



轻量型框架资源手册

轻量型框架

在 EJB 技术之前，我们开发一个复杂 Java 企业应用系统时，会在代码设计中充满各种底层技术的味道，或者说，那时的 Java 更象 C++，Java 企业应用开发者需要学习更多更全面软件技术才能动手编程。EJB 则将很多底层技术：缓存、池、安全以及事务封装在特别的 EJB 服务器中，从这点上解脱了开发者的工作。在这个时代，Java 技术重心就在于 J2EE 服务器，或者称应用服务器，中间件服务器。各个厂商争相研制功能强大，能够支持几十台集群计算的 EJB 服务器，并依次为荣，Bea 的 Weblogic 是这方面的佼佼者。但是，EJB 问题同时也暴露出来了，在 EJB2.1 以前，复杂的 XML 配置；各种稀奇古怪的调试运行问题使的 EJB 开发调试变得非常不容易，开发者经常在错误信息中迷失方向。

当然好戏最终以轻量化的 EJB3.0 出世而收场，但是我们还是很有必要思考一下：我们开发者到底需要什么样的技术？这样才能在瞬间变化的发展潮流中坚持自己观点，而不是人云亦云，迷失方向。

关于 Spring

Spring 打着轻量简单极具号召力的口号横空出世，可以说恰逢其时，很快赢得了广大开发者的响应，虽然这场 without EJB 运动中有时过于极端，但是至少表达了大家对 EJB 2.1 的不满，虽然在这个 EJB 阵营，聚集中软件领域最顶端的厂商权威：IBM BEA Oracle SAP SUN 等，但是开发者还是坚决地说 NO，并且进行了自己的选择。

- ❖ 轻量级 Java 框架和面向服务架构
- ❖ Java One: Mule 设计师希望将 REST 带入 SOA

关于 Ruby on Rails

Ruby on Rails 是一个 Web 应用程序框架, 是一个相对较新的 Web 应用程序框架, 构建在 Ruby 语言之上。它被宣传为现有企业框架的一个替代, 而它的目标, 简而言之, 就是让生活, 至少是 Web 开发方面的生活, 变得更轻松。

ruby on rails 使用的实时映射技术和元编程技术, 免去了开发人员在开发过程中编写大量样板文件代码的烦恼。在少数需要使用样板文件代码的时候, 开发人员可以通过 ruby on rails 内建的生成器脚本实时创建, 而不再是通过手工编写。rails 的这个特点可以使开发人员更专注于系统的逻辑结构, 而不必为一些琐碎的细节所烦扰。

- ❖ SOA 结束了大型程序的年代 为 Ruby on Rails 敞开大门
- ❖ 全面解析 Ruby on Rails 的 Web 服务

轻量型框架未来发展方向

轻量框架一个特点是: Ioc 或称 Dependency Injection (DI), 通过 Ioc 模式的使用, 所有的 POJO 可以实现最大化的松耦合, 具体详情可了解 Ioc 模式/DI。

Ioc/DI 是所有轻量框架的一个共同特点, 不同的只是具体 Ioc/DI 实现方式不同, DI 使用轻量化的框架容器将服务或其他对象注射进入 POJO。这样, 所有对象实例的创建和管理由容器实现, 这样, 大大减轻了开发者的工作。目前, Java 世界轻量型框架层出不穷, 从当初的 Spring 到 HiveMind 以及 JBoss Seam, JdonFramework 也是符合轻量框架特点的一个国产开源中间件产品。可以说: 轻量型框架是未来构件架构发展方向。

-
- ❖ Web 服务性能构成要素
 - ❖ 创建并部署基于 Jersey 的 RESTful Web 服务
 - ❖ 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

轻量级 Java 框架和面向服务架构

有些事情碰巧以架构的名义不完全按计划成功进行。企业 Java Beans 是其中之一。尽管许多个人和公司在较大规模的 J2EE 标准内创建这个组件的标准有一定任务，EJB 把一些循环如 IBM 对 Java 的重大贡献带进来。有时，EJB 被描述为臃肿的，有时更糟糕。

EJB 在思想方面要早于 Java，因为它紧随在 IBM 用旧金山组件为代码命名的努力之后。EJB，特别是因为它的演变，被广泛认为过于复杂。这可能是因为它试图完全涵盖所有。

