

Web开发经典丛书

# RESTful API开发实战

使用REST JSON XML和JAX-RS构建微服务  
大数据和Web服务应用

Pro RESTful APIs: Design, Build and Integrate with  
REST, JSON, XML and JAX-RS, First Edition

[美] Sanjay Patni 著  
郭理勇 译

Apress®



清华大学出版社

Web 开发经典丛书

# RESTful API 开发实战

使用 REST *JSON* XML 和 JAX-RS 构建微服务  
大数据和 Web 服务应用



[美] Santjay Patni 著

郭理勇 译

清华大学出版社

Sanjay Patni

Pro RESTful APIs: Design, Build and Integrate with REST, JSON, XML and JAX-RS, First Edition  
EISBN: 978-1-4842-2664-3

Original English language edition published by Apress Media. Copyright © 2017 by Apress Media. Simplified Chinese-Language edition copyright © 2018 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Apress 出版公司授权清华大学出版社出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字：01-2017-5755

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。  
版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

### 图书在版编目(CIP)数据

RESTful API 开发实战 使用 REST JSON XML 和 JAX-RS 构建微服务 大数据和 Web 服务应用/(美)桑杰·帕特尼(Sanjay Patni)著;郭理勇译.—北京:清华大学出版社,2018  
(Web 开发经典丛书)

书名原文:Pro RESTful APIs: Design, Build and Integrate with REST, JSON, XML and JAX-RS, First Edition

ISBN 978-7-302-49211-5

I. ①R… II. ①桑…②郭… III. ①互联网络—网络服务器—程序设计 IV. ①TP368.5

中国版本图书馆 CIP 数据核字(2017)第 331724 号

责任编辑:王军 韩宏志

封面设计:牛艳敏

版式设计:思创景点

责任校对:曹阳

责任印制:杨艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 装 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:148mm×210mm 印 张:4.625 字 数:125 千字

版 次:2018 年 2 月第 1 版 印 次:2018 年 2 月第 1 次印刷

印 数:1~3200

定 价:48.00 元

---

产品编号:076424-01

# 译者序

每个互联网从业人员都有这样一种感觉：RESTful API 概念既熟悉又陌生。熟悉的是我们能在大量开放平台或开源项目中看到它的身影，如常见的 OpenStack、Kubernetes API 等，众多企业基于开放的 RESTful API 实现了业务扩展和资产获利；而陌生的是我们似乎很难找到 RESTful API 的准确定义，也不了解如何真正在实际项目中使用它。如 REST 和 SOAP 协议到底有何区别？RESTful API 和 HTTP 协议到底存在什么关系？REST 的安全性如何保证？令人欣慰的是，拜读 Sanjay Patni 先生这本著作后，这些问题都将迎刃而解。

REST 一词是由 Roy Fielding 博士于 2000 年在他的博士论文 *Architectural Styles and the Design of Network-based Software Architecture* 中提出的，实际指一种有助于创建和组织分布式系统的架构风格。它并不是一个标准或准则，而是一种基于资源的架构风格。基于 REST 风格构建的 API 应该满足 CS 模式交互、统一的资源接口、透明的分层系统以及支持无状态和缓存等条件约束，另外需要保证 API 的安全性等。以 REST 风格构建的系统将在性能、可扩展性、可移植性、可靠性等多个方面得到提升和优化。

本书作者 Sanjay Patni 拥有 15 年以上的企业级软件开发经验，尤其在构建 RESTful API 方面有深厚的理论研究和技術实践背景。本书主要

从 RESTful API 的架构、设计和编码三个方面进行深入介绍。首先阐述 REST 的基本原理和准确定义，并对 REST 与 SOAP 协议的差异，以及 XML/JSON 等数据交换格式等进行全方位比较。其次在 API 设计和建模方面讨论 API 的完整生命周期和 RESTful API 设计的最佳实践，着重介绍 API 组合和框架如何实现 API 的一致性和可重用性，以及 API 平台管理、API 安全性和缓存机制等。另外通过一个实际案例(播客 podcast)演示如何通过 RAML 建模工具实现 API 接口的完整定义，并基于 JAX-RS 规范实际编写了一个 REST 应用，包括 API 的外观层、数据访问层以及服务层的实现等，为我们提供了可应用于企业实际场景的标准应用示范，使得 RESTful API 不再是虚无的概念描述，而是可真正用于企业实践的架构风格实现。

