

SAMS

〔美〕 Kenn Scribner , Mark C. Stiver 著
刘勇 宋继斌 译

应用 SOAP 实现 XML Web 服务

.NET 精选系列

人民邮电出版社

6 7P312xM
S21
.NET 精选系列

应用 SOAP 实现 .NET XML Web 服务

[美] Kenn Scribner, Mark C. Stiver 著

刘 勇 宋继斌 译



A0979832

人民邮电出版社

图书在版编目 (CIP) 数据

应用 SOAP 实现 .NET XML Web 服务 / (美) 斯特瑞伯纳 (Sturibner, K.), (美) 斯蒂弗 (Stiver, M. C.) 著; 刘勇, 宋继斌译. — 北京: 人民邮电出版社, 2002.5
ISBN 7-115-10255-4

I. 应... II. ①斯... ②斯... ③刘... ④宋... III. ①通信协议, SOAP ②可扩充语言, XML—程序设计 IV. ①TN915.04 ②TP312

中国版本图书馆 CIP 数据核字 (2002) 第 022420 号

版 权 声 明

Kenn Scribner, Mark C. Stiver: Applied SOAP: Implementing .NET Web Services

Copyright © 2002 by Sams Publishing

Authorized translation from the English language edition published by the Sams Publishing.

All rights reserved.

本书中文简体字版由美国 Sams 出版公司授权人民邮电出版社出版。未经出版者书面许可, 对本书任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

.NET 精选系列

应用 SOAP 实现 .NET XML Web 服务

- ◆ 著 [美] Kenn Scribner, Mark C. Stiver
译 刘 勇 宋继斌
责任编辑 陈冀康
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67180876
北京汉魂图文设计有限公司制作
北京朝阳展望印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 20.5
字数: 545 千字 2002 年 5 月第 1 版
印数: 1-4 000 册 2002 年 5 月北京第 1 次印刷

著作权合同登记 图字: 01 - 2001 - 4095 号

ISBN 7-115-10255-4/TP · 2845

定价: 35.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

前 言

本书讲述的是为 Internet 建立应用程序的一种新的方法——该技术被称为“Web 服务”(Web Services)。

Internet 的相关领域已经发生了巨大的变化,尤其是在信息和服务的共享方面。如果暂时忽略销售广告中的那些浮夸成分,你会发现 Web 服务的确是一种使用现有的、经过证明的 Internet 技术让多个应用程序共享信息的新途径。

但 Web 服务的作用范围并不仅仅限于 Internet。它还为服务(services)程序的共享提供了机会,使之可以在同一台机器上让企业内部网(intranet)中的不同应用程序之间得到共享,甚至还可以在某些设备之间实现服务程序的共享。

Web 服务提供了互操作性(interoperability),而 XML 是使之得以实现的基础部分之一。然而,仅有 XML 是不够的——简单对象访问协议(Simple Object Access Protocol, SOAP)为各种系统使用 XML 作为通信协议定义了一种共同的途径。

将 SOAP 应用于 Web 服务

笔者在自己的前一本书 *Understanding SOAP: The Authoritative Solution* 中分别从分布式应用和远程过程调用(Remote Procedure Call, RPC)的观点出发讨论了 SOAP,并着重强调了一些协议专有的特性。尽管并非所有人都想了解 SOAP 的细节,理解自己选择的协议的实现方式肯定能够让你成为一名更称职的 Web 服务开发人员。

自 *Understanding SOAP: The Authoritative Solution* 一书出版以来出现了许多新的技术,例如 WSDL 和 UDDI,这些技术使将允许使用 SOAP 的那些系统的能力得以扩充。与此同时,涌现出许多新的 SOAP 实现方式(根据最新的非官方统计,约有 68 种实现方式),而整个行业都在注视着 Web 服务在各种现实应用中获得发展的动力。

正是由于有许多这样的变化,我们才在本书中采用了一个稍有不

同的切入点：我们决定在更广义的范围内探讨 SOAP——适当地将 SOAP 应用于 Web 服务。尽管 SOAP 并非是实现 Web 服务的唯一途径，但其他可供选择的方法——例如直接使用 HTTP GET 和 POST——却缺乏灵活性，因而在本书中将不会加以强调。