更窄的焦点框架已经出现，用来作为完全成熟的具有 EJB 的 J2EE 的替代。其中最主要的是 Spring 框架，它使用面向外观的编程技术，以确定总体的设计模式，而集中在当天最常见的工作，网络应用程序成为重点。

来自 SearchSOA.com 姊妹网站 TSS.com 下 2008 年的 Java 发展趋势调查的初步数据显示，76.8% 的调查参与者使用了 Spring。作为商业代理管家的 Spring 是 SpringSource，致力于促进 Spring-style 应用建设。我们同公司的创始人总裁 Rod Johnson 进行了交谈。他告诉我们，Spring 和 SOA 有很强的互补性。

便携性是驱动 Java 的关键，如果服务捆绑到服务器，就还没有最终决定。

Johnson 认为“SOA 是区别于整合的不同事情，这也正是我们认为 Spring 是有益的。”

他继续说道“在使用 EJB 的 J2EE 的传统方法往往在非常特定的环境下约束您的代码：单片机应用服务器环境。虽然 Spring 可以令人满意[在单片服务器环境]，Spring 容器本身也是便携式”。

能够使用 Spring 可以更好地保证，‘对分离关注’。这是 Spring 的目标，并以它的模型视图控制器（MVC）来概括。

Johnson 表示使用 Spring “您可能有更大的自由，重用您的业务逻辑，并在非传统的部署方案上进行部署。”

Johnson 解释道“EJB 先于 SOA。这是组件模型和分布式计算技术为一体的技术。它还假设你会使用 RMI 或 CORBA 的远程风格。EJB 模型仍然是耦合到应用服务器。问题是在大多数情况下，结合组件模型和分布并没有工作。并提出了对最终服务的很多假设，这是一种均匀分布的粒度的世界，你在导线两侧都有 Java。它并不能反映今天的现实。”

他表示 Spring 框架的哲学是你应该有一个独立于任何远程战略或分布模型的高品质的组件模型，这打破了通过应用服务器耦合的 EJB，为企业架构师提供更多的选择。

现在，Spring 社区正在制订一项建立在‘概念合同第一’的模式上的 Web 服务项目。Spring 3.0，将透过大量的 Spring MVC 的支持 REST。Spring 组件模型将自动支持 REST 风格的通信。

虽然许多成功的应用已使用 EJB 建成，它从来没有完全被欢迎。在表面看来，对于轻量级框架的东西，有一个‘让我们做吧’的态度。这并不意味着，这种办法不能在大公司的 SOA 项目存在合作存在，但并不意味着他们支持 SOA。

(作者: Jack Vaughan 来源: TechTarget 中国)

Java One: Mule 设计师希望将 REST 带入 SOA

合 JMS 力量的 Spring framework 和 Mule ESB 是建造 SOA 的最佳方法。MuleSource 公司的软件设计师和 Xfire 的创建者向企业开发商推荐 Spring framework 和 Mule ESB。在 JavaOne 的会谈上，他同样也谈到了自己在 Mule Galaxy、在基于 REST 的开放源注册表/储存库的工作。该基于 REST 的开放源注册表/储存库这个月末将进入 GA。

既然你已经参加了 JavaOne，现在我想问问你对 Sun 公司的 JavaOne 管理工作下 Java EE 目前的状况如何看待呢？



Dan Diephouse:我认为 Sun 公司取得了良好的进展。我们许多客户都采取了轻量级的方法。在 Java EE 中，他们不需要太多的员工，所以我认为 Profile 对于 Sun 公司来说是一个明智的举措，也是个出路。现在，我看到许多人都在使用这种方法，这也是我们所采用的方法。人们会说：“好吧，我将使用 JMS，我将要用 Spring。为了集成化我们将引进 Mule。Java EE 在朝着正确的方向发展。但是我仍然能看到人们在一段时期内会使用该最佳方法。

最佳方法可能是长期最好的方法么？

Diephouse:我个人认为这是最佳方法。许多人都在寻找，但是人们在使用的却是最佳方法。他们在使用 Spring 和 Mule。

Spring 似乎在开发商之间很受欢迎，你见过有很多人使用么？

Diephouse:我们许多用户都在使用它。Mule 实际上将 Spring 捆绑起来，所以我们的结构是建立在 Spring 之上的。它是一个非常强大的轻量级架构，并且应用非常普遍。

