
有关状态的争论	159
状态管理方案	166
ASP.NET 的会话状态功能	169
会话状态配置	173
例子: 管理会话状态	178
应用程序状态	182
自定义 cookie	183
无状态 Web 服务和票据	185
SOAP 标题	187
 第六章 异步服务	198
异步使用	198
异步服务	214

第七章 缓存和分析	226
ASP.NET 缓存	226
简单的输出缓存	229
数据缓存	235
缓存依赖性	244
使用缓存代替有状态设计	251
分析 Web 服务	256
 第八章 调试、追踪和日志	 262
在 Visual Studio .NET 中进行调试	262
使用异常	266
使用 .NET 记录错误	273
SOAP 扩展	281
高级 SOAP 扩展	289
 第九章 安全性和认证	 297
潜在的脆弱点	297
Web 服务认证	299
IIS 安全特性	300
ASP.NET 安全性	306
程序安全性	317
自定义的认证技术	327
过程安全性	335
 第十章 发布和发现	 338
发布	338
UDDI	339
UDDI 企业注册服务	341
注册企业实体	344

M3543/06

WSDL 和 UDDI	350
Visual Studio .NET 和 UDDI	362
UDDI 消息传递	363
使用 .NET 访问 UDDI	368
WS-Inspection 选择	371
自定义 Internet Explorer 测试页	373
 第十一章 互操作性	376
各个规范的状态	376
实现上的差异	378
SOAP 互操作性	382
WSDL 互操作性	385
HTTP 互操作性和 M-Post	385
已知的互操作问题	386
使互操作性达到最好	389
 附录一 名称空间快速参考	391
 附录二 Web 服务技术	421
 词汇表	429

前言

在过去几年里，我们一直生活在与Web有关的无休止的大肆宣传之中。一些技术专家、专栏作家和CEO们不断宣称Internet将要彻底改变一切，从整个社会的社会结构到购买小商品的方式。虽然已经发生了一些变化，但是其他根本性的预言和雄心勃勃的新技术现在看来似乎遇到了一点麻烦。

人们经常想起的一种说法是“可编程的Internet”，在这种Internet上，各种类型的应用程序和设备不断通信，开放而轻松地共享特性和功能。而事实是，虽然我们现在有了一个连接全球计算机的空前的网络，但是我们还没有开始意识到它的全部潜力。虽然出现了像COM/DCOM、CORBA/IIOP、RMI和XML-RPC这样雄心勃勃的技术，但是目前Internet还是主要用于发送电子邮件和检索HTML页面。如果没有费用高昂的定制工作，那么专用平台上的程序仍然无法使用其他平台上的设备。

现在，Microsoft（以及像IBM和Sun这样的传统竞争对手）已经提议了一个新的“Web服务平台”，该平台基于XML和相关的技术，目标是让应用程序在Internet上共享功能就像在本地计算机上一样容易。其潜力惊人，实现方式简单而优美，并且令人惊诧的是，它是开放的。这种服务从一开始就合并了XML，跨平台支持是它固有的特性，并且可以从参与此平台的开发商那里获得开发工具来处理所有繁重的工作。在Internet上已经散布了Web服务和简单的Web服务目录，从占星术和翻译服务到Microsoft自己的TerraService（提供对1.5兆字节的卫星图像SQL Server数据库的访问）。像eBay和McAfee这样的公司努力使用.NET开发一些创新的Web服

务解决方案，而像 Google 和 Amazon 这样的重要网站已经使用 Web 服务接口展示了它们的一些能力。

Microsoft .NET Framework 是这个平台的一个实现方式，也是一个可以很容易地创建可在其上面互操作的服务的工具。Microsoft 已经（至少是在某种程度上是这样）放弃了它传统上不公开的技术和专用标准的偏爱。Unix 计算机上的 Java 客户可以使用 .NET 服务，而 .NET 客户也可以同样容易地与使用其他大多数非 .NET 语言编写的 Web 服务交互（条件是它们遵循了基于 XML 的新兴的 Web 服务标准协议，如 SOAP 和 WSDL）。使用 COM 和 CORBA 进行分布式编程的复杂性和麻烦已经被简单的、基于 XML 的轻量级标准所取代。概括起来就是：基于 XML 的一般 Web 服务（特别是 .NET Web 服务）是一次从根本上进行改变和改进的尝试，以达到全球连接的长期目标。

