



全面解析 REST 体系结构风格

全面解析 REST 体系结构风格

REST (Representational State Transfer) 是代表状态传输的缩写。它代表了分布式超媒体系统的体系结构风格，该风格是 Roy Field 在他的论文中定义的。它是一种针对网络应用的设计和开发方式，可以降低开发的复杂性，提高系统的可伸缩性。REST 提出了一些设计概念和准则： 1. 网络上的所有事物都被抽象为资源 (resource)； 2. 每个资源对应一个唯一的资源标识 (resource identifier)； 3. 通过通用的连接器接口 (generic connector interface) 对资源进行操作； 4. 对资源的各种操作不会改变资源标识； 5. 所有的操作都是无状态的 (stateless)。

什么是 REST?

REST (Representational State Transfer) 是代表状态传输的缩写。它代表了分布式超媒体系统的体系结构风格，该风格是 Roy Field 在他的论文中定义的。REST 在系统中加入一系列的限制，为系统确保了极大的可扩展性。

❖ 什么是 REST?

❖ REST: Web 服务设计的简易性

REST 的应用

REST 架构应用程序的创建意味着一种风格，或者说是一种思考问题的方式，而不是一个机械的规则应用，而根本的规则应该是 GET、POST、DELETE、PUT、OPTIONS 及 HEAD 这些 HTTP 方法都只能用于特定目的。

❖ 支持 REST 架构 Web 服务：在规范中的应用

- ❖ 支持 REST 架构 WEB 服务：在工具包的应用
- ❖ 使用 NetBeans IDE 发布 RESTful Web 服务

REST 与 SOA

创建 SOA 不是件容易的事。你需要重新定位应用开发部建立程序的方式，并起草一项全面的企业架构计划，而在这个过程中你不可能没有遇到一个高难度的问题。如果 SOA 很容易的话，那么我们就没有必要浪费精力谈论 SOA。然而，正是因为实施 SOA 的艰苦工作，才可以使后来 SOA 的应用变得顺畅。

- ❖ REST 与 SOA 的轻松生活
- ❖ REST 数据服务 完美体现 SOA 风格
- ❖ 独家专访：如何看待开源 ESB 和基于 REST 的 SOA？

REST 与 SOAP 的比较分析

与其将 REST 和 SOAP 进行比较，还不如将 REST 将和 SOA 进行比较。REST 是一个体系结构类型。而 SOAP 是一个消息传递协议。REST 应用程序可能使用 SOAP 协议，但是在大多数的情况下，REST 应用程序只是简单地使用基于 HTTP 的 XML 传递消息。使用 SOAP 代替简单的基于 HTTP 的 XML 的好处在于，SOAP 提供了一种机制，这种机制可以从应用程序的有效负载中清晰地分离出基础功能。

- ❖ REST 与 SOAP 之间的比较分析
- ❖ REST 与 SOAP 之比较——SOAP 篇
- ❖ 何时是使用 SOAP 或者 REST 的适当时机？

什么是 REST



Anne Thomas Manes, Burton Group 的研究部总监, 是 Web 服务领域知名的技术专家。她曾在 2001 年被《网络世界》评为“网络最具影响力的 50 位人物”之一; 2002 年被《企业系统期刊》评为“100 名杰出 IT 领导者”之一。加入 Burton Group 之前, Manes 曾是 Bowlight 公司的创建者和 CEO, 专注软件行业分析和咨询。她具有二十四年的行业经验, 曾担任 Systinet 公司(从事 Web 服务基础设施)的首席技术官, 革新 Sun 的 Web 服务战略, 并在 Patricia Seybold Group 担任分析师。作为分布式计算技术领域的知名专家, Manes 还参与了 W3C, OASIS, JCP, UDDI 和 WS-I. 的标准开发。她还是业界最权威出版《Web 服务期刊》编辑专家组的会员之一; 多次参加会议演讲, 发表无数文章, 并著有《Web Services: A Manager's Guide》一书, 由 Addison Wesley 出版。

问: 什么是 REST (Representational State Transfer)? 是否有合适的相关资料介绍呢?

答: 关于 REST 的大量资源, 可以访问这个网站: <http://rest.blueoxen.net/cgi-bin/wiki.pl>。

REST 是代表状态传输的缩写。它代表了分布式超媒体系统的体系结构风格, 该风格是 Roy Field 在他的论文中定义的。REST 在系统中加入一系列的限制, 为系统确保了极大的可扩展性。虽然 REST 定义了大量的重要体系结构限制(例如将用户接口和服务器实现的清晰分离, 无状态, 以及缓存能力)。REST 关键的特有要求是统一接口的要求。REST 要求使用通用的, 统一的以及非特殊的接口。它不允许应用程序专用接口。在 Web 世界中, REST 接口是由 HTTP 接口定义的, 支持基本的应用程序方法: GET, POST, PUT 和 DELETE。(除了这基础的四个方法外, HTTP 没有其他的方法。)这种统一接口限制, 跟面向服务体

协结构风格（SOA）相反，后者为每种类型的服务定义了不同的接口（例如 WSDL 接口类型）。

(作者: Anne Thomas Manes 来源: TT 中国)

REST: Web 服务设计的简易性

Web 服务的请求者和提供者之间的连接给双方都带来了大量的工作，其中包括像同意实现业务功能、认同技术合同细节这样的事情，当然还包括从服务到应用程序的宏伟计划的整合。但是，如果过于经常使用标准化的 SOAP/WSDL 方法，这只会使这种情况变得极度复杂。

还有一种办法，就是部署被称为 REST 的 Web 服务兼容架构，它代表“表述性状态转移”（Representational State Transfer）。REST 是由 Roy Fielding 于 2000 年在其博士论文中所命名的一种技术。这种技术与行业中的主要规范对立竞争，虽然他是否有创造这一技术的先见之明还值得商榷，但是自从那时起，REST 的简称在网络服务设计的舞台上，已经扮演了主要角色。