如果说 Spring 非常普遍，你是如何看待 EJB 的呢？假设你的 druthers 要选择走 Spring 或者是 EJB 路线，你更喜欢哪个？

Diephouse:我更倾向于选择走 Spring 路线。因为，这更简单。他们在那里建了许多东西，并且更易使用。

说到你在 MuleSource 的工作，目前你在关注些什么？在未来的几个月里，我们能看到哪些新的事物？

Diephouse:你也许对 Galaxy，我们的注册表/储存库很熟悉。我们看看市场，看到了一个开放源注册表/存储库的机遇。看到人们需要管理自己的服务和应用程序，但是同时也极度需要开放源共同体，因为整个注册表空间被我的一个同事称作“富人的游戏”参加这个游戏花费太大。解决方案是以 Web 服务为中心，并且目前不一定能反映开发商们的基础设施。

许多人在做 RESTful 和其它方面的事情时，你也需要能够管理这些事情。注册表和存储库都是很贵的。它们都是以 Web 服务为中心，它们可以是重量级的但是它们不会与开放源工具整合。所以我们都是从自己的注册表和存储处开始的，我们说要以开放源的方式开发。我们把它变成轻量级的。我们使其更易使用。我们将它和所有开发商使用的从主机到 Spring 乃至 Mule 工具整合起来。我们通过一个 RESTful API 将其扩展。我们使用 Publishing Protocol API，并且比起 UDDI，它更容易使用。

所以我们仅仅是试图建立一个人们需要和能使用并且能够帮助教会人们如何使用的方案。所以我们首要关注的是如何建立该方案？

此刻 Mule Galaxy 注册表/储存库项目的状态如何？你在试用该项目么？

Diephouse:我们在试用。并且五月就要进入 GA。所以这个月我们有 1.0 版本进入 GA，到六月，我们将有一个企业版本。

你提到了 REST 和 Atom，但是你能更加详细的告诉我们当 Galaxy 进入 GA 时，什么是可行的吗？当这些开发商下载了 GA 版本时，他们能发现什么？

Diephouse:其核心是一个可伸展的人工制品和元数据储存库，因此你能在里面储存东西并管理元数据。然后我们在其周围添加类似服务和人工制品生命周期管理之类的特征。因此我要问：在储存库中的东西都在 QA 中么？到底在生产些什么？因为有依赖性管理，所以我们能想象，什么依赖该人工制品和 WSDL。这样你更清楚谁在使用什么弄明白逆能力之类的事。这里有个政策管理事项，所以我能执行政策。我可以说：“我可以保证所有的 WSDL 都听从 WS-I Basic Profile 。所以它们和那里所有的 Web 服务都可以相互操作。你也可以实施安全政策。你可以说：“我要保证每个人都在他们自己的 Web 服务中使用 WS-Security。”用户可能有上百种的服务，他们只需要找出什么服务是可行的，怎样才能真正使用这些服务就可以了。

所以并不是我们只可以同 Web 服务一起工作，我们也可以和像 OSGi 服务之类的服务一起工作。一个客户可能在 JARs 或 OSGi 束内部有自己所有的服务。你想从 IDE 中指出并研究这些服务。所以在 Galaxy 里，它们都是可以被研究和查询的。你可以看到那里到底有什么。通过 Atom API, 这个非常简单的插件，你就能把它们取出来，并在你的 IDE 中以一个开发商的眼光观察什么服务是可行的。你可以点击并查看什么服务是可行的。

和其它的注册表/储存库相比，Galaxy 所提供的注册表/储存库有那些优势？

Diephouse:我们的一个优势是，我们不特别关注 Web 服务。我们尝试同其它的服务一同工作。另一个优势我们是开放源的，所以用户可以找到我们。已经有人找到我们并为不同的平台编写插件。这些插件可能没和注册表/储存库捆绑在一起。另一个优点是我们和类似 Maven, Spring, Mule, Apache 的开放源基础设施整合在一起。这就非常容易编写你自己的插件