本书的内容

本书从开发人员的角度讨论了 Microsoft .NET Web 服务。我们介绍了如何使用 .NET Framework 及其 ASP.NET 工具创建 Web 服务，但更重要的是，我们考虑了设计有效、可扩展并且健壮的 Web 服务所需的最佳方法。本书还介绍了 .NET Web 服务技术的一些基础知识：在幕后工作的开放标准，如 SOAP 和 WSDL。这方面的知识不是使用 .NET 编写 Web 服务所必需的，但是在决定如何使用它的特性以及如何解决它的局限性时，这些知识是非常有用的。

在整本书中，我们还展示了如何“扩展”Web 服务。Web 服务协议最好的一个特性就是它们的简单性。与早期的分布式对象技术不同，基于 XML 的 Web 服务标准没有指定处理安全性、事务或者对象合并的机制。为了提供这些特性，你需要设计和实现自己的解决方案，并利用像 COM+ 和 ASP.NET 这样的平台技术。在整本书中，你将会看到开发这些定制解决方案的最佳方法。

本书的读者对象

与本书所涵盖的内容同样重要的是本书“没有”包括哪些内容——C# 程序设计语言的语法、ASP.NET 的基础知识或者 .NET Framework 的基础知识。本书面向已经

具有一些 .NET 经验的专业开发人员。如果你还不熟悉 C# 和 .NET，那么 O'Reilly 有很多其他优秀的 .NET 书籍供你入门，包括 Jesse Liberty 编著的 *Programming C#* (第二版) (2002)、Liberty 与 Hurwitz 合著的 *Programming ASP.NET* (2002)、Thai 与 Lam 合著的 *.NET Framework Essentials* (第二版) (2002) 以及 Drayton、Albahari 和 Neward 合著的 *C# in a Nutshell* (2002)。

另一方面，如果你是一位经验丰富的 .NET 开发人员，则不需要担心会浪费时间去重复学习有关类型、元数据以及公共语言运行库的基础知识。相反，我们的讨论将会只围绕 Web 服务的主题进行。例如，你将会看到使用已断开的 DataSet 的 Web 服务，但是我们不会介绍 ADO.NET 的基础知识。这种浓缩的主题使本书在目前关于 .NET 的书中显得很独特，其他大多数书籍都尝试讲授 .NET Framework 基础知识、C# 或 VB.NET 程序设计语言以及其他编程基础知识，将这些知识混乱地组织在一起。

注意：如果你熟悉 .NET 平台，那么可能已经意识到 C# 代码可以逐行转换为等价的 VB.NET 语句。不幸的是，由于篇幅限制，本书中的例子只用 C# 代码展示。书中解释了影响 VB.NET 程序员的所有可能的区别，VB.NET 开发人员仍然会发现本书是 Web 服务基础知识和 Web 服务设计最佳方式的极好的参考资料。有关的其他信息，参见 Jose Mojica 编著的 *C# and VB.NET Conversion Pocket Reference* (O'Reilly, 2002)。

使用本书需要些什么

为了最好地使用本书，你需要以下软件：

- Windows 2000 Professional 或 Windows XP (使用 .NET Framework 进行开发的最低要求)。
- Windows 的 Internet 信息服务 (Internet Information Services, IIS) 组件。这个特性是标准 Windows 操作系统的一部分，但是默认情况下没有安装它。在理想情况下，应该在安装 .NET Framework 之前安装它。为了确保它的存在，从 Add or Remove Programs 窗口中选择 Windows Components。
- .NET Framework，可以是可重发布 (SDK) 形式或作为 Visual Studio .NET

零售软件包的一部分。你还可以使用 .NET Framework 的第二个测试版，不过建议使用发布版。不过，不支持第一个测试版（或许会与本书中的例子兼容）。

整本书中都在介绍 Visual Studio .NET (VS.NET) 的 Web 服务特性，你可能会发现它的自动错误检查、集成的调试以及 IntelliSense 语句完成是不可或缺的。不过，也可以非常容易地手工编写 .NET Web 服务，并使用命令行编译器 `csc.exe` 进行编译。本书也介绍了这种方法。在很大程度上，我们重点介绍纯粹的代码以及 Web 服务背后的技术。你使用的开发方法是次要的。如果你使用 VS.NET，我们假设你已经正确安装和设置了它。

