HZ BOOKS
华章科技

畅销书全新优化和升级，阿里云大数据产品架构师/Java技术专家撰写

深刻解读JAX-RS的标准和API设计；Jersey的使用要点和实现原理，以及基于REST的Web服务的设计思想和原则

Java
核心技术
系列

Java RESTful Web Service实战

（第2版）

Java RESTful Web Services in Action
Second Edition

韩陆 著

机械工业出版社
China Machine Press

Java
核心技术
系列

Java RESTful Web Service实战

（第2版）

Java RESTful Web Services in Action
Second Edition

韩陆 著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Java RESTful Web Service 实战 / 韩陆著 . —2 版 . —北京：机械工业出版社，2016.7
（Java 核心技术系列）

ISBN 978-7-111-54213-1

Ⅰ. J… Ⅱ. 韩… Ⅲ. JAVA 语言 – 程序设计 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2016）第 156331 号

Java RESTful Web Service 实战（第 2 版）

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：李 艺 责任校对：董纪丽
印 刷：北京市荣盛彩色印刷有限公司 版 次：2016 年 8 月第 2 版第 1 次印刷
开 本：186mm × 240mm 1/16 印 张：18.75
书 号：ISBN 978-7-111-54213-1 定 价：59.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88379426 88361066 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzit@hzbook.com

Foreword 第 2 版序一

韩陆是我在阿里巴巴的同事，业余时间大家经常一起聊新的 Java 技术。REST 对当前软件开发非常重要，除了我们一直了解的 Service API、Open API、移动端对后端的 Gatway API 调用（这些基本都是 REST 模式设计的），现在很多的 DevOps 操作也是通过 REST API 完成的，如我们了解的 Docker 和 SpringBoot Actuator API 都是 REST 风格的，另外 HTTP/2 的逐步采用，也为 REST 带来更多的功能和性能的提升。对 Java 程序员来说，本书非常难得，你可以深入了解 JAX-RS 标准和 Jersey 框架；为了方便落地开发，书中更着重介绍了 Spring Boot 和 Spring Cloud，这些知识目前涉及的中文图书并不多；最后结合 Docker 容器技术，给出了完整基于 SpringBoot REST 服务应用容器部署的思路。本书的每一个技术点都可以单独成书，用以详细阐述，但是能够浓缩到一本图书中，挑战和难度确实比较大，希望这些新的技术和思想能够帮助到真正前进的程序员。

阿里巴巴资深技术专家、速卖通中间件掌门人 陈立兵（花名：雷卷）

第 2 版序二 *Forword*

认识韩陆已有近 10 年的时间，那时他在北京航空航天大学软件学院做硕士毕业论文。他的论文写的是面向对象数据库引擎的设计与实现，完全自主实现了包括文件读写、缓存、索引和事务、数据访问接口等完整的面向对象数据库引擎。从那时就了解到他是一位技术达人，喜欢钻研和实践各种最新的技术。所以当听说他出版本书时一点都不觉得意外，他就是这样一个热衷于新技术的人。

早在本书第 1 版的时候，他就找到我希望为本书写一篇序，那时我婉拒了，因为我本人对 RESTful 相关技术并没有太多的接触，不敢贸然推荐。作为一种轻量级 Web 服务实现架构，两年多来 RESTful 架构得到了普遍认可和使用；越来越多的学生也开始学习相关的技术，而本书就是这方面非常有意义的参考资料。书中首先系统地解读了 JAX-RS2 标准，之后基于 JAX-RS 标准的参考实现：Jersey 开发框架，系统地讲解了如何基于该框架开展 RESTful Web 服务的实践。本书实践性很强，体系较为完整，涵盖了 RESTful Web 服务开发各个层面的问题；书中不仅提供了一些典型场景的代码示例，还有完整的项目案例的讲解，这些实践代码能够有助于读者开展具体的项目实践。与第 1 版相比，第 2 版还新增了有关微服务和容器化等目前热门应用技术实践的内容，有助于读者了解最新的技术发展方向。

北京航空航天大学　谭火彬

Foreword 第 1 版序一

——REST 开发的理想与现实

REST 是一种分布式应用的架构风格，也是一种大流量分布式应用的设计方法论。REST 是由（构成了 Web 基础架构的）HTTP、URI 等规范的主要设计者 Roy Fileding 博士在其 2000 年的博士论文（中文版名为《架构风格与基于网络应用软件的架构设计》）中提出的。到目前为止，关于 REST 最系统、最全面的论述，仍然是 Fielding 的博士论文。

REST 就是 Web（World Wide Web，简称 Web 或者 WWW）本身的架构风格，是设计、开发 Web 相关规范、Web 应用、Web 服务的指导原则。不符合 REST 风格要求的架构和技术，很难在 Web 这个生态系统中得到繁荣发展。在我看来，Roy Fielding 博士就是 15 年以来对于分布式应用架构设计理论贡献最大的人。Fielding 在 HTTP 规范的设计过程中，并没有采用当时大行其道的 DO（Distributed Object，分布式对象）风格，而是自出机杼、另辟蹊径，提出了一整套新的设计方法论。Fielding 的开创性工作，极大地推动了分布式应用设计理论的发展。