Xfire 的创始人 Diephouse 在星期一的第二部分谈话中探讨了为什么不管 UDDI 发生什么，Mule 将在注册表/储存库方面超过 Web。

(作者: Michael Meehan 来源: TechTarget 中国)

SOA 结束了大型程序的年代 为 Ruby on Rails 敞开大门

Engine Yard 公司的首席技术官 Tom Mornini 表示，单机百万线应用的时代已经结束，面向服务架构(SOA)是这一时代的终结者。该公司提供 Ruby and Rails 主机服务器。

他在最近的采访中说“我认为使用大型程序的年代已经结束了”“有些程序看起来很大，但是随着时间的推移，它们将最终成为许多小程序的结合体。”

通过为全球市场的业务提供灵活性，SOA 的可组合性改变了应用开发比赛。在全球市场中，商业机会不是一成不变的。

Mornini 说“我实在看不出任何其他方式可以满足存取数据，改变流体的需求，以便在企业内外跟上时代的步伐。”“这就是为什么未来能解决所有问题的单机百万线应用在这一点上仅仅是个遗迹。”

Mornini 认为，这不再是 SOA 是传统应用开发选择的问题，而是除了 SOA 以外，我们没有其它的选择。

他说“这些大型程序很难管理和维护，很难想像单机应用会成为未来发展的方向”。

Engine Yard 公司的首席技术官认为带有 REST 的 Ruby on Rails 是为 SOA 建立新一代的服务和应用的一种方法。与 Java 不同，Java 是在 SOA 应用开发时代前开发的项目，他注意到，Ruby on Rails 和 REST 怀抱 SOA 为理念向世人提供了一个前所未有的方法。

Mornini 说“拥有一个服从该框架的牢固而又深厚的面向服务架构就是 Rails 的秘诀”该架构的开发商认为（它的 SOA 功能）是该平台的一大优势。

他认为 Ruby on Rails 非常适合 SOA 开发。新发布的 Rail 2.0 令该框架更容易为 SOA 应用以及旧数据存取所接受。他承认，原有的 Rails 框架与旧数据存取关系并不是十分融洽。今年推出的新模型已经超过了前者。

他说，例如，Rails 组提供的代码增加了许多新的功能，通过以服务的形式将旧数据曝光，使得在 SOA 应用中访问旧数据变得更为简便。

Engine Yard 公司的首席技术官说 “由于遵循了售后服务书籍和网络视频记录的规程，Rails 令开发商使用 RESTful 数据变得更为简单”。

他说，“如果你遵循 RESTful Rails 的标准过程，在系统外用 Rails 编写了一个程序，就会自动得到该程序展示的一个建立在 XML-over-HTTP 基础之上的 API。

但是如果要使其运转，“继续使用 Rails”很重要。Mornini 说这就是 Rails 遵循既定规程的妙招。

(作者: Rich Seeley 来源: TechTarget 中国)

全面解析 Ruby on Rails 的 Web 服务

Ruby 曾只是一种模糊的、面向对象、动态类型的“脚本语言”，直到 Rails Web 应用框架的出现，Ruby 才突然引起人们的关注。两者的结合使用，很快以“Ruby on Rails”为人所知。我猜想，其原因可能是这种叫法传达出的形象远甚过“Rails on Ruby”吧——不管怎么样，我们将它缩写为“RoR”。

作为一种解释型语言，Ruby 的性能并不是特别出众，但是由于 RoR 非常流行，人们花了很多努力对此加以改变。最初的解释器由 C 语言编写，但是现在的版本采用 Java (JRuby)或.NET (IronRuby)编写，可能还有其它的语言版本。目前最稳定的 Ruby 版本是今年三月发布的 1.8.6 版。1.9 版本有望出现重要更新，而 2.0 版本则会有极为重大的变化。

Rails 是什么？

Rails Web 应用框架并不是试图成为所有开发人员的一切。它主要注重使用关系型数据库和“模型—视图—控制器”设计模式的应用。最初，Ruby on Rails 是由 David Hansson 创建的。当时他正在 Web 设计公司 37Signals 开发一个项目管理工具。第一个开放源码的版本于 2004 年发布。Web 开发人员发现 RoR 使创建面向数据库的 Web 应用程序变得非常简单，这个行业“术语”也因此迅速出现。