相对于讨论 RPC，本书把注意力更多地投入到面向文档的（document-oriented）通信上。使用 SOAP 作为一种 RPC 机制尤其有助于保持对象的一致性和现有 RPC 协议支持的其他典型特性。事实上，大多数现有的实现方式都是基于 RPC 途径的，但是这将很快发生改变。在 business-to-business（B2B）环境中，基于文档的解决方案更直接，并且更易于使用结构语言来描述。我们将尽力从正反两方面来讨论 RPC 和 XML 文档的使用，因为，随着时间的推移，这将成为一个互操作性问题。

正如我们提到过的那样，你不需要再建立自己的 SOAP 实现方案（或许这让你觉得有些失望）。正如你可以想见的那样，现在已经有许多适用于各种不同的编程语言和平台的好的实现方案。主流方案包括 Apache 公司的 Java SOAP Toolkit（即 IBM 公司原来的 SOAP4J）、Microsoft 公司的 SOAP Toolkit 以及一些用于 Perl、C++ 和其他语言的不同库程序。

既然如此，我们为何不选择讲解 Microsoft 公司的 .NET 框架（.NET Framework）呢？最初，我们做出这种选择是因为 .NET 已经建立了一种全新的运行时（runtime），它是专门为 Internet 应用程序开发而设计的。但是，随着逐步深入的了解，我们又发现部分原因在于 .NET 本身就不错。它特别容易激发那些从事过多年 Windows 开发工作的人们的兴趣。Windows 开发一直具有某些通病，这对于一个经历过多次改头换面和结构性变化的系统而言不足为奇。.NET 再一次为一切从头再来提供了一个场所。Microsoft 公司的开发人员对那些在计算机行业中大行其道的技术进行了长期的审视，并针对它们进行了改进。最终的结果就是一个新的框架的诞生，这个框架用来建立一些应用程序，这些应用程序简化了程序设计模型，同时保持了对现有的某些结构（例如 COM）的向下兼容。

我们早就知道在软件开发中没有万金油。要成为一个好的软件开发人员，注重实效才是最重要的。你需要了解事物的所以然，能够对自己选择的技术加以评价，还需要乐于尝试新事物。或许正是所有这些才促使你选择本书进行阅读，而它们也是我们决定编写这本关于 Web 服务和 .NET 的书籍的原因。

总之，本书有三个主题。首先，本书从结构和设计的立场出发逐步向 Web 服务逼近，以分析在这种新的范例（paradigm）中应该解决的那些问题，尤其是在 .NET 框架中。其次，本书演示了如何使用 .NET 框架和新的 Visual Studio .NET 工具来实现 Web 服务。最后，本书还讨论了关于 Web 服务和 SOAP 的更高级的问题，包括 W3C 在 XML 协议（XML Protocol，XMLP）方面所做的工作的有关信息。

当今的企业级应用程序

软件开发是一门不断变化的艺术，而程序员则担负着与最新的技术趋势保持同步的艰巨任务。本书的目的在于教会你如何使用 .NET 框架应用具有实效的解决方案来建立 Web 服务。Web 服务的最大优点在于它适用于许多方面的问题，这些问题包括复杂的 B2B 系统集成、连网的设备、分布式开发，以及许多其他的问题。但是，它对互操作性和简单性的承诺为我们带来了光明的前景。

行业的期待

我们所从事的领域中的许多进步都使得软件开发成为一件更容易的工作，但有时也使得该工作更加困难。设计方法、编程语言、操作系统、网络协议、分布式体系结构——所有这些都影响着我们编写软件的方式。不幸的是，所有这些技术导致了大量的软件兼容性问题，而加上了 Internet 的背景，这样的局面则愈演愈烈。

计算机行业一直试图用某些途径来解决这些不兼容问题。Java 从编程语言的角度解决了这个问题，使得应用程序只需编写一次就可应用于任何支持 Java 虚拟机 (Java Virtual Machine, JVM) 的平台。通过使用 Java 的远程方法调用 (Remote Method Invocation, RMI) 技术，应用程序在分布式环境中的操作更加容易了。但是，当涉及到 Java 和 RMI 时，为了保证互操作性，你必须确保连接的两端都运行着 Java。在受控环境中，这是一种可行的解决方案，但是 Internet 远非一个受控环境。