是什么使 REST 不同于标准化的 SOAP/WSDL？是 REST 的简易性。它抛弃了后者必需的一些冗繁的要求，而这些要求在实现 Web 服务的基本功能时，在许多情况下都被证明是不必要的。

为了更好的了解 REST 的理念，让我们选取 SOAP/WSDL 设计中一个典型的过程：

- 1 - 提供者在现有的应用程序上创建并实现 Web 服务接口。
- 2 - 提供者创建 WSDL 契约，从而将 Web 服务细节提供给潜在的用户。
- 3 - 用户获取 WSDL 契约以便利用该契约。
- 4 - 用户在特定的平台——Java, C#, PHP, Perl——上执行 WSDL，并且创建通讯程序。

这个过程中，最大的困难在用户身上。他不仅需要按计划提取 Web 服务的有效载荷——例如 SOAP 消息——利用其内嵌于程序做一些有用的事情，而且也需要按计划通过 WSDL 发出起始服务请求，以取得有效载荷。如果你像许多人一样，在不同的情况下都经历了这个过程，也许你会认为这是处理 Web 服务时已存在的一个事实。但是，这个过程往往能证明是不必要经历的，所以我们不如退一步，探寻 Web 服务的实质。

许多 IT 部门的待议事项上都有 Web 服务，其主要原因很简单：为了在不同系统之间实现互操作性，例如一个与 Java 应用程序通信的 SAP ERP，或一个连接到 a.NET 新闻服务的 PHP 网站。了解了这一点，让我们分析一个非常普遍的服务——Atom 和 RSS 新闻提要，它完全符合这一解释，而且其核心也是 REST。

新闻摘要，最初只是在博客中的一个构想，而现在几乎在每一个主流网站上都有应用。它允许提供者发布网站更新，而且允许用户按需获取这些更新。在后台，提供者的站点可以建在 Java、PHP 或 Perl 中，同时提供一个 RSS 格式或 Atom 格式的 XML 有效载荷。随后，这些更新就可以被任何一种语言写成的程序所使用。

如果你在网上迅速一瞥，就可以看到新闻摘要充斥在每个角落。有许多应用程序（用户）都从网上收集 RSS 数据源，从而提供丰富的内容，而这些数据源都来自第三方服务。这个过程很简单，而且相比于 RSS/Atom 服务仅提供阅读功能来说，它实际上更为灵活。由于 REST 过程是在 HTTP 环境中执行，其所有的参与人员——get、post、put 及 delete 的操作人员——都即将实现需要实现的任何商业功能。

如果你还是对 REST 理念的含义有点半信半疑，认为它更学术化，或者认为 RSS 和 Atom 新闻信息是一个孤立的情况，那么让我们再看看一些使用 REST 的生产型应用程序。

REST API 允许开发人员录入众多的信息来源，从而将相同的数据整合入第三方应用程序。雅虎的 Web 服务产品也是基于 REST 建立，以至于他们目前甚至不提供 SOAP 支持。雅虎只使用 REST。

看了上述几个例子，你应该能感觉到 REST 开发的 Web 服务背后有强大的社会支持，但是别急着下结论。就像所有的技术一样，REST 也有其缺点。如果在 REST Web 服务中有一个共同论题的话，那就是它们处理的数据都具有的非必要性。

通过应用程序传输较为敏感的数据，如财务数据或采购信息，需要使用更先进的特性，比如业务交易，故障恢复功能和服务质量。而考虑到 REST 的简易性，这项工作相对来说就比较复杂。但这并不是说这一情况不能通过 REST 实现，而是说根据这类企业的需求，SOAP/WSDL 方法更为有利，更适合他们。

SOAP，通过其一系列的配套规范，如 WS-政策，WS-安全性，WS-可靠性和 WS-业务交易管理等，清晰详细的解释了如何从问题刚出现时就解决这些问题；而在 REST 中，这些问题被整合进程序中，变得更为复杂。

虽然 REST 不是适用于每一种情况，但是它可以使 Web 服务的创建和交流工作简单很多。鉴于其在主要门户网站中广泛的应用，以及其简洁实用的方法，在探索未来 Web 服务设计的事业中，这是一个不得不采取的方法。

(作者: Daniel Rubio 译者: Eric 来源: TT 中国)

支持 REST 架构 Web 服务：在规范中的应用

REST 是英文 Representational State Transfer 的缩写，中文翻译为“表述性状态转移”，它是由 Roy Fielding 在他 2000 年的博士论文中提出的一个术语。毫无疑问，这篇论文是近几年来最有影响力的学术研究成果之一。Fielding 的论文题目“架构风格与基于网络的软件架构设计”（“Architectural Styles and the Design of Network-based Software Architectures”）强调说，REST 架构应用程序的创建意味着一种风格，或者说是一种思考问题的方式，而不是一个机械的规则应用，而根本的规则应该是 GET、POST、DELETE、PUT、OPTIONS 及 HEAD 这些 HTTP 方法都只能用于特定目的。

对资源的思考

Web 地址的概念是“对抽象或物理资源的标识”（RFC 2396, 1998），它是万维网概念的核心。而由于研究人员已经探索出了多种描述方法，所以这个概念已变得相当模糊。那些曾开发 SOAP、网络服务编程的远程过程调用（RPC）风格的一些人表示，他们采用 REST 风格时，有时候会碰到一些困难。现在，期望 URL 直接指向进程或是服务器上的离散文件已成为一种趋势。为了说明 REST 观点的特殊之处，我们引述 Fielding 的论文如下：

资源不是被存储的对象，也不是服务器处理存储对象的一种机制。资源是一个概念性映射——服务器收到（标识映射）的标识符，并将其应用于当前的映射过程（通常是特殊深度树遍历与/或哈希表的组合），找到并执行当前负责的处理程序，然后该处理程序执行将选择适当的基于请求内容的动作+响应。所有这些执行过程-具体的事务题都隐藏在网络界面下；一般客户不能利用这些特性，只有通过网络界面的认可才可以。

SOAP 和 REST 的要求对比

网络服务的核心问题是解释请求，并按规定路线将请求发送到正确的进程中，并获得返回的期望表示法。基于 SOAP 的服务不仅要查看请求报头，而且也要解析 XML 格式的请求主体，并返回 XML 格式的文本。