开发人员发现 RoR 中许多内置的假设条件和命名规则大大减少了运行基本 Web 应用程序所需的代码行。命名规则取代了其它 Web 应用框架常用的复杂的 XML 配置文件。RoR 的支持者称之为“惯例优于配置”原则。

RoR 1.2 版发布于 2007 年 10 月，作出了很大改进，如支持 REST 风格的请求。Rails 2.0 现处于“release candidate 1”阶段，有望支持更多的 REST。Rails 目前已支持 SOAP，不过 Rails 开发人员鼓励使用 REST 方式。

MVC 和 Web 应用程序

“模型—视图—控制器”（MVC）设计模式首次出现在 Smalltalk 中。Smalltalk 是 Xerox 公司开发的面向对象的先驱语言。许多开发人员认为，用 MVC 方式思考能够帮助理清复杂应用程序的结构。

模型

通过将数据存储和检索的所有内容放入“模型”组件中，开发人员可以将细节分离到一个定义明确的界面背后。可能在开始设计项目时，你使用的是一个包含虚假数据的模型，然后转移到使用同一界面、由主存数据库支持的另一模型中，最后升级到大型的商业数据库，但不改变项目的其它内容。

在 Rails 的应用中，模型类建立在能提供所有基本功能的基类（base classes）之上。每个数据表中创建一个客户模型类，再从表中提取名字，以创建访问方式。RoR 工具提供了可以从模型类中自动产生源代码框架的 Ruby 脚本。

视图

“视图”用于以特殊的方式提供数据。目前，Web 应用程序可能需要提供多种不同的视图。尽管 RoR 最初是为 HTML 网页而设，但新版本也能支持 Web 服务中使用的多种格式。通常，视图由模版文件产生。模版文件包括经常出现的各种要素，如 HTML markup 标签、plus 标签等，这些标签可以由一个或多个模型中的数据进行动态评价。

RoR 提供的基类支持 HTML、XML 和 JavaScript 格式的视图生成。下面的表格给出了符合格式的客户端形式。JSON 代表 JavaScript Object Notation，这种格式能够传送 XML 之类的复杂数据，而且能节省速度和批量。

格式	客户端	举例
HTML	典型的浏览器	充斥着用户具体信息的HTML网页
XML	使用JavaScript的浏览器	AJAX 式动态网页
JSON	使用JavaScript的浏览器	使用JSON 而不是 XML技术的动态网页
JSON	Web服务客户端	Python 和支持JSON解析的其他语言

XML 模版可以用于创建简单的 SOAP 消息，但是大多数情况下，利用 RoR 创建 SOAP 或 XML-RPC Web 服务似乎都会在控制器组件中产生很多输出，从而干扰标准的 MVC 模式。我能找到的 SOAP 服务例子都采用的先进的 WS-*特征并不比其他框架采用的多。RoR 开发人员似乎都更重视将 REST 式的 Web 服务作为未来发展的趋势。

控制器

“控制器”通常用于处理来自用户或服务器的请求输入，以决定采用哪种模型对象处理数据、采取什么措施以及哪类视图能产生回应。控制器类能够利用大型的帮助类软件库来处理 HTML 要素。RoR 的最新版本建立了控制器基类，能够处理使用 REST 术语的 URL 请求，从而加大了对 REST 式请求的支持。

Ruby 和 Rails 的合法性

Ruby 语言是具有版权的免费软件，拥有自身的许可条件。Rails 框架是在 MIT License 极为开放的条件发布。David Hansson 已经提交了对“RUBY ON RAILS”、“RAILS”和其它官方 Rails 标识使用的商标保护文件。最初试图采用基于 Java 的脚本语言 Groovy 进行平行开发（即 Groovy on Rails）的想法遭到了 Hansson 的反对，因此目前这个项目称为 Grails，而且是独立开发。

