

<<RESTful Web Services>>

图书基本信息

书名：<<RESTful Web Services Cookbook中文版>>

13位ISBN编号：9787121143908

10位ISBN编号：7121143909

出版时间：2011-9

出版时间：电子工业出版社

作者：Subbu Allamaraju

页数：299

译者：丁雪丰,常可

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<RESTful Web Services>>

内容概要

本书是《RESTful Web Services Cookbook》的中文翻译版，由Subbu Allamaraju所著。

《RESTful Web Services Cookbook》(中文版)从实践出发，涉及设计 RESTful Web服务的各个方面，通过问题描述、解决方案、问题讨论的形式在14个章节中详细讨论了统一接口、资源、表述、URI、链接、请求、缓存、安全等诸多内容。无论读者是否设计过RESTful Web服务，具体使用哪种语言，都能在阅读过程中有所收获。本书也可作为手册，根据具体问题描述在书中查找解决办法。

<<RESTful Web Services>>

作者简介

作者：(美国)阿拉马拉尤 (Subbu Allamaraju) 译者：丁雪丰 常可 合著者：李锟

<<RESTful Web Services>>

书籍目录

前言

第1章 使用统一接口

- 1.1 如何保持交互的可见性
- 1.2 何时需要权衡可见性
- 1.3 如何维护应用程序状态
- 1.4 如何在服务器端实现安全和幂等的方法
- 1.5 如何在客户端处理安全和幂等方法
- 1.6 何时使用GET方法
- 1.7 何时使用POST方法
- 1.8 如何使用POST方法创建资源
- 1.9 何时使用PUT方法创建新资源
- 1.10 如何使用POST方法实现异步任务
- 1.11 如何使用DELETE方法实现异步删除
- 1.12 何时使用自定义HTTP方法
- 1.13 何时及如何使用自定义HTTP标头

第2章 识别资源

- 2.1 如何从领域名词中识别资源
- 2.2 如何选择资源粒度
- 2.3 如何将资源组织为集合
- 2.4 何时将资源合并为复合资源
- 2.5 如何支持计算或处理函数
- 2.6 何时及如何使用控制器来操作资源

第3章 设计表述

- 3.1 如何使用实体头来注解表述
- 3.2 如何解释实体头
- 3.3 如何避免字符编码不匹配
- 3.4 如何选择表述格式和媒体类型
- 3.5 如何设计XML表述
- 3.6 如何设计JSON表述
- 3.7 如何设计集合表述
- 3.8 如何保持同构的集合
- 3.9 如何在表述中使用可移植的数据格式
- 3.10 何时使用实体标识符
- 3.11 如何在表述中编码二进制数据
- 3.12 何时以及如何提供HTML表述
- 3.13 如何返回错误
- 3.14 如何在客户端处理错误

第4章 设计URI

- 4.1 如何设计URI
- 4.2 如何将URI用做模糊标识符
- 4.3 如何让客户端将URI视为模糊标识符
- 4.4 如何保持酷的URI

第5章 Web链接

- 5.1 如何在XML表述中使用链接
- 5.2 如何在JSON表述中使用链接

<<RESTful Web Services>>

- 5.3 何时以及如何使用链接标头
- 5.4 如何分配链接关系类型
- 5.5 如何使用链接来管理应用程序的流程
- 5.6 如何处理临时URI
- 5.7 何时以及如何使用URI模板
- 5.8 如何在客户端使用链接
- 第6章 Atom和AtomPub
 - 6.1 如何利用Atom建模资源
 - 6.2 何时使用Atom
 - 6.3 如何使用AtomPub服务和分类文件
 - 6.4 如何针对Feed和Entry资源使用AtomPub
 - 6.5 如何使用媒体资源
- 第7章 内容协商
 - 7.1 如何标明客户端偏好
 - 7.2 如何实现媒体类型协商
 - 7.3 如何实现语言协商
 - 7.4 如何实现字符编码协商
 - 7.5 如何支持压缩
 - 7.6 何时以及如何发送Vary头
 - 7.7 如何处理协商失败
 - 7.8 如何使用代理驱动的内容协商
 - 7.9 何时支持服务器驱动的协商
- 第8章 查询
 - 8.1 如何针对查询设计URI
 - 8.2 如何设计查询响应
 - 8.3 如何支持有大量输入的查询请求
 - 8.4 如何存储查询
- 第9章 Web缓存
 - 9.1 如何设置过期缓存头
 - 9.2 何时设置过期缓存头
 - 9.3 何时以及如何在客户端中使用过期缓存头
 - 9.4 如何支持复合资源的缓存
 - 9.5 如何保持新鲜且温暖的缓存
- 第10章 条件请求
 - 10.1 如何生成Last-Modified和ETag头
 - 10.2 如何在服务器端实现条件GET请求
 - 10.3 如何从客户端提交条件GET和HEAD请求
 - 10.4 如何在服务器端实现条件PUT请求
 - 10.5 如何在服务器端实现条件DELETE请求
 - 10.6 如何从客户端发起无条件GET请求
 - 10.7 如何从客户端提交条件PUT和DELETE请求
 - 10.8 如何使POST请求条件化
 - 10.9 如何生成一次性URI
- 第11章 其他内容
 - 11.1 如何复制资源
 - 11.2 如何合并资源
 - 11.3 如何移动资源