书中很多内容都给译者留下了深刻印象，首先在构建 RESTful API 的基础 URL 时，我们需要从原有的“动词+名词”的设计向“名词+HTTP 动词”的观念转变，使得基础 URL 简单而又直观。其次在 API 组合的管理中，我们需要解决稳定高效的 API 服务提供与企业创新(和实验)之间的矛盾，因此我们需要通盘考虑 API 的一致性、可重用、版本和变更管理等。此外，企业级应用需要关注 JavaScript 跨域访问解决方案、OAuth 2 协议的安全性保证以及多种缓存机制等。

如果希望真正透彻掌握 RESTful API 的设计理念和实际应用，译者建议大家主动完成每章的程序练习，代码其实是最好的老师。每章最后一节都包含详细丰富的环境设置和代码，使得我们更容易理解和掌握 RESTful API 的精髓。

最后要感谢清华大学出版社能给我这次宝贵的翻译机会，把自己关于 RESTful API 的一些学习和理解在本书中与大家分享。尤其是要感谢本书的编辑，在本书的翻译过程中付出了很多的心血和努力，非常感谢他们的帮助和鼓励。另外感谢妻子以及我的家人，感谢你们一直以来对我的包容和理解；本书也献给我未来的孩子，希望你会喜欢老爸的这份礼物。

翻译英文书籍涉及多个概念的中文释义，我每次都力求与维基百

科、搜狗百科、百度百科等保持一致。鉴于译者水平有限，错误或瑕疵在所难免，如果对书中有任何意见或建议，也欢迎大家批评指正，我将感激不尽！本书全部章节由郭理勇翻译并整理，参与翻译的还有杨小琼、贺珊珊、王永胜等。

希望这本书能有力推动业界对 RESTful API 的统一定义和应用，更好地构建统一、高效、可扩展的分布式系统应用。

## 作者简介



**Sanjay Patni** 是一位注重实际成果的技术专家，在创新技术方案与业务实际需求的协调上具有丰富的经验，长期致力于企业业务流程的优化和运营效率的提升。

在过去五年中，他一直在 Oracle 公司的 Fusion Apps 产品研发团队任职，在那里他发现了对 Fusion Apps 代码管理实现自动化的机会，其中不仅涉及 GA 版本的交付发行，还包括正在进行的演示、开发和测试代码。他提出并开发了自助服务 UX 用于代码请求和审核，减少了 80% 的手工步骤。他还发起了 12 次代码快速迭代，通过使用工作流和 RESTful API 等自动化技术与其他子系统进行集成，使得大约 100 多个手工步骤实现了自动化。

在加盟 Oracle 前，他已经在软件行业工作了 15 年以上，为不同的行业提供关键技术解决方案。他的职责包括对基于 Web 的企业级产品和解决方案提供技术创新、需求理解和分析，技术架构设计，以及推进软件敏捷开发等。他率先创新使用 Java 来构建业务应用，不断推动和完善用于企业级业务应用构建的 Java API，并获得 Sun Microsystems 公司颁发的奖项。

Sanjay 曾担任 RESTful API 设计和集成培训或课程的客座讲师、技术导师。他拥有强大的计算机科学教育背景，硕士毕业于印度理工学院(IIT)。

# 技术审稿人简介



Massimo Nardone 拥有超过 22 年的安全、Web/移动开发、云计算和 IT 设施等领域的丰富经验，对网络安全和 Android 有着狂热的技术激情。