CORBA 和 DCOM (DCE/RPC) 从连网的角度解决了互操作性问题。拥有了共同的将信息编码成网络数据包的方式 (使用“网络数据表示”，即 Network Data Representation)，编程语言不再是一个问题。相反，用序列化 (serialization) 方法来保持兼容性就成了值得关注的问题。这通常意味着，要建立兼容组件，你必须使用某个特定的开发商的实现方案。尽管这是所有开发商都乐于见到的，但让 Internet 上的每个人都同意使用某一开发商的软件却是不可能的。

Web 服务提供了一个抽象层，这个抽象层隐藏了对开发人员而言似乎不容忽视的许多实现细节。以编程语言为例：程序员倾向于采用一种特定的语言，而不考虑其他 (或许同样具有竞争力) 的语言。有时候，这种选择是根据经验得来的，而有时候则纯粹是出于某种喜好，例如习惯于在代码中看到分号和花括号。撇开这种感受赖以存在的基础不谈，这种选择对开发人员来说是必须面对的，而且似乎永远也不能回避。诚然，.NET 框架在把不同的编程语言嫁接到共同的环境方面做了许多工作，但这只有在那些支持 .NET 的平台上才能够实现。Web 服务让这种工作更进一步——它打破了各种平台之间的障碍允许不同种类的系统彼此交互。

Internet 引入的技术都带有一种共性。互操作性是大多数成功的 Internet 技术背后的推动力量。没有互操作性，一种技术就不大可能被采用；不被采用，即使是最值得注意的技术最多也只能算作是一个没有价值的毕业设计项目。Internet 之所以如此成功，其原因在于日益受到欢迎的标准化运动。这场运动以采用 TCP/IP 作为 Internet 通信协议的基石为开端，以 HTTP 和 XML 被人们接受继续发展，直到最近采用 SOAP。

实际上，SOAP 还没有被视为一种标准——至少在 W3C 看来它还不是一个标准。但是，SOAP 被看作业界所接受的 (industry accepted)，从各开发团体对它表现出的极大兴趣就可以看出这一点。事实上，W3C 下辖的 XML 协议机构 (XML Protocol Activity) 现在还对 SOAP 进行考察。在这个论坛中，W3C 成员们有机会将这种技术定型为能够满足行业需要的技术。

当今的 Web 应用程序每分钟要处理成千上万的事务，以解决各种各样的问题。当我们在建立某种体系结构时，该结构中的应用程序能够与别的应用程序进行交互，事务处理的频率和工作量也将逐步增长。这将迫使我们用全新的眼光来看待 Internet，最终导

致构成网络的那些系统发生质的变化。对可靠性、可用性、性能以及类似属性的关注都将在这种新的开发模型下得到满意的答案。

解决方案的探索

对于分布式计算，我们要了解的还有很多，而 Web 服务将迫使我们拓展自己在 Internet 方面的知识。时机已经成熟，我们必须从标准的角度出发思考问题了，这个标准将允许所有类型的系统同时各显风采。

在 Web 服务背后，最重要的概念是：软件的功能可以被共享，而不必考虑隐藏在这些服务程序背后的实现细节。这个层次上的集成是通过松散耦合的 (loosely coupled) 服务程序来完成的，它们为建立下一代应用程序提供了一个灵活而反应迅速的结构。

谁应该阅读本书

本书适合那些对 Web 服务还不太了解的中高级开发人员阅读。它将向你介绍如何使用 .NET 框架和 Visual Studio .NET 来构建 Web 服务，还将帮助你理解围绕这种新的结构的一些问题和容易犯的错误。软件工程师、程序员和 Web 开发人员将在本书中找到许多概念，以便在创建此类系统时将其应用到他们的日常开发工作中。一如既往，我们的目标仍然是通过讨论、示例和源代码来提供一些实用的信息，节省你的开发时间。

本书中的示例大多数都含有 Visual C# 代码，偶尔也带有小的 Visual Basic 代码段。之所以选择 C# 是因为它与 Java 和 C++ 编程有着密切的关系，而后两者应该可以满足最广大的读者的要求。

本书主要讲述的是用 .NET 实现的 SOAP 协议。尽管在第 4 章 “.NET Web 服务与 SOAP” 中讲述了一些介绍性的 SOAP 知识，我们还是假定大多数读者已经对该协议十分熟悉，并且仅仅是在将 SOAP 应用于 .NET 应用程序时寻找一些新的切入点。