REST 架构服务只需要检查请求报头和 URL，确定处理该请求的进程和返回的资源表达法。重要的是，要记住一个特定的资源可以有好几种不同的表达法。举例来说，日历任命可能会以文本、HTML、XML、PDF 或图像的格式返回。

REST 研究产业的日益发展，引起了网络服务相关工具——例如标准、语言图书馆和集成驱动电子设备（IDE）——产品开发商对新思路的探索。我们看到，他们将 REST 纳入到现有的工具中的尝试，也看到他们对全新的方法的采用。首先，让我们看看 REST 在最近规范中的应用。

WSDL 2.0 规范中的 REST 应用

在认定开发商越来越多的从 SOAP 开发转向 REST 风格开发后，W3C 工作组在最近发表的 WSDL 2.0 规范中，纳入了他们所称的“HTTP 支持”。这个版本制作了许多年，是一项对 1.0 版本的重大修订，采用重组降低了其复杂性。在理论上，一个 WSDL 2.0 文件可以描述 SOAP 程序，也可以描述 REST 程序。以前用来从 WSDL 文件创建网络服务的工具，将需要重大修改以适应 WSDL 2.0。

WADL：一种新方法

开放源代码项目之一是 GlassFish 公司执行的 WADL——网络应用描述语言。WADL 文件以 XML 的格式对 REST 风格的网络服务做出了描述，称其适合创造 Java 客户端的计算机进程。同时为获取资源和路径，也提供了对这两项的柔性描述。

JAX-RS REST 架构网络服务中的 Java API

这个规范——按照 Java Community Process 公司提供的方法正在向 JSR 311 发展——还处于“编辑草稿”的阶段，其最新的版本发表于 2007 年 12 月 4 日。目前该规范仍然没有完成，还有许多悬而未决的问题。其目的是，在 POJO（普通 Java 对象）类中使用 Java 注解，从而使 POJO 类能以 REST 架构 Web 资源显现出来。使用 Java 注解当然需要 Java 1.5 版本或更新的版本，这对大多数用户来数，应该不成问题。Jax-RS 不能提供服

务发现或是协助客户端编程，但却能保证对现有的 servlet 容器和 Jax-WS 工具组的兼容。

纯文本规范

在陷入 XML 服务描述语言的麻烦之前，让我们转过身往回走一点，去考虑那些简单的方法。雅虎公司关于创建 REST 请求的说明文件，只是满载例子的文本性描述。毫无疑问，各种搜索服务有其各自的指令集合，但反过来，它们都附有简单的文字解释。REST 纯粹主义者会说，雅虎的请求并不是百分之百的 REST，因为他们使用了 URL 查询格式化，包括一个指定期望表示法的参数，而不是一个请求报头。但是不管怎样，雅虎的请求也可以完成工作，而且雅虎公司为不同的描述语言提供了不同的范例以及不同的软件开发工具包（SDK）。

程序员的看法

现在，如果我必须写一个简单的 REST 架构 Web 服务的话，我就会用我熟悉的 Java servlet 编写请求解释。如果要对该服务进行描述，使其他人建立客户群的话，我会写一个浅显的文字解释，并借助 WADL 描述进行补充。不过，如果我检查到最新的编程工具包，在下一篇文章中，也许我会改变主意。

(作者: William Brogden 译者: Eric 来源: TT 中国)

支持 REST 架构 WEB 服务：在工具包的应用

在上一篇文章中，我描述了规范的创建人如何处理业内对 REST 架构 Web 服务（RESTful Web services）引发的兴趣。我发现新方法和现有标准的修改中都包含 REST。

而且，我发现同样的事情也发生在软件工具箱的开发商身上，这些软件工具箱广泛应用于分布式系统。换句话说，在现有的项目中添加 REST 界面，是为了满足开发商的利益，而且少数全新的工具中也添加了 REST 界面。我非常惊奇地发现，众所周知的开源 Java 消息服务（JMS）的最新版本，也增加了一些 REST 的相关功能。

Java 消息服务

JMS 是面向消息中间件（MOM）的一个例子，是在松耦合分布式系统中执行数据交换的最简单的方式之一。JMS API 因为其基本思想十分简单，自 2001 年发布以来几乎没有什么变化。JMS 服务器将“生产者”发出的信息加以管理，并将其储存，以备“消费者”随后对其进行异步阅读。服务器可以持续提供这种服务，所以并没有要求生产者和消费者必须同时连接到网络。

每条消息都是独立而完备的，都包括路由信息以及一个消息主体。相比于分布式系统的 Javaspace 模式——消息内容决定消息命运，JMS 服务器只注重路由信息，而忽略了消息主体。

基本的通讯模式有两种，一种是发布-订阅，一种是点对点的消息队列。所有消息订阅用户都可以阅读发布的消息，甚至是某一特定的话题。比方说某公司中由多个客户抄的一个价格表。消息订阅用户不需要在信息发布时处于连接状态，只要他们已经注册了对该话题的持久连接。

阅读发送到点对点消息队列的消息时需要占用该消息的使用权。可能会有多个用户连接到同一个队列，但是只有一个用户能得到所有消息，因此一个队列在计算网格中的分配工作只能是理想化的。因为消息队列和发布的话题都需要通过名字寻址，因此 REST 架构的资源寻址很容易适合 JMS 寻址方案。

Apache ActiveMQ JMS 服务器

我免费下载并安装了一个最新的 ActiveMQ 构件，这是一个非常完整的下载包。为了用提供的例子对其检验，你只需要一个 1.5 版或更新版本的 Java SDK，以及一个 Ant 构件工具包。Jetty servlet 容器也包含在下载包中，因此你不必安装任何 HTTP 服务器。ActiveMQ 工具包已经变得非常强大，相比简单的 JMS 支持，它提供了更多的功能。举例来说，它支持 JavaScript 脚本的专门服务，而且支持 AJAX 请求的处理。