RoR 平台

Rails 开放源码开发人员通常采用 Apple Mac OS X，最新发布的 OS X 10.5 版本（即 Leopard）包括了 RoR。使用 Windows 和 Linux 的开发人员在安装 Ruby 应该不会出现什么问题，一旦安装了 Ruby 语言，“RubyGems”软件包管理器可以负责安装 Rails 的最新发布。你还需要配置关系型数据库，不过 RoR 能与最流行的开放源码和商业系统合作。MySQL，SQLite 和 PostgreSQL 都是很流行的选择方式。还有多种包含 Web 服务器和数据库的软件包，可以使安装更加简单。

目前很流行的 IDEs（集成式开发环境）——Netbeans 和 Eclipse 加进了支持 Ruby 编码的模组。还有一些项目也可创建 IDEs，且专门支持 RoR。这些情况大多数基于 Eclipse。

结论

目前，RoR 主要用于创建基于浏览器的 Web 应用程序。“模型—视图—控制器”设计模式非常灵活，因此可以扩展应用，在不改变数据模型组件的情况下覆盖大量客户端。开发人员若感觉在 Web 服务中使用 SOAP 太复杂，一定会对支持 REST 的 RoR 很感兴趣。

(作者: William Brogden 来源: TechTarget 中国)

Web 服务性能构成要素

自从 Web 服务被当作应用程序后，依照响应时间和存储需要，Web 服务的性能就成为了我们要考虑的主要因素。使用 XML 和 SOAP 是 Web 服务高效性的最大障碍，许多工具包的开发者都在解决性能问题方面付出了努力。本文的目的就是将 Web 服务的构成分成几个要素。

- 信息传递
- 信息分析
- 建立服务对象
- 后端流程
- 建立响应
- 信息传递

一般来说 REST 服务要求非常紧密，而 SOAP 信息要求相对于整个庞大的信息规模而言有效负荷的规模就很小了。SOAP 信息中命名空间引用的激增也会增加对带宽的要求。我之前在一篇文章中看到的 Fast Infoset 标准意在使 SOAP 信息更为紧凑，分析更为迅速。Sun 公司公司的开放源 Glassfish 服务器项目包括 Fast Infoset 实施，但是我不能肯定是否有人在用它。

这里还有许多建立 XML 信息的方法。有些人将其概括为标准的“gzip”压缩算法，这种算法可以建立一个良好的紧凑信息。由于 gzip 可以用几种语言编写，所以在 HTTP 服务器的应用更为广泛，并且很容易实施。

信息分析

在这一步骤中，服务必须通过一些工具包，和参数方法来确定要求的目标。目标和参数可以直接通过 RESTful Web 服务从 URL 中进行抽取，但是需要分析 SOAP 信息。

人们在改进 SOAP 要求的 XML 分析阶段耗费了很大精力。所有的 XML 分析程序都要扫描引入字符并且要识别多个 XML 要素，将所有的要素数据打包到特定的编程语言，例如 Java 对象。这些分析程序的不同之处在于如何处理这些数据。

在文档对象模型或者是 DOM 中，当内存对象复制原有 XML 文档时结束时，所有的 XML 标签和真正的请求数据也随之结束了。创建 DOM 会占有很多 CPU 内存和时间。

DOM 的一个好处就是像 XPath 这样的工具可以通过使用简化标记定位数据。原有的 Apache SOAP 工具箱使用 DOM，但是速度很慢，还会占很大内存。

XFire 和 Axis2 工具箱项目使用的是构造 SOAP 的最新方法，该方法意在通过使用事件驱动的分析程序来提高性能，例如 StAX。StAX 被称为“牵引”分析程序，因为它能生成所需的事件，服务可以借此找到一切所需的数据，在整个 SOAP 信息读取之前就可以停止分析，所占的内存也相对较小。

XFire 工具包的高速度是出了名的，它于 2005 年建成。并于 2007 年五月在市场发布了 XFire1.2.6 版本。但是当 XFire 项目和 IONA 公司的 Celtix 项目结合在一起组成了一个叫做 CXF 的 Apache 项目时，CXF 却从 Axis2 项目分离了出来，所以现在是两个由 Apache Software Foundation 封装的完全分离的开放源高效 Web 服务项目。在这些基于 StAX 的工具包中，分析和建立服务对象同步进行。

建立服务对象

建立服务对象通常指当向服务要求抽取价值时所发生的“数据绑定”步骤，这些价值传递的方法会完成服务要求。通过这些基于 Java 的服务，建立新的对象会将抽取的价值进行初始化。