本书内容

本书分为 4 个部分。

第一部分，“Web 服务基础”，描述了什么是 Web 服务以及 .NET 结构是如何应用到这种新的范例中的。

- 第 1 章，“Web 服务基础知识”，对 Web 服务做出了一个简要的解释，并指出为什么它们是 Web 应用程序开发的未来。

- 第 2 章，“.NET 结构与 Web 服务组件”，带领你快速漫游了 .NET 结构。它还概括地介绍了公共语言运行时 (Common Language Runtime) 和一些具体的 Web 服务组件。

- 第 3 章，“Web 服务与 XML”，介绍了如何用一些新的 .NET XML 类将 .NET 结构与 XML 结合到一起。

- 第 4 章，“.NET Web 服务与 SOAP”，解释了如何使用 SOAP 为 Web 服务提供一致的序列化格式、故障模式和公共协议。

- 第 5 章，“Web 服务描述与发现”，详细地讲述了 Web 服务描述语言 (Web Service Description Language, WSDL) 的细节以及通用描述、发现和集成 (Universal Description, Discovery, and Integration, UDDI) 的概念。

第二部分，“实现 Web 服务”，向你介绍了 ASP.NET 和一些 Visual Studio .NET 工具。

- 第 6 章, “用 ASP.NET 编写 Web 服务”, 介绍了从服务器端的角度出发进行的开发。

- 第 7 章, “使用 .NET Web 服务”, 讲述了微软的最新开发环境中提供的一些最新的特性。

第三部分, “更高级的 Web 服务”, 讨论了一系列问题, 并指出了在遇到这些问题时要考虑的诸多事项。

- 第 8 章, “.NET Remoting”, 涉及了诸如怎样使用 SOAP 来实现有效载荷、加密和另外的编码格式问题。

- 第 9 章, “极端 Web 服务”, 通过介绍与 Web 服务相关的一些极端问题的基础知识来结束本书的讨论, 这些问题包括身份认证、授权、权限以及数字签名。

- 第 10 章, “.NET 与 Web 服务安全”, 概括了在保障 Web 服务的安全时可以采用的一些选项。

此外, 本书还在以下附录中提供了一些参考资料:

- 附录 A: “.NET Web 服务示例”;
- 附录 B: “使用 ATL Server 创建 Web 服务”;
- 附录 C: “XML 协议与 SOAP”;
- 附录 D: “.NET Web 服务资源”。

目 录

第一部分 Web 服务基础

第 1 章 Web 服务基础知识	3
1.1 什么是 Web 服务	3
1.1.1 穷人的 Web 服务	3
1.1.2 XML 消息	6
1.1.3 语法与语义	6
1.1.4 Web 服务相关术语	7
1.2 Web 服务的发展历程	8
1.2.1 Internet 浪潮	8
1.2.2 各种 Internet 标准	8
1.3 Web 服务的各种用途	12
1.3.1 Business-to-Business	13
1.3.2 向消费者展示功能	13
1.3.3 整合不同种类的系统	13
1.3.4 快速开发环境	14
1.4 Web 服务的特性	14
1.4.1 性能	14
1.4.2 简单性	16
1.4.3 安全性	16
1.4.4 可靠性和可用性	17
1.4.5 一致性	17
1.5 用 Visual Studio .NET 创建 Web 服务	17
1.5.1 创建服务	18
1.5.2 创建客户端	20
1.5.3 在网络上跟踪消息	26