最让人感兴趣的是 REST 架构 Web 服务的例子。该程序包括：

- 从一个发布-订阅的股票价格主题读取数据，创建一个股票投资展示。
- 使用 POST 将信息发送至发布-订阅话题或是点对点队列。
- 使用 GET 从队列中阅读信息。

ActiveMQ 通过 GET 从队列阅读信息的方式，说明了一个有趣的问题。在严格的 REST 架构中，一个 GET 不应当改变资源，但 JMS 要求读取点到点队列必须删除该消息。所以我猜想，将 REST 加装到其他框架中，也可能会碰到类似的问题。下面是开发商不得不说的一些话：

注意，严格的 REST 架构要求 GET 是一个只读操作。因此，严格来说，我们不应该使用 GET 允许用户利用消息。尽管我们允许了 GET 的使用，这是因为它稍微简化了 HTTP /DHTML /Ajax 的一体化。

作为典型的开源项目，与下载包一起提供的说明文件比较薄。不过，可以下载含有典型的 Javadocs 注释的源代码，而且该项目网站也提供有丰富的资源。

Restlet——一种新方法

关于 REST 架构 WEB 服务的全新方法，我们找到了一个例子，这就是 Restlet 项目。Restlet 是 Noelios Consulting 公司的标志性产品，它是 API 的最初模板。这个开放源代码项目已经获得开放源代码促进会（Open Source Initiative）的“通用开发和发布许可”（Common Development and Distribution License）。其首次公开发布是在 2005 年 11 月，目前版本为 1.0.7，对 1.1 版本有较大的修改。

Restlet 开发商认为只有完全取代了 servlet API，才可以充分利用 REST 的价值。例如，servlet API 每个请求只能使用一个线程的假设，避免了 Java 的“NIO”非阻塞输入/输出类的使用，同时限制了其可扩展性。而 NIO 类的使用可以允许计算元件之间使用更为灵活的连接。

Restlet 例子在惯于使用 servlet API 的开发商看来，可能乍看上去会觉得很不可思议。举例来说，他们认为没有什么可以与 servlet Session 相比。令人感到幸运的是，最近由 Leonard Richardson 和 Sam Ruby 合编的一本名叫“REST 架构 WEB 服务”的书问世，其中包含了很多使用 Restlet 架构实现服务的例子。

可下载的 Restlet 共享包包括实现 Restlet API 的 Java 类，以及一个被称为“Noelios Restlet 引擎”的参考实现。这就建立了一个完整的 Web 服务器，提供了基本静态档案的高效服务。不过，如果你已经决心使用现有的 servlet 引擎，比如 Tomcat，就必须提供一个运行 Restlet 程序的适配器。而且，还应当提供能连接 ApacheWeb 服务器或微软 IIS 的适配器。

下一步是什么？

我会寻找其他的例子，无论是旧框架中应用 REST 新特性的例子，还是全新方法的例子。也许我会找到足够的范例，可以写出第三部分文章。

(作者: William Brodgen 译者: Eric 来源: TT 中国)

使用 NetBeans IDE 发布 RESTful Web 服务

集成式开发环境主要由两个开放源软件占主导地位软件所支配，即 NetBeans 和 Eclipse。今天我们主要看其中一个软件 NetBeans 是如何为 RESTful Web 服务 services 提供支持的。

NetBeans 是 Sun 公司购买的一家商业 IDE 公司所开发的软件，2000 年作为开源软件发布。该软件由 JAVA 语言编写，具有非常广泛的用户群。同 Eclipse 一样，NetBeans 由许多加载模块组成，可以支持多种语言和设计技术。NetBeans 为许多 Sun 公司最钟爱的技术提供技术支持，包括 Glassfish 的开源 Java EE5 应用服务器。虽然服务器是 Glassfish 开发社团的核心，许多项目是独立存在的。有些项目肯定最前沿的 Web 服务技术。其中一项包括 JAX-RS 的实施，即针对 RESTful Web 服务的 Java API 或 JSR 311，称为 Jersey。令我高兴的是，我发现 Jersey 的实施取得显著进步，它已经增加为 2007 年 12 月发布的最新 NetBeans 6.0 版本的插件。

安装 NetBeans

我下载并安装了 NetBeans 6.0。尽管里面的插件尽量少，但还是一个十分巨大的数据包。插件管理器功能提供了一套很好的系统，以保持随时更新并获取所需插件。安装包括 Glassfish 服务器，并能检测系统中的其他服务器。尽管 NetBeans 可以和 Java JDK 5 一同工作，不过要是在安装 NetBeans 前将其升级到 JDK 6，就更加理想了。

添加 RESTful Web 服务支持

插件管理员通过连接中央储存器完成 NetBeans 升级并添加新的功能。我加载了一个叫做“RESTful Web 服务”的插件，更新了其他几个模块。该插件是和 Jersey 0.5.1 相对应的测试版本。所以目前还处于开发阶段。

我按照 Jersey（援引下面的网络资源）的范例在 NetBeans 创建了一个新的 RESTful Web 服务项目。主要是填写 URIs, 中出现的名称、JAVA 软件包名和类名。这项服务的目的是检测输入词，根据 Metaphone 语音编码算法，再返回一串语音相近的词。我规定通过 URI 路径进入该资源返回字符串为“lookup(查看)”。

JAX-RS 规范总是用“资源”这个词，强调对服务的 REST 思考方式，而不是截然不同的 SOAP、JAX-RPC 对服务的思维方式，作为激活命名程序。IDE 按照基于 RESTful 模式创立的 Java 编码使用该惯例为“LookupResource”（查找资源）类建骨架代码，如下所示。每一项请求都会产生一系列新的 LookupResource 实例。

“@Path”注释告诉 Jersey 运行时 servlet 将 URL 中的“lookup”映射到改服务类。

“@HttpContext”注释使运行时 servlet 自动“注入”UriInfo 对象引导。当一项 LookupResource 的实例生成时，该对象引导就具有所有的请求 URI 信息。

```
@Path("lookup")
```

```
public class LookupResource {
```

```
    @HttpContext
```

```
    private UriInfo context;
```