在过去 20 多年，他一直致力于编程开发和教学，包括 Android、Perl、PHP、Java、VB、Python、C/C++、MySQL 等，拥有意大利萨莱诺大学计算机科学专业的硕士学位。

他曾经担任项目经理、软件工程师、研究员、首席安全架构师以及信息安全经理等，同时作为 PCI(国际安全标准)/SCADA(数据采集与监视控制系统)审计员，还是 IT 安全/云/SCADA 高级架构师等。

他的技术栈包括安全、Android、Cloud、Java、MySQL、Drupal、Cobol、Perl、Web 和移动开发、MongoDB、D3、Joomla、Couchbase、C/C++、WebGL、Python、Pro Rails、Django CMS、Jekyll、Scratch 等。

他目前担任 Cargotec Oyj 公司的首席信息安全官，曾任赫尔辛基理工大学(Aalto University)网络实验室的访问讲师和主管，拥有四项国际专利(PKI、SIP、SAML 和 Proxy 领域)。

Massimo 已为不同的出版公司审阅了 40 多种 IT 类图书，同时是 *Pro Android Games* 的作者之一(Apress, 2015)



# 前 言

众所周知，数据库、网站以及业务应用之间都需要数据交换。这通过定义标准的数据格式、传输协议或 Web 服务来实现，常见的数据格式有 XML(Extensible Markup Language, 可扩展标记语言)、JSON(JavaScript Object Notation, JavaScript 对象表示法)等，常见的传输协议或 Web 服务包括 SOAP(Simple Object Access Protocol, 简单对象访问协议)，以及目前更受欢迎的 REST(Representational State Transfer, 表述性状态传递)等。开发人员通常需要设计自身应用的 API 接口，使得应用能集成特定的业务逻辑并运行在操作系统或服务器上。本书涵盖以上数据交换概念和通用的数据格式，并重点阐述如何构建 REST 风格的 API。

对于 Web 系统的交换来说，你将学习 HTTP 协议，包括如何使用 XML。另外本书还比较了 SOAP 和 REST，介绍无状态转移的概念。同时介绍软件 API 设计和最佳实践等。本书后半部分将重点讨论遵循 JAX-RS 标准的 RESTful API 的设计和实现，以及通过 Java API 构建 RESTful Web 服务。你将学习如何使用 JSON 和 XML 构建和使用 JAX-RS 服务，并通过实际案例使用 RESTful API 将众多不同的数据源集成在一起(包括关系型数据库和 NoSQL 数据库等)。你将应用这些最佳实践完成一个小型软件系统 API 的设计与实现，并以 RESTful API 的方式公开可用的 API 服务。

本书适用于那些在实际项目中需要使用数据交换的软件开发人员，对那

些希望了解数据交换方法以及如何与业务应用交互的数据专家同样有所帮助。书中的案例练习要求读者具有 Java 编程经验。

本书的主题包括：

- 数据交换和 Web 服务
- SOAP 与 REST，有状态与无状态
- XML 与 JSON
- API 设计简介：REST 和 JAX-RS
- API 设计实践
- 设计 RESTful API
- 构建 RESTful API
- 与 RDBMS(MySQL)进行交互
- 使用 RESTful API(比如 JSON、XML)
- API 安全性-OAuth
- API 缓存

### 源代码下载

读者可访问 [www.apress.com/9781484226643](http://www.apress.com/9781484226643) 下载源代码，也可扫描本书封底的二维码直接下载。

# 目 录

第 1 章 RESTful API 的基本原理 .....	1
1.1 SOAP 和 REST 的比较 .....	3
1.2 Web 架构风格 .....	4
1.2.1 CS 模式 .....	5
1.2.2 统一资源接口 .....	5
1.2.3 分层系统 .....	5
1.2.4 缓存机制 .....	6
1.2.5 无状态 .....	6
1.2.6 按需编码 .....	6
1.2.7 HATEOAS .....	6
1.3 安全性 .....	7
1.4 什么是 REST? .....	8
1.4.1 REST 基础知识 .....	8
1.4.2 REST 基本原理 .....	9
1.5 小结 .....	10
第 2 章 API 设计和建模 .....	11
2.1 API 设计策略 .....	11
2.2 API 创建流程和方法论 .....	13