1.6 接口设计技巧	27
1.6.1 以往的经验	28
1.6.2 什么是接口	28
1.6.3 使用 SOAP 对信息进行编码	29
1.6.4 接口的改版	29
1.6.5 接口的复杂性	31
1.7 总结	33
第 2 章 .NET 结构与 Web 服务组件	34
2.1 建立 .NET 的动机	34
2.1.1 COM 的好处与局限	34
2.1.2 要考虑的其他技术问题	35
2.1.3 一种更好的模型	36
2.2 .NET 框架	37
2.3 公共语言运行时	37
2.3.1 Microsoft 中间语言	37
2.3.2 实时编译	40
2.3.3 公共类型系统	40
2.3.4 程序集与托管代码	48
2.3.5 安全性	48
2.3.6 System 名称空间	48
2.4 Web 服务	52
2.4.1 发现	53
2.4.2 描述	53
2.4.3 协议	53
2.5 总结	53
第 3 章 Web 服务与 XML	54
3.1 作为线缆表现方式的 XML	55
3.1.1 XML 与松散耦合	56
3.1.2 XML 与互操作性	56
3.2 使用 XPath 查询 XML 元素	57
3.3 XML 基础	57
3.3.1 文档、元素和属性	58
3.3.2 实体引用与 CDATA	60
3.4 URI 与 XML 名称空间	61
3.4.1 URL 与 URN	61
3.4.2 XML 名称空间	62
3.5 XML 结构	63

3.5.1 理解 XML 结构	63
3.5.2 .NET Web 服务与 XML 结构	67
3.5.3 XPath 剖析	68
3.5.4 XPath 运算符	71
3.5.5 XPath 内部函数	72
3.6 用 XLink 识别 XML 元素	73
3.7 XML 转换	75
3.7.1 XSLT 剖析	76
3.7.2 XSL 模板	78
3.8 .NET 的 XML 结构	79
3.8.1 读取 XML 数据	80
3.8.2 写入 XML 数据	80
3.8.3 用.NET 导航 XML	84
3.8.4 用.NET “拖” XML 元素信息	84
3.8.5 .NET 与 XPath	85
3.8.6 .NET 与 XLink	88
3.8.7 .NET 与 XSL	89
3.9 总结	91
第 4 章 .NET Web 服务与 SOAP	92
4.1 为什么需要 SOAP	93
4.1.1 为什么要理解 SOAP	93
4.1.2 SOAP 的优点	94
4.2 SOAP XML 对象模型	95
4.3 SOAP 封套	95
4.4 SOAP 报头	97
4.5 SOAP 主体	100
4.5.1 SOAP 主体序列化相关术语	100
4.5.2 SOAP 主体属性	101
4.5.3 SOAP 远程方法序列化	103
4.5.4 简单数据类型的 SOAP 序列化	105
4.5.5 复合数据类型的 SOAP 序列化	109
4.5.6 SOAP 结构体序列化	109
4.5.7 SOAP 数组序列化	111
4.5.8 SOAP 故障	115
4.6 .NET SOAP 类	117
4.6.1 .NET SoapFormatter 类	117
4.6.2 .NET SOAP 框架类	120

4.7 总结	121
第 5 章 Web 服务描述与发现	123
5.1 Web 服务描述语言	123
5.1.1 抽象与具体	124
5.1.2 客户端的观点	135
5.2 通用描述、发现和集成	136
5.2.1 什么是 UDDI	136
5.2.2 UDDI 是怎样工作的	136
5.2.3 UDDI 与安全	137
5.2.4 tModel	137
5.2.5 查询模式	138
5.2.6 浏览	138
5.2.7 钻透	140
5.2.8 调用	141
5.2.9 发布	142
5.2.10 专用操作	143
5.3 总结	143
 <i>第二部分 实现 Web 服务</i>	
第 6 章 用 ASP.NET 编写 Web 服务	147
6.1 .NET 中的 Web 服务处理	147
6.1.1 ASP.NET Web 服务结构	148
6.1.2 .NET Remoting 与 .NET Web 服务之比较	149
6.2 Web 服务与 Visual Studio .NET	150
6.2.1 Visual Studio Web 服务项目	150
6.2.2 从“Hello World”开始	151
6.2.3 WebMethod 属性	155
6.3 控制 SOAP 串行化格式	156
6.3.1 SOAP 方法属性	156
6.3.2 数据整形	159
6.3.3 进一步定制 Web 服务 SOAP 数据包	162
6.4 添加 SOAP 报头	169
6.4.1 报头处理	170
6.4.2 指定 SOAP 报头的方向	171
6.4.3 附加的 SOAP 报头	172
6.5 添加一个 SOAP 扩展	172
6.5.1 扩展流处理	172