在下面的 java 编码中，我留下了 NetBeans 创建骨架代码时的注解。“@GET”和“@ProduceMime”注解说明该方法中的框架能满足 HTTP GET 的请求，返回一个格式串 HTML。当然，该方法编写的内部细节取决于程序设计师。这个例子展示了第一步，证实了查询参数值可以从 UriInfo 目标中抽取。

```
/**
```

```
 * Retrieves representation of an instance of
```

```
 com.wbrogden.phonetic.LookupResource
```

```
 * @return an instance of java.lang.String
```

```
 */
```

```
@GET
```

```
@ProduceMime("text/html")
```

```
public String getHtml() {
```

```
    // the context contains all query information
```



```
String word = context.getQueryParameters().getFirst("match") ;  
return "Match String " + word ;  
}
```

“@ProduceMime” 注释用来查找与请求中最匹配的资源提供者的类。使用 NetBeans 的“运行”命令，启动我的 Tomcat 服务器。TOMCAT 启动信息显示，当架构 servlet 启动时，Tomcat 搜索执行类。传送我的浏览器到下面的 URI

<http://localhost:8080/MetaPhone1/resources/lookup?match=word>

产生了预料之中的文本“匹配字符串”，证实框架能够确定我的服务级别，创建一个实例。这叫做 getHtml() 法。当 URI 运用查询格式是，进一步的实验表明除了通过查询格式，我无法让系统认定运用 REST 分层模式的 URI。下面的 URI 将传送“一些词”文本到查看资源，但 Tomcat 报道出错。

<http://localhost:8080/MetaPhone1/resources/lookup/someword>

在下载 Jersey 独立版本中，还有许多为分层模式提供支持的例子，所以看样子我还得弄清如何能让 NetBeans 插件做同样的工作。

Netbeans 为程序设计和调试提供极大的支持，例如，在编辑 getHtml() 方法代码，编辑器向我展示了所有“上下文”对象支持的方法。与 Web 相关联的程序设计显著特性就是能够记录完整的 HTTP 请求和回应文本。

(作者: William Brodgen 译者: 杨君 来源: TT 中国)

REST 数据服务 完美体现 SOA 风格

因为 REST 风格不同于面向对象的编程风格，它是通过统一资源标志符 (URIs) 对所有事物命名，因此它们可以被检索，而且它们非常适合创制数据服务的应用程序，他说到。他设计了 Burton 的案例，在此案例中，他将 REST 的使用作为 Webcast 的补充。这个分析公司上个星期在 Webcast 中发表了 SOA 原则如何应用到数据存取和管理。

Lacey 一开始就解释了 REST 是什么以及 REST 如何融入到数据服务的世界。“它是一种风格，”他说到：“它不是一个技术，它的分布式计算环境的风格，作为其关键的抽象概念，促进了资源的发展。于是一个资源就变成了简单的东西，它可以被命名。可以被命名意味着它能够通过 URI 的方式被提供。”

从业务数据的角度，已经被命名了的资源可以是任何显而易见的事物，比如一个雇员，或者某种股票或者项目文件，他说到。

“但是它也能够公平地表现一些不明显的事物，比如说今天购买的第三项东西，或者说 2007 年所有的销售量，或者仅仅第二季度，或者仅仅是昨天，甚至仅仅是昨天 3 点的等等。”Lacey 解释到说。“有一种无限供给的资源定位符，你可以使用资源定位符来命名无数的资源。总之，资源是简单的数据，并且当你谈论数据服务的时候资源变得很有趣。”

他说原先最早描述 REST 的文章指出，“一个系统架构的数据元素的性质和状态是 REST 中一个关键的环节”，不像一个分布式对象系统。

“换言之，在 REST 里面，数据是第一级的公民，”Lacey 说到，“不像其他的系统，比如 SOAP 或者 DCOM 或者 CORBA，重点放在过程上，在 REST 里面，数据才是焦点。”这就是使得 REST 非常独特地适合于数据服务之处，Burton 的顾问说到。

为 ROA 做好准备

虽然用行话来说，承认所有人都“升级到这里”，然而 Lacey 又提供了另外一个资源导向型架构 (ROA)，此资源导向型架构是为数据服务使用 REST。

“如果 REST 是一种风格，那么以资源为导向的架构就是这种风格的一种实现，就好像如果面向对象编程是一种风格，那么 Java 就是一种实现一样。” Lacey 说到，“资源导向型架构就是是应用到真实世界的 REST。”

在这个世界上，他解释到说，所有的每条信息都拥有自己的 URI。HTTP 是用来在网络上提供数据和服务的可用性的。资源可以通过统一的接口得到。并且所有对这些资源的操作可以归结为四个非常常见的命令，获得，插入，删除，提交。

“在一个资源导向型架构中的最后一个关键的组成部分是，一切的资源都应该包含连到其它资源的链接。” Lacey 说到，“换句话说，我们提供新的发现。我们创造了一个 Web，因此当我得到一个雇员的时候，它应该提供此链接给他的经理。它应该能够提供链接到他的部门。因此我可以通过我的资源搜索到并且得到我想要的信息。”

他说到，使用 REST 作为数据服务的概念目前正在各种平台上审定中，这些平台包括 Astoria，微软实验室早期的 beta，以及一个开源的叫做 SnapLogic 的平台。他说到，开发者可以选择使用成熟的开发框架，容易得到的有 Restlet——他认为 Restlet 是继 Ruby on Rails 之后最为成熟的。对于 Python 程序员来说有 Django 以及 CherryPy 可供选择。Beta 开发项目也会有帮助，Lacey 说到，此项目是 IBM 公司的 Zero 项目和微软公司的 .NET 3.5，这些项目中都将包括更多的 REST。在 Java 的世界里面，存在 JAX-RS，它将为 Java 提供 REST 的 API，他谈到。

“所有这些现在都可以使用了，” Lacey 说到。“现在成熟的产品已经能够用来方便地构建数据服务了。”

(作者: Rich Seeley 来源: TT 中国)