这个步骤中有一个不断扩散的方案，有些人为了 Web 服务设计了全新的方案，有些人则是在现有的应用框架上安装了插件例如 Spring。从这个基准点来看，数据绑定会使生产量得到大大的提高。CXF 和 Axis2 试图尽可能的支持更多的数据绑定方案。常用的数据绑定方法包括：

- JAXB: XML 绑定的 Java 应用程序接口是标准 Java 数据库的组成部分
- POJO: Axis2 和 CXF 都为建立在 Plain Old Java Objects 基础上的 Web 服务建立提供至少一个以上的方法
- Spring 框架对象: Spring 是一个广为应用的开放源应用框架
- Castor: 对于 Java 来说是一个流行的开放源 XML 绑定框架

后端流程

借于 Web 服务的设计的原因, 后端流程, 例如, 和其它阶段相比, 数据搜索整理会占用的时间更久。如果你的潜在 Web 服务也属于这种类型, 你就没必要按照这个标准选择 Web 服务, 这个标准只能衡量使用仿造结果的响应时间。

能够扑捉复杂后端流程的标准技术是一个全新的理念。通过复制要求的频率, 可以成倍提高速度。微软公司为建立 Web 服务量身打造的工具使用的是 ASP.NET, ASP.NET 含有一个机制, 程序设计人员依靠这个机制发出指令, 指定系统必须将输出复制到储藏处, 这些是输出将在储藏处保留一段时间。

建立响应

在原有 Apache SOAP 工具箱内, 通过建立输出信息的 DOM 来创建响应, 并将其序列化到响应流。很显然这样效率很低。通常用于数据绑定的不同工具箱也会提供对象数据序列化入 XML 的机制。这主要取决于 Web 服务工具例如 Axis2 和 CXF, 以便将数据封装到正确的 SOAP 外壳中。

(作者: Bill Brogden 来源: TechTarget 中国)

创建并部署基于 Jersey 的 RESTful Web 服务

我发现基于 Jersey 工程的 Web 服务发展支持可以作为插件为 NetBeans 所使用。我想看看是什么参与在 Jersey 工具包的基础上创建和部署 Web 服务。

首先我们要知道什么是 Jersey。

Jersey 和 the JSR-311 RESTful Web 服务 API

JSR 代表 Java 规范要求。是 Sun 公司资助的 Java Community Process 所采用的机制，用于招聘新的专家小组，以推动新 Java 标准的演化。JSR-311 是为 RESTful Web 服务的 Java API 而设计的产品。通常叫做 JAX-RS。JAX-RS 的目的是提供一个带有注释的 API 把 Java 类和接口联系起来，以便为 RESTful Web 服务建立一个资源类，这些接口可能会添加到 POJO（普通 Java 对象）。

Sun 公司资助的 GlassFish 小组开放源代码项目的一部分。Jersey 工程有任务提供工作着的参照执行 JSR - 311。除了 Java 接口和实施 API 的类以外，Jersey 提供了一个 servlet，它可以容纳资源类的任何成员。带有现时用户测验的工作中的执行非常重要，它为研究 JSR-311 规范的专家小组提供反馈，例如，用户组邮件列表中经常含有带有新特点的要求。

使用由 Java 语言编写的注释

注释是一个特殊的文字附注或元数据，通过 1.5 版本补充到 Java，注释在 Java 源代码可以影响编译和由此产生的 Java 类的运行期行为。在 Java 注释的意义类似于编译的 Java 接口声明。为了方便程序设计师，注释被设计者更多的应用于编程框架并使编程风格得以执行。从句法角度来说，注释用以编译器可以确认的@符号作为标识符，现在目前有 40 个 JSR-311 指定的标识符。

JAX-RS 注释是如何创建一个资源类的。

如果你正在使用 NetBeans RESTful Web 服务插件。它会在你建立源代码初期给与帮助，这些源代码中包含能够支持 REST 格式请求所要求的最小注解。让我们看一些有关注释的例子——以下添加到类的@路径注释，定义了 URI 的一部分，该 URI 用于定位类以及从 URI 中提取的变量的名称。

```
@Path( "matching/{domain}/{word}" )
```