6.5.2 修改 XML	177
6.6 错误与 SOAP 故障	177
6.6.1 默认的.NET SOAP 故障处理	178
6.6.2 使用 SoapException 定制 SOAP 故障	178
6.7 Web 服务状态管理	178
6.8 调试与部署	181
6.8.1 调试	181
6.8.2 .NET Web 服务的部署	183
6.9 Web 服务与最佳惯例	186
6.10 总结	186
第 7 章 使用.NET Web 服务	188
7.1 Visual Studio .NET Web 服务支持	189
7.2 使用 Web 服务	190
7.2.1 创建 Web 引用	195
7.2.2 Web 服务配置文件	206
7.3 SOAP 报头	208
7.4 截取和修改 SOAP 数据包	210
7.5 再谈部署和调试	218
7.6 总结	218
 <i>第三部分 更高级的 Web 服务</i>	
第 8 章 .NET Remoting	223
8.1 .NET Remoting 结构	223
8.1.1 Remoting 边界	224
8.1.2 Remoting 对象模型	225
8.1.3 Remoting 通道	226
8.2 可远程使用的对象	226
8.2.1 可串行化对象	226
8.2.2 通过引用来调度对象	227
8.2.3 对象生命期	228
8.2.4 配置.NET remoting	229
8.3 .NET remoting 样例	230
8.4 总结	232
第 9 章 极端 Web 服务	233
9.1 嵌入式 XML	233
9.1.1 实体引用	234

9.1.2	CDATA 段	234
9.1.3	base64 编码	235
9.1.4	带格式的 XML 消息	238
9.1.5	带有附件的 SOAP 消息	239
9.2	SOAP 和附件	240
9.3	事务处理	242
9.3.1	事务处理权限标识语言	242
9.3.2	将事务处理应用于 Web 服务	243
9.4	调试与 Web 服务	244
9.5	Web 服务文档资料	246
9.5.1	概述	246
9.5.2	设计方法	246
9.5.3	API 摘要	248
9.5.4	API 参考	249
9.5.5	数据结构	249
9.5.6	错误信息参考	249
9.5.7	测试环境	250
9.5.8	文档样例	250
9.6	总结	252
第 10 章	.NET 与 Web 服务安全	253
10.1	安全术语与概念	253
10.2	应用程序级安全与系统级安全之比较	254
10.3	Web 服务与安全	255
10.4	Web 服务安全的范围	255
10.4.1	Intranet Web 服务安全选择	256
10.4.2	Internet Web 服务安全选择	260
10.5	.NET 安全	272
10.5.1	.NET 基于证据的安全	273
10.5.2	COM+安全	274
10.6	总结	275
第四部分 附录		
附录 A	.NET Web 服务示例	279
附录 B	使用 ATL Server 创建 Web 服务	291
附录 C	XML 协议与 SOAP	300
附录 D	.NET Web 服务资源	308

第一部分



Web 服务基础

第 1 章 Web 服务基础知识

第 2 章 .NET 结构与 Web 服务组件

第 3 章 Web 服务与 XML

第 4 章 .NET Web 服务与 SOAP

第 5 章 Web 服务描述与发现

第 1 章

Web 服务基础知识

本章内容：

- 什么是 Web 服务。
- Web 服务的发展历程。
- Web 服务的各种用途。
- Web 服务的特性。
- 用 Visual Studio .NET 创建 Web 服务。
- 接口设计技巧。

现在，在几乎所有的计算机行业杂志上都可以看到与 Web 服务有关的标题。在一片诸如“新的范例”的称道声中，我们是否真的见证了一种新技术的诞生呢？

不幸的是，答案并不明确——它或多或少地取决于你的看法。Web 服务可以用多种方式来使用，包括：

- 参与 business-to-business (B2B) 事务处理；
- 向客户展示软件的功能；
- 将不同种类的平台和编程语言整合在一起；
- 为产品开发提供一个经过简化的平台。

1.1 什么是 Web 服务

Web 服务可以指代在 Internet 上可供访问的任何功能，它们一般（但并非一定）在通信协议中使用一条或多条可扩展标识语言（eXtensible Markup Language, XML）消息。Web 服务使用操作概念来表示一条请求消息与零个以上的响应消息之间的关联。当这些操作组合起来满足某些特定的目的时，就形成了一个接口（interface）。

1.1.1 穷人的 Web 服务

Internet 上已经充斥着许多传统类型的 Web 服务，它们更为人所知的名称是 Web 页