REST 与 SOA 的轻松生活

关于面向服务架构，我想向用户提一个问题，“它有没有使你的生活变得更轻松？”显然，创建 SOA 不是件容易的事。你需要重新定位应用开发部建立程序的方式，并起草一项全面的企业架构计划，而在这个过程中你不可能没有遇到一个高难度的问题。如果 SOA 很容易的话，那么我们就没有必要浪费精力谈论 SOA。然而，正是因为实施 SOA 的艰苦工作，才可以使后来 SOA 的应用变得顺畅。

SOA 应该能减少编码工作，缩短项目周期，使商业与 IT 更好的融合。作为回报，SOA 应具有某种简易性。

在这十年中的前几年里，服务导向开始重新占据主导地位，虽然这时没有人想到表述性状态传输（REST）会成为 SOA 中的一部分，但是 REST 的确在 SOA 概念中植入了 SOA 应以简易性为中心的理念。首先，也是最重要的，REST 是简单的。它利用的是 HTTP，工作量极小。特别是最近对 REST 兴趣颇浓的伯顿集团，一开始称 REST 为“SOA 的下一件大事”，而随后又注意到其非常适用于数据服务，因为 REST 重视的是数据，而不是过程。REST，从理论上讲，可以帮助实现极少的编码工作以及实现快速组装。如果应用程序失败而转向使用组件服务的话，那么，REST 相比于占有更多比例的编码方式，应该比较容易卸载。当然，有一个潜在的问题是它一直仍未实现企业级的应用，但是这一情况即将被改变。

IBM 取消了其基于 REST 的孵化器，零工程计划，而 Mulesource 聘请了 Xfire 的设计者 Dan Diephouse，统筹 REST 安装工作。这一周将在我们的新闻报道中介绍微软引入程序设计舞台中的一个新简称，MEST，数据交换状态转移（Messaging Exchange State Transfer）。它与 REST 相关，而且你可以相信它会更多的遵循 REST 模式。预计市场将会有大量的人员使用基于 REST 的计划，目的就是为了使您的生活更简单。这一实现方式实在有趣。

(作者: Michael Meehan 译者: Eric 来源: TT 中国)

独家专访：如何看待开源 ESB 和基于 REST 的 SOA？

开源 ESB（企业服务总线）的前景如何？是否有闭源 ESB 的应用空间或者能否采取某种开闭源混合的方式？在这次采访 Paul Fremantle 的第一部分中，这位 WS02 公司的共同创始人兼副总裁论述了这些问题，并介绍了 WS02 公司 ESB 产品的核心——Apache Synapse 开源 ESB。在合作建立 WS02 公司、开发基于 Web 服务标准的开源产品之前，Fremantle 曾担任 IBM 的高级技术人员，创建了 Web Services Gateway，而且带领团队开发并组装成 WebSphere 应用服务器的一部分。他作为团队的成员，也为 WebSphere Application Server 6. 开发了综合服务总线技术。目前，他是 OASIS Web 服务可靠交换技术委员会（WS-RX）的主席之一，该委员会研究使用 SOAP 进行可靠信息交换的标准。他第一次参与开源的研究可追溯到最初 Apache 的 SOAP 项目。Fremantle 获得过牛津大学的数学和哲学硕士，以及计算技术的理科硕士。



新推出的 WS02 Enterprise Service Bus 和你正着手进行的 Apache Synapse 项目有什么联系？

Paul Fremantle: Apache Synapse 是核心运行的动力，而且如果我们想提高核心运行能力，需要执行的代码都在 Apache Synapse 中。我们并非是贬抑其他的核心代码，但根本上，核心运行是以 Apache 项目为基础。

那么 WS02 公司的价值观是什么？

Fremantle: 我们为 ESB 提供支持，无论其用于高质量的商业培训、支撑还是服务。我们有一个图形用户界面。这是一个完全基于网络的用户界面，允许用户对基础位置的 Synapse 进行配置、监控和管理。

这是一个基于 Ajax 的 Web 界面吗？

Fremantle: 是的, 它是一个基于 Ajax 的 Web 界面。它的作用之一是公开所有的 API 管理及服务, 因此你可以将其与其他界面区别出来。

这么说来, 这是一个将管理内置的 ESB?

Fremantle: 正是, 但它的源程序是完全开放的, 包括管理控制程序也是。

还有没有其他的特点?

Fremantle: 我们有两样法宝。第一, 我们在常数存储器中编写信息, 而不是套用存储库中的大型信息树或信息模型。不过在某些情况下, 也必须在存储库中建立信息模型。信息不可能都被流化, 但要是可以流化的, 我们就可以做到。第二, 我们有一个完全无阻碍的运输模式。因此, 我们可以处理大量的通信连接, 而不会用光所有线程或是受到阻碍。我们非常重视提高运行能力时运行的稳定性, 同时也注重通过采用某种清洁、简单的模式提高其简易性。

一些厂商试图提供开源 ESB 和所谓的“闭源” ESB。你怎样看待?

Fremantle: 当我在 IBM 工作的时候, 我接触过开源和闭源共存的情况。我发现客户总是很难分辨两者之间哪一个更好, 特别是在过去几年中, 开源各项性能的品质不断上升。这是为什么我们没有推出企业版、标准版、免费版和共享版中任何一个版本的一个原因。我们只生产简单的开源产品, 客户可以直接购买。我们倒认为这对客户来说更为简便。

将来会不会有闭源 ESBs 的应用空间?

Fremantle: 我认为, 有一些产品会非常适合用闭源 ESB。举例来说, 有一些金融机构有超高的通信要求性能。如果每秒需要处理上百万条信息, 就需要非常复杂的高度调试软件产品。而这个产品的市场可能是 30 到 100 个客户, 也就是说它不是一个开放的市场。所以, 如果我要生产这么一个产品的话, 我不会使用开源。

但是另一方面, ESB 正成为解决问题的不二之选。即使是小公司也看到了 ESB 的好处。因此, 我认为开源正在完全接手这类市场。所以, 既然两三个高质量的开源产品能够很好的解决问题, 而且有更低的获取成本, 为什么要使用专利产品, 限定于某一个特定的供应商呢?