所以如果你 Web 应用的名称是“MetaPhone”，并且控制资源类的 Jersey servle 被映射到“resources”，URI 请求一个地名表发音近似于“leander”，地址是
`http://myserver/MetaPhone/resources/matching/Places/leander`

正如我们在下面详细讨论的，RESTful 应用容器利用运行时间注释“GET”“ProduceMime”找到最适合的方法来处理 HTTP GET 请求，并且 ProduceMime 符合请求首部，PathParam 注释将输入参数从路径中提取出来。

```
@GET
@ProduceMime("text/html")
public String getHtml(
    @PathParam("domain") String d,
    @PathParam("word") String w
) {
```

我填补了 gethtml 方法，用一些事先预置在周围的语音代码查找代码和数据。返回的字符串是带有和地址表相匹配的 HTML 格式的页面。

一个 RESTful Web 应用的生命周期

在 Jersey，支持 JAX-RS 资源的 servlet 是位于
`com.sun.ws.rest.impl.container.servlet package.` 的 ServletAdaptor 类。RESTful

Web 服务的部署描述符将这个 servlet 映射到一个名称如“resources”并详细指明当服务器启动时，必须初始化。ServletAdaptor 支持。搜寻包含 JAX-RS 注释的 Java 类并监督其使用 HTTP 方法，确保其由 MIME 类型。除了 HTML 和纯文字外，Jersey 工程还支持 XML 和 JSON 输出类型。

任何支持不同 HTTP 方法和资源类型的资源类成员，都能在一个应用程序里得到支持。

JAX-WS API 为定位符合请求的最佳匹配提供规则。

当一个可识别的 URI 被转发到 ServletAdaptor 时，它利用 Java 反射建立一个资源类实例，并准备将嵌入在类里的注释输入到适当的方法里。无论应答是怎样的，选择方法的输出成为应答的正文。

重要的是每一个要求会导致一个资源类的新实例建立起来。当要求完成后，实例会被抛弃。尽管这个额外的对象建立可能显得有些浪费，它却大大简化了整个过程，这取决于程序管理员来提供静态方法或其它类来长期保持数据。

将其部署到一个 servlet 容器

如果你正在使用 NetBeans 寻求发展，它会自动创建一个包含所有数据库和类的 WAR 文件，这些数据库和类通过收集所有 Jersey 支持的数据库来支持服务。

将其布置到轻便服务器上

例如 Sun 公司发布的 Java SE 6，同 Jersey 文件样例里所显示的一样。可以用“lightweight Web server”创造最简单可行的服务器，它可以接收 HTTP 请求，并在单一资源服务器类里将请求映射到正确的方法，返回一个 HTTP 应答。轻便服务器的一个缺点就是没有提供所有能够为典型 servlet 定位的数据库。所以需要程序员把所有的 JAR 文件都放在类路径里。

我发现，在运行 Windows 时，作为一个独立应用能够启动轻便服务器的 Jersey 实例代码就会出现问题，该实例用一个标准该实例方法呼叫这个简单的方式让操作员能停止服务器。

系统读取 ()

这在 Unix Java 实施中工作正常，但却在 Windows 命令提示里，冻结了整个程序。解决方法就是让 loop - (ctrl)C 里的启动线程休眠打破回路，推出程序。

```
while( true ){  
    try { Thread.sleep(1000);  
    } catch(InterruptedException e) {}  
}
```

这个单独的例子用了不到 3MB 来支持一个简单的 RESTful 服务实例。轻便服务器可以嵌入更大的应用程序中为一个简单确定的资源提供访问。

Jersey 的前景

JerseyJSR-311 的实施正在迅速的演进。我用的 0.6ea 和 0.7ea 版本 4 月底就投入市场了。幸运的是，NetBeans RESTful Web 服务插件在每次产品发布不久就会升级，并为你正常 NetBeans 插件管理员所接受。最终，Jersey 将成为一个和 GlassFish Java EE 开源服务器一起运送的产品品质参照实施。但我不能确定什么时候能实现。

(作者: William Brogden 来源: TechTarget 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 来源: TechTarget 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 Javaspaces 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器。

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 来源: TechTarget 中国)