2.2.1	流程	13
2.2.2	API 方法论	14
2.2.3	域分析或 API 描述	14
2.2.4	架构设计	15
2.2.5	原型设计	16
2.2.6	实现	16
2.2.7	发布	16
2.2.8	API 建模	16
2.2.9	API 建模的比较	18
2.3	最佳实践	19
2.3.1	保持基础 URL 简明直观	19
2.3.2	错误处理	20
2.3.3	版本控制	22
2.3.4	局部响应	23
2.3.5	分页	23
2.3.6	多格式	24
2.3.7	API Façade	24
2.4	API 解决方案架构	24
2.4.1	移动解决方案	25
2.4.2	云端解决方案	25
2.4.3	Web 端解决方案	26
2.4.4	集成解决方案	26
2.4.5	多终端解决方案	26
2.4.6	智能电视解决方案	26
2.4.7	物联网	26
2.5	API 解决方案中的利益相关者	26
2.5.1	API 提供者	27
2.5.2	API 消费者	27
2.5.3	最终用户	27
2.6	小结	33

第 3 章 XML 与 JSON 介绍 .....	35
3.1 XML 简介 .....	35
3.1.1 XML 注释 .....	36
3.1.2 XML 的重要性 .....	37
3.1.3 如何使用 XML .....	38
3.1.4 XML 的优缺点 .....	38
3.2 JSON 简介 .....	38
3.2.1 JSON 语法 .....	39
3.2.2 JSON 的重要性 .....	40
3.2.3 如何使用 JSON .....	41
3.2.4 JSON 的优缺点 .....	42
3.3 XML 和 JSON 的比较 .....	42
第 4 章 JAX-RS 介绍 .....	51
4.1 JAX-RS 简介 .....	51
4.1.1 输入和输出内容类型 .....	52
4.1.2 JAX-RS 注入 .....	53
4.2 REST 实现 .....	55
第 5 章 API 组合和框架 .....	65
5.1 API 组合架构 .....	65
5.1.1 需求 .....	65
5.1.2 一致性 .....	65
5.1.3 可重用 .....	66
5.1.4 可定制 .....	66
5.1.5 可发现 .....	66
5.1.6 持久性 .....	66
5.2 如何实施这些需求——治理? .....	67
5.2.1 一致性 .....	67
5.2.2 可重用 .....	67
5.2.3 可定制 .....	67

5.2.4	可发现	68
5.2.5	变更管理	68
5.3	API 框架	68
5.3.1	流程 API——服务层	69
5.3.2	系统 API-数据访问对象	69
5.3.3	体验 API-API 外观	70
5.3.4	服务层实现	70
第 6 章	API 平台和数据处理器	81
6.1	API 平台架构	81
6.1.1	我们为什么需要 API 平台	81
6.1.2	什么是 API 平台	82
6.1.3	API 平台需要具备的功能	82
6.1.4	API 平台是如何组织的, 什么是 API 平台的架构	84
6.1.5	API 架构如何适应围绕企业的技术架构	85
6.2	数据处理器	86
6.2.1	数据访问对象(DAO)	86
6.2.2	命令查询职责分离(CQRS)	86
6.3	小结	101
第 7 章	API 管理和 API 客户端	103
7.1	外观	103
7.1.1	外观模式	103
7.1.2	API 外观	104
7.2	API 管理	105
7.2.1	API 生命周期	106
7.2.2	API 下线	107
7.2.3	API 盈利	108
第 8 章	API 安全性与缓存机制	115
8.1	API 安全性-OAuth 2	115