在过去的一年里, 应用于 SOA 的开源软件是否取得了进步?

Fremantle: 是的。我认为我们的第一组软件很稳定。我们在 2007 年已经为 SOA 提供了一个更为坚实的平台。现在我们正在同更大的机构洽谈, 比如说财富 500 强公司。他们表示正在认真考虑将开源用于 SOA。

除了 ESB, 你觉得将开源应用于 SOA 方面, 还有哪些新的项目或新技术特别能引起你的兴趣?

Fremantle: 我们最近刚启动了基于 REST 的 SOA 注册项目。现在有开源的 UDDI 项目和基于 ebXML 的项目, 但是仍然有很多人买很昂贵的专利产品。我发现相对于我们的那个项目, UDDI 和 ebXML 两者都是过于沉重和复杂的解决方法。因此我们回头去找最初的原理, 等我们再回来查看的时候, 我们认识到, 面对资料库/存储库时, Web 资源才是最根本、最重要的。这才是真正的管理资源。所以, 这个想法引领我们采用了 REST 模式。这就是为什么我们要建设一个完全基于 REST 的资料库/存储库。

(作者: Rich Seeley 译者: Eric 来源: TT 中国)

REST 与 SOAP 之间的比较分析



Anne Thomas Manes, Burton Group 的研究部总监, 是 Web 服务领域知名的技术专家。她曾在 2001 年被《网络世界》评为“网络最具影响力的 50 位人物”之一; 2002 年被《企业系统期刊》评为“100 名杰出 IT 领导者”之一。加入 Burton Group 之前, Manes 曾是 Bowlight 公司的创建者和 CEO, 专注软件行业分析和咨询。她具有二十四年的行业经验, 曾担任 Systinet 公司(从事 Web 服务基础设施)的首席技术官, 革新 Sun 的 Web 服务战略, 并在 Patricia Seybold Group 担任分析师。作为分布式计算技术领域的知名专家, Manes 还参与了 W3C, OASIS, JCP, UDDI 和 WS-I. 的标准开发。她还是业界最权威出版《Web 服务期刊》编辑专家组的会员之一; 多次参加会议演讲, 发表无数文章, 并著有《Web Services: A Manager's Guide》一书, 由 Addison Wesley 出版。

问: 听说 REST vs. SOAP 之间的对比, 有点像苹果和橙子之间的对比, 你同意还是不同意呢? 为什么呢?

答: REST 跟 SOAP 的对比, 正如将小汽车和轮胎对比一样。REST 是一个体系结构风格。SOAP 是一个消息传递协议。REST 应用程序可以使用 SOAP。W3C SOAP 1.2 建议书正是针对支持 REST 的应用程序特别设计的。

(作者: Anne Thomas Manes 来源: TT 中国)

REST 与 SOAP 之比较——SOAP 篇

本文的标题“REST 与 SOAP 之比较”确实有些让人误解。REST 是代表性状态传输的名称首字母缩写，与其说它是标准，不如说是一种风格。然而，在我的前一篇文章中，正如我们所讨论的，众多从事 Web 服务的软件设计师们认为 SOAP 过度复杂，于是，类似 eBay 和 Google 的服务都采用了 REST 风格的约束来暴露其大量数据。

比较 REST 和 SOAP 的“风格”

REST 依赖一套简单的“动词”，把所有的复杂性都转移到了指定资源的“名词”中。与此不同，SOAP 却有一套相当复杂的 XML 格式化命令和数据传输选项。

在 Web 服务发展的初期，XML 格式化消息的第一个主要用途是，应用于 XML-RPC 协议，其中 RPC 代表远程过程调用。在 XML 远程过程调用 (XML-RPC) 中，客户端发送一条特定消息，该消息中必须包括名称、运行服务的程序以及输入参数。相反，REST 风格的请求却不关心正在运行的程序是什么，它仅仅请求命名资源。

XML-RPC 只能使用有限的数据类型种类和一些简单的数据结构。人们认为这个协议还不够强大，于是就出现了 SOAP——其最初的定义是简单对象访问协议。之后，大家逐渐意识到 SOAP 其实并不简单，而且也不需要必须使用面向对象语言，所以，现在人们只是沿用 SOAP 这个名称而已。

XML-RPC 只有简单的数据类型集，取而代之，SOAP 是通过利用 XML Schema 的不断发展来定义数据类型的。同时，SOAP 也能够利用 XML 命名空间，这是 XML-RPC 所不需要的。如此一来，SOAP 消息的开头部分就可以是任何类型的 XML 命名空间声明，其代价是在系统之间增加了更多的复杂性和不兼容性。

另外，非常重要一点是，REST 是需要请求 HTTP 的，与其相比，SOAP 更具优势，SOAP 消息可以由所有能够处理 Unicode 文本的传输方式来传送，很可惜，这一点通常不被人们所认可。事实是，由于 HTTP 穿透防火墙的便捷性，以及开发商们对 Web 非常熟悉，因此，人们还在继续强调着 HTTP 传输。

随着计算机行业的觉醒，人们发现了基于 XML 的 Web 服务的商业潜力，于是，各家公司开始不断地发掘想法、观点、论据以及标准化尝试。W3C 曾经设法以“Web 服务活动”的名义来组织成果展，其中也包括实际做出 SOAP 的 XML 协议工作组 (XML Protocol Working Group)。与 Web 服务有关的标准化成果——从某种程度上说与 SOAP 相关或者依赖于 SOAP——的数量已经倍增到了令人惊讶的程度。

最初，SOAP 是作为 XML-RPC 的扩展而发展起来的，它主要强调的是，通过从 WSDL 文件中所获得的方法和变量名来进行远程过程调用。现在，通过不断进步，人们发现了更多的使用 SOAP 的方式，而不仅仅是采用“文件”方式——基本上是使用一个 SOAP 信封来传送 XML 格式化文件。无论如何，要掌握 SOAP，了解 WSDL 所扮演的角色是最根本的。