本书的结构

本书首先介绍 Web 服务的基础知识：它们的工作方式、如何创建它们以及如何在客户中使用它们。有了 .NET 开发工具，一旦你理解了规则和限制，编写简单的 Web 服务并不比创建一个普通的类更困难。本书的绝大部分——后面的 6 章——都是介绍如何将通常需要的高级特性集成到 Web 服务中。通常，介绍了一些技术细节来解决特定的问题，但是没有提前介绍它们，那可能会因为一些额外的细节而使讨论变得更混乱。例如，SOAP 标题是一种实现自定义状态管理的强大方式，而 SOAP 扩展是一种执行自动追踪的方式。

本书最后介绍了如何发布 Web 服务，使之能够很容易访问，以及你如何打破 Web 服务与非 .NET 代码之间的界限，实现真正的跨平台解决方案。

下面的简介说明了每一章包含的内容：

第一章：理解 Web 服务

历史上总是充满了一些失败的 Web 技术，而 Microsoft 对这些失败的贡献决不是最少的，包括早期像 ActiveX 文档和 Visual Basic Web 类这样的方法。不过，有了 .NET，我们有很好的理由相信 Microsoft 已经从过去的失败中吸取了教训。在本章中，你将了解到为什么 .NET XML Web 服务应该用于其他技术停住脚步的地方，以及 Web 服务在未来的应用中将会起到什么作用。你还将了解到作为 .NET Web 服务基础的一些标准，如 WSDL、SOAP 和 HTTP。

第二章：创建 ASP.NET Web 服务

本章首先介绍每一个开发人员都很喜欢的例子，即经典的 HelloWorld 程序——这次要将它彻底改造为一个 Web 服务。在结束本章之前，你将转移到一个 DNS 搜索服务，学习如何在 Internet Explorer 中使用 HTTP GET 测试 Web 方法。在此过程中，你将学习 .NET 的 Web 服务类型、虚拟目录和 Web 服务部署以及如何在 Visual Studio .NET 中开发 Web 服务或者使用命令行编译器编译它。

第三章：使用 Web 服务

可以使用提供了 XML 解析器的任何语言以及支持 HTTP 协议的任何平台来编写 Web 服务客户。不过，你很可能发现，.NET Framework 中包括的工具使 .NET 应用程序成为了最佳选择。在本章中，你将学习如何生成一个代理类，它将自动处理 Web 服务通信，并允许任何 .NET 应用程序——包括 Windows Forms、ASP.NET 页面甚至控制台实用程序——访问 Web 服务，而不必担心底层的细节。

第四章：处理数据类型

在不同的语言、平台和操作系统之间共享数据带来了许多新的问题和必须考虑的事项。本章深入讨论了 Web 服务数据类型以及如何对它们编码。你将了解到哪些可以作为 Web 方法参数或者返回值发送，以及如何使用数组、DataSet、XML 节点以及自定义结构。我们还将深入讨论 .NET 的 XML 序列化结构，你将学习自定义的数据整形 (shaping)。

第五章：管理状态

对无状态程序与有状态程序的优越性的争论一直都存在着，.NET Web 服务也是如此。本章将讨论 Web 服务为什么不支持属性过程，并探讨导致许多状态维护 Web 服务类失败的艰难现实。我们将考虑客户端和服务端的状态选项，从 ASP.NET 状态服务到自定义的 cookie。我们还将讨论如何使用令牌来跟踪自己的轻量级会话，并将它们与 SOAP 标题结合，以便提供一个优秀的解决方案，用于必须维护一些基本信息的情况。

第六章：异步服务

.NET 使 Web 方法的工作方式就像普通的本地函数一样，本地函数在默认情况下就是同步的，并在允许继续执行之前等待响应。虽然这是默认的行为，但是没有什么理由让它来决定你如何在成熟的应用程序中使用 Web 服务。本章展示

了如何取消这种要求并解放你的代码,使用一般的优化模式来使你的客户更有效。这些技术包括同时调用多个方法、使用回调和事件通知以及创建多线程客户。我们还将讨论如何创建这样的 Web 方法:创建一个异步组件,让它开始工作,然后立即返回。这允许你实现一些高级技术,如自动进度跟踪和批处理。

第七章:缓存和分析

为了创建一个成功的 Web 服务,需要在编写代码时考虑到性能。本章展示了使用输出缓存加速访问,使用数据缓存替代或补充状态管理的最佳方法。我们还将讨论如何使用性能计数器并分析 .NET Web 服务。这允许你在部署 Web 服务之前就了解到所尝试的性能增强的结果。