8.1.1	角色	116
8.1.2	令牌	116
8.1.3	注册成客户端	117
8.1.4	授权授予类型	118
8.1.5	隐式授予流程	119
8.1.6	资源拥有者密码凭据授予	121
8.1.7	客户端凭据授予	122
8.2	缓存机制	123
8.2.1	服务器缓存机制	124
8.2.2	HTTP 缓存机制	124
8.2.3	Web 缓存机制	126
8.3	小结	129

## 第 1 章



# RESTful API 的基本原理

API 已经不是新兴事物了。近几十年间，API 一直作为接口使得应用之间可以相互通信。但 API 的作用在过去几年中发生了巨大变化。越来越多的创新型公司发现，通过为业务伙伴提供 API 接口可利用数字资产获利，此外，还可通过为合作伙伴提供更多功能来扩展价值主张，以及通过多种终端和设备来实现与客户的连接。创建 API 意味着：允许所在组织内或组织外的其他人，利用你的服务或产品来创建新应用、吸引客户或拓展业务。

其中内部 API 可通过在新应用中最大化重用性和增强一致性，来提升开发团队的生产效率。公共 API 可通过允许第三方开发人员增强你的服务或带来新客户从而使你的业务增值。一旦开发人员能利用你的服务和数据开发出新应用，就会形成一种网络效应，从而对底层业务产生显著影响。例如，Expedia 通过 API 向合作伙伴开放旅行预订服务来建立 Expedia 联盟网络，给公司带来了新的收入增长点，目前每年能产生 20 亿美元的收益。再如，Salesforce 通过发布 API 使得合作伙伴能够扩展其平台的功能，这些基于 SOAP(JAX-WS)以及最近的 RESTful(JAX-RS)的 API 收益已经贡献了公司年收入的半壁江山。

最初使用的 SOAP Web 服务依赖许多技术(如 UDDI、WSDL、SOAP、HTTP)和协议，用于在服务提供者和消费者之间传输和转换数据。可使用 JAX-WS 创建 SOAP Web 服务。



后来, Roy Fielding 于 2000 年发表了他的博士论文 *Architectural Styles and the Design of Network-based Software Architecture*。他创造了 REST 一词, 一种分布式超媒体系统的架构风格。简而言之, REST(表述性状态传递, Representational State Transfer 的缩写)是一种有助于创建和组织分布式系统的架构风格。这个定义中的关键词应该是“风格”, 因为 REST 很重要的一个方面(也是撰写本书的一个主要原因), 在于 REST 是一种架构风格, 而不是一个准则、一个标准, 也不是为最终形成 RESTful 架构而需要遵循的一系列硬性规则。

本章详细介绍 REST 基本原理、SOAP 与 REST 的比较以及 Web 架构风格。

总之, 以 REST 风格组织的分布式系统将在以下几个方面得到改进:

- **性能:** REST 提出的通信方式是高效简单的, 可在采用它的系统上实现性能提升;
  - **组件交互的可扩展性:** 任何分布式系统都应能很好地处理这方面的问题, 而 REST 提出的简单交互极大地满足了这一点;
  - **接口的简单性:** 简单接口允许简化系统之间的交互, 反过来又可提供上述优点;
  - **组件的可变性:** 系统的分布性质以及 REST 提出的关注点分离概念(稍后再次讨论), 可使组件以最低成本和风险彼此独立地进行修改;
  - **可移植性:** REST 与技术和语言无关, 这意味着可通过任何类型的技术来实现和使用它(存在一些限制, 稍后将谈到, 但不涉及具体技术);
  - **可靠性:** REST 提出的无状态(stateless)约束(见稍后的讨论)使得系统在发生故障后更容易恢复;
  - **可见性:** 重述一次, REST 提出的无状态(stateless)约束增加了所述请求的完整状态(稍后会谈到这些约束, 到时大家就明白了)。
- 从这个列表中可推断出 REST 的一些直接优点。以组件为中心的设计使得系统容错性非常好。单个组件的故障不影响系统的整体稳定性, 这对于任何系统来说都是一个很大的优点。另外