Web 服务描述语言或 WSDL

为了创建一个用于描述 Web 服务的 XML 格式化文件，Web 服务描述语言 (WSDL) 标准提供了足够多的细节，以便能够构建出客户端代码，从而访问服务或者服务器端代码以提供服务。一个服务的 WSDL 文件将会为你提供以下几个方面的内容：

- 用于访问服务的地址信息
- 用于传送信息的传输协议 (例如，通道数)
- 用于所有可使用功能的名称和接口使用方法
- 在所有的请求和响应中所使用的数据类型

2001 年 3 月，W3C 推出了 WSDL 1.1 版本用于讨论，这并不是最终确定的规范。W3C Web 服务描述工作组目前正在开发该规范的 2.0 版本，基本上已经到了尾声。虽然，WSDL 通常是用于特定的 SOAP 服务，但是，从理论说，它是完全可以用于特定的 REST 风格的 GET 或者 POST 操作的。

能够根据服务的 WSDL 描述来自动创建客户端和服务端代码，支持这一功能的开发环境目前使用得很广泛，以便能够适用于 Web 服务器和 Web 服务客户端的不同程序设计语言。如果你使用 Google 搜索“SOAP IDE”的话，大概会出现上百万条相关信息。也有这样的工具，根据 Java 或 C# 对象来生成相应的 WSDL 和代码。自动生成代码也许能够使你的开发效率更高，但是离优化却是越来越远。

安全与 SOAP

如果企业使用 SOAP 来传送有价值的信息的话，那么，安全就是最重要的问题。由 OASIS 组织发起，计算机行业的领导者们已经联合开发了一套标准，称为 WS-Security。这个标准对基本的 SOAP 通信做出了改善，以便能够处理以下几个问题：

消息机密性——由于拦截 HTTP 消息的方式非常多，因此，在请求和响应过程中，必须能够对所有重要信息加密。很幸运，现在的加密技术非常先进，我们能够对消息内容进行加密，以保证消息不被修改。

客户和服务身份——必须能够核实 SOAP 请求来源的身份。

结论

在开发人员的意识里，对于 Web 服务的开发而言，REST 和 SOAP 风格各有千秋。SOAP 拥有更为详尽的标准化成果和开源工具。除此之外，现在，有许多集成开发环境能够在现有代码的基础上，依据接口方法自动生成 SOAP。如果你需要使用 WSDL 来发布你的服务，或者你需要一些安全功能如消息签名和加密，那么，SOAP 能够确保消息的安全性。另一方面，如果你希望使用简单接口来公布一些信息，而不需要繁琐的处理过程，那么，REST 也许是最佳选择。

(作者: William Brogden 来源: TT 中国)

何时是使用 SOAP 或者 REST 的适当时机？



Anne Thomas Manes, Burton Group 的研究部总监，是 Web 服务领域知名的技术专家。她曾在 2001 年被《网络世界》评为“网络最具影响力的 50 位人物”之一；2002 年被《企业系统期刊》评为“100 名杰出 IT 领导者”之一。加入 Burton Group 之前，Manes 曾是 Bowlight 公司的创建者和 CEO，专注软件行业分析和咨询。她具有二十四年的行业经验，曾担任 Systinet 公司（从事 Web 服务基础设施）的首席技术官，革新 Sun 的 Web 服务战略，并在 Patricia Seybold Group 担任分析师。作为分布式计算技术领域的知名专家，Manes 还参与了 W3C, OASIS, JCP, UDDI 和 WS-I. 的标准开发。她还是业界最权威出版《Web 服务期刊》编辑专家组的会员之一；多次参加会议演讲，发表无数文章，并著有《Web Services: A Manager's Guide》一书，由 Addison Wesley 出版。

问：何时是使用 SOAP 或者 REST 的适当时机？

答：与其将 REST 和 SOAP 进行比较，还不如将 REST 将和 SOA 进行比较。REST 是一个体系结构类型。而 SOAP 是一个消息传递协议。REST 应用程序可能使用 SOAP 协议，但是在大多数的情况下，REST 应用程序只是简单地使用基于 HTTP 的 XML 传递消息。使用 SOAP 代替简单的基于 HTTP 的 XML 的好处在于，SOAP 提供了一种机制，这种机制可以从应用程序的有效负载中清晰地分离出基础功能。HTTP 消息传递包括了消息头，该消息头可以用于传递会话和安全信息，但是它从应用程序有效负载中分离基础功能的能力相当有限。SOAP 头可以包含任意类型的信息，从而可以使中间件为应用程序进行更多的运行时处理。（这些运行时处理包括了路由，消息传递，安全，审计，延续性，会计，以及很多其它的服务）。

总的来说，REST 比起 SOA 更具有扩展性，所以对于高扩展性要求的应用程序来说，我更倾向于选择 REST，但是除了可扩展性以外，还有很多其他的标准需要考虑。如果需要的话，你也可以让 SOA 具有扩展性。

我认为，当在 SOA 和 REST 之间进行选择时，开发者和架构师的主要依据是编程风格和开发者的开发经验。使用 SOA 的时候，应用程序主要将带有请求操作和提供输入信息的请求信息发送到服务方。服务对输入信息进行处理并返回相应信息。（SOA 还可以支持其他消息交换模式以及异步编程）。使用 SOA 的时候，底层基础设施可以将 XML 和语言专有对象匹配起来。（如 XML 到 Java）

在使用 REST 的时候，应用程序会将所有的请求信息编码为 URL。（然而你可能不想将敏感信息编码为 URL，例如信用卡号码）。应用程序可以使用 POST 方法提交用于处理的信息，例如订单号码等。在使用 REST 的时候，应用程序一般直接使用 XML。

如果开发者习惯于使用语言对象，并且使用 RPC 或者 RMI 类型编程模型，他们会认为 SOA 更为直观。如果开发者习惯于使用 SAX 和 DOM，并且使用消息传递类型编程模型，他们会认为 REST 更为直观。

（作者：Anne Thomas Manes 来源：TT 中国）