<<RESTful Web Services>>

- 11.4 何时使用WebDAV方法
- 11.5 如何支持跨服务器的操作
- 11.6 如何获取资源的快照
- 11.7 如何撤销资源更新
- 11.8 如何为部分更新提炼资源
- 11.9 如何使用PATCH方法
- 11.10 如何批量处理相似的资源
- 11.11 如何触发批量操作
- 11.12 何时使用POST来合并多个请求
- 11.13 如何支持批量请求
- 11.14 如何支持事务

第12章 安全

- 12.1 如何使用基本身份验证来验证客户端
- 12.2 如何使用摘要身份验证来验证客户端
- 12.3 如何使用三方OAuth
- 12.4 如何使用两方OAuth
- 12.5 如何处理URI中的敏感信息
- 12.6 如何维护表述的机密性与完整性

第13章 可扩展性与版本控制

- 13.1 如何维持URI的兼容性
- 13.2 如何维持XML和JSON表述的兼容性
- 13.3 如何扩展Atom
- 13.4 如何维持链接的兼容性
- 13.5 如何实现支持可扩展性的客户端
- 13.6 何时需要版本化
- 13.7 如何版本化RESTful Web服务

第14章 服务发现

- 14.1 如何编写RESTful Web服务的文档
- 14.2 如何使用OPTIONS

附录A 辅助读物

附录B REST概述

附录C HTTP方法

附录D Atom Syndication Format

附录E 链接关系类型

索引

<<RESTful Web Services>>

章节摘录

版权页：插图：在任何分布式的客户端 / 服务器环境中管理变更都是很困难的。

在这样的环境中，客户端依赖于服务器能履行自己的承诺。

RESTful Web服务也不例外，对它们而言，这包括uRI、资源、结构体和表述的内容、它们的格式，以及每个资源的HTTP方法。

直到您开始考虑向后兼容性（backward compatibility）之后，对服务器的变更好像都成了有益的了。

变化是向后兼容的，在修改服务器时就无须升级客户端。

客户端会忽略您升级了服务器这一事实，继续按原有方式进行调用，就好像什么变化都没发生过一样，当然这不包括服务器升级期间的不可用时间。

还有一种兼容性被称为向前兼容性（forward compatibility），当有多个客户端和服务器在不同时间点升级时，这一点是很重要的。

在这种情况下，较新的客户端可能会与老的服务器进行交互。

向前兼容性的目的是保证较新的客户端能继续使用老的服务器，虽然功能有所减少，但不会出现问题。

您的应用程序中仅需要考虑向后兼容性，还是需要同时考虑向后和向前兼容性取决于您的操作环境。

本章描述的内容能帮助您处理好这两种兼容性。

让您能够保持兼容性的特性是可扩展性，这是为未来的变化预留余地的设计过程。

<<RESTful Web Services>>

媒体关注与评论

“ 无论您打算使用何种语言构建RESTful服务，本书都是一本极佳的指南。
内容丰富、考虑周到，囊括了与服务设计与实现相关的很多最佳实践。
” ——Eben Hewitt架构师 “ Java SOA Cookbook ” 作者

<<RESTful Web Services>>

编辑推荐

REST设计哲学激起了Web与企业应用开发者的无尽遐想，但用这种方式来开发真正的Web服务并非易事。

《RESTful Web Services Cookbook(中文版)》包含了超过100个技巧。

帮助您充分利用REST，HTTP和Web基础设施。

无论使用何种编程语言和开发框架，您都将了解到如何为客户端/服务器应用设计满足性能、可扩展性、可靠性及安全目标的。

RESTful Web服务。

《RESTful Web Services Cookbook(中文版)》的每一节都包含一到两个问题描述，带有简单易学、步骤详尽的解决方案，还有使用HTTP请求与响应、XML、JSON和Atom片段的例子。

您还将看到针对每个解决方案的实现指南，讨论其中的利弊和权衡。

了解如何设计满足多种应用场景的资源，成功设计表述与URI，使用链接与链接标头来实现超文本约束，理解何时及如何使用Atom和AtomPub，为了支持缓存，该做什么，不该做什么，了解如何实现并发控制，处理涉及复制、合并、事务、批处理及部分更新的高级用例，保护Web服务，支持OAuth。

Subbu Allamaraju，Yahoo！

架构师。

他设计了RESTful web服务的标准及实践，目前负责开发者平台的架构工作。

Subbu之前就职于BEA System，Inc..开发Web服务及基于Java的软件。

他参与了4本J2EE~H关书籍的编写，均由Wrox发行出版。

<<RESTful Web Services>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>