第八章:调试、追踪和日志

从程序员的角度来看,.NET Web 服务可能与传统的类相似,但是它们不能引发普通的 .NET 异常。本章将讨论如何控制错误,并使用特殊的 SOAP 异常通知客户。你还将学习如何使用 Visual Studio .NET 的集成调试器、如何在多项目 Web 服务解决方案中使用其著名的单步执行,以及如何对自动日志和追踪使用 .NET Framework 服务。本章最后介绍了 SOAP 扩展以及一些示例代码,这些代码展示了如何使用自定义扩展来检查 .NET 的对象层,并分析或记录发送的原始 SOAP 消息。

第九章:安全性和认证

Web 服务标准目前没有提供任何集成的安全机制,但是你有无数个选择,从传输层安全到 IIS 认证、到 ASP.NET 自己的认证服务。在本章中,你将学习怎样才能保护 Web 服务——更重要的是,哪些技术最合适。

第十章:发布和发现

现在,你已经创建了完美的 Web 服务,怎样才能让别人了解这一点呢?本章将讨论一些允许客户获得他们需要的 Web 服务的技术,包括从简单的 DISCO 文件到在线 Web 搜索引擎和 UDDI 注册服务(让企业共享有关其内部处理的细节)的所有技术。我们将讨论使用 DISCO 文件的静态和动态发现机制,DISCO 文件将 Web 服务合并到简单的组中。我们还将讨论 UDDI 注册服务、如何使用它注册企业和添加服务,并考虑它是否已经准备好投入使用了。

第十一章:互操作性

本章重点讨论服务互操作性,讨论在 SOAP 实现之间可能出现的互操作问题以

及目前做了哪些工作来消除SOAP实现中的差异。本章还介绍了你可以采取的一些简单措施,以确保 .NET Web 服务对其他实现方式来说是可访问的。

附录一: 名称空间快速参考

这个附录提供了System.Web.Service名称空间中的主要类的一份按字母顺序排列的参考资料。

附录二: Web 服务技术

这个附录提供了一个列表,按类别列出了一些最常提到的 Web 服务标准和技术,以及到何处查找更多的信息。

排版约定

本书中使用了以下字体约定:

斜体 (*Italic*) 用于以下情况:

- 路径名、文件目录以及文件名
- Internet 地址, 如域名和 URL
- 新术语 (出现在定义它们的地方)

等宽字体 (Constant width) 用于以下情况:

- 代码示例和命令
- .NET 程序中的名称和关键字, 包括方法名称、变量名称、名称以及名称空间

建议与评论

请将有关本书的意见和问题按以下地址发给出版社:

美国:

O'Reilly & Associates, Inc.

101 Morris Street

Sebastopol, CA 95472

中国：

100080 北京市海淀区知春路 49 号希格玛公寓 B 座 809 室

奥莱理软件（北京）有限公司

本书有一个 Web 站点，列出了例子、勘误表以及其他所有信息。你可以在以下地址访问这个页面：

<http://www.oreilly.com/catalog/prognetws>

有关本书的意见或者要提出技术问题，请向以下地址发电子邮件：

bookquestions@oreilly.com

info@mail.oreilly.com.cn

有关图书、会议、资源中心以及 O'Reilly Network 的更多信息，请访问 O'Reilly Web 站点：

<http://www.oreilly.com>

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

致谢

编写一本介绍 .NET Framework 任何一个方面的书籍都是一个挑战。书店的书架上充满了介绍 .NET 的书籍，但是很少有书介绍了实际的开发人员在使用这个新平台时遇到的真正问题和难点。详细介绍低层的细节对于像 O'Reilly 这样的出版商来说是非常适合的任务，O'Reilly 的每一个人都值得因为他们的技术而获奖。如果没有 O'Reilly 的许多人的帮助，本书是不可能完成的，这些人包括 John Osborn，他看到了我们优势互补的风格，并将我们组织到一起来共同编写这本书，还有 Nancy Kotary，她使本书在无数次检查和修订的过程中没有离题（即使这需要每天发数十封电子邮件）。我们还要衷心感谢技术审校们，他们及时提出意见并测试我们创建的代码，包括 Martin Gudgin、Chris Jones 和 B. Robert Helm，还有 Brian Jepson，虽然他和这个项目不直接相关，但总是可以指望他提供有益的反馈或者联系 Microsoft 的人员。没有这些人以及 O'Reilly 其他许多人的帮助，本书是不可能完成的。