有趣的是，其实基于 SOAP/WSDL 的“大 Web Service”（以下简称 Web Service），几乎是与 REST 同时发展起来的。虽然在 Web Service 中也使用了对象，但是 Web Service 其实是 RPC 风格的，而不是 DO 风格的。Web Service 在最初几年发展很快，很大原因是它解决了 DO 风格难以解决的异构系统（不同的硬件系统、不同操作系统、不同的编程语言，等等）之间互操作性的问题。

然而遗憾的是，设计 Web Service 协议栈的核心人员，几乎都是来自于企业应用阵营的，尤其是来自于 IBM 和微软两家公司的人。这些企业应用的专家们没有充分认识到 Web 基础架构的巨大优点，甚至可以说并没有理解 HTTP 协议究竟是用来做什么的、为何要如此设计。在 Web Service 协议栈的设计之中，仍然有深深的企业应用痕迹。Web Service 虽然

宣称能够很好地支持互操作，然而因为协议栈的复杂性很高，在实战中互操作性并不好（例如升级过程困难而且复杂）。此外，Web Service 仅仅将 HTTP 协议当做一种传输协议来使用，还依赖 XML 这种冗余度很高的文本格式，这导致 Web Service 应用性能低下。很多开发团队宁可使用 Hessian 等轻量级的 RPC 协议，也不愿意使用 Web Service。在面向互联网的大流量 Web 应用（包括 Web 服务在内）这种运行环境中，Web Service 在复杂性、互操作性、性能、可伸缩性等方面的短板更加突出。因此，设计今日面向互联网的 API，已经很少有人会考虑 Web Service。这使得 Web Service 的使用被局限在企业应用运行环境之中，其名称中的“ Web”更像是一个笑话（除了都使用 HTTP 协议，基本上与 Web 没什么关系）。假如在 2000 年，设计 Web Service 规范的专家们，能够认真读一下 Fielding 的博士论文，或者找 HTTP、URI 等 Web 基础架构规范的核心设计人员深入交流一下，Web Service 很可能就不是现在这个样子了。不过，历史是无法假设的。

在 Java 世界中，与大 Web Service 相对应的规范是 JAX-WS。在大 Web Service 已经成为明日黄花之后，Java 世界急需一套新的规范来取代 JAX-WS。这套新的规范就是 JAX-RS：Java 世界开发 RESTful Web Service（与 RESTful API 含义相同，可混用）的规范。虽然起步很晚，毕竟走上了正确的道路。

从 Java EE 6 开始，JAX-RS 在 Java EE 版图中，作为最重要的组成部分之一，逐步取代了 JAX-WS 的地位。在所有 Java EE 相关规范中，JAX-RS 是优点很突出的一个。例如，完全基于 POJO、很容易做单元测试、将 HTTP 作为一种应用协议而不是可替代的传输协议（因此提高了性能）、优秀的 IDE 集成，等等。可以说，在大多数场合，JAX-RS 完全可以取代 JAX-WS，作为 Java Web Service 开发的主要技术。JAX-RS 同样也可以完全取代 Hessian 等基于 HTTP 协议的 RPC 风格远程调用协议。毕竟 HTTP 本身就是一种 REST 风格的应用协议，以 REST 风格来使用 HTTP，才是最高效的使用方式。

Jersey、CXF 等支持 JAX-RS 规范的 REST 开发框架还支持输出 WADL。WADL 支持客户端代码自动生成，还可以将 WADL 导入到 SoapUI 等测试工具中，然后做自动化集成测试。从开发 Java 企业应用、取代 JAX-WS 的角度来看，JAX-RS 已经做得非常棒了。

尽管如此，不可不提的是，JAX-RS 这套规范，仍然存在着很多遗憾。需要特别指出的是，JAX-RS 规范并不等于 REST 架构风格本身，REST 的内涵要比 JAX-RS 广泛得多。学会了使用 JAX-RS 了，并不等于就完全理解了 REST，开发者仍然需要下工夫认真学习一下本源的 REST 究竟是什么。

例如，JAX-RS 规范对于应该如何定义一个资源，以及应该如何使用 HTTP 作为一个统一接口来操作资源，显然缺乏必要的指导。有很多开发者只是简单地将以前 JAX-WS 中的一个 endpoint 接口转换成一个资源接口。接口的方法太多，导致映射到的 HTTP 方法不够

用，这也难不倒他们，在 URI 路径中加一些字符串就能够解决（例如，三个接口方法都映射到 POST，但是其 PATH 不同）。这样的开发方式非常常见，虽然开发者使用了 JAX-RS 规范，但是开发方式完全是 RPC 风格的，可以说与 REST 风格没有任何关系。

此外，JAX-RS 规范目前尚不支持 HATEOAS（将超文本作为应用状态的引擎，REST 风格的核心特征之一），从著名的 Richardson 成熟度模型（由《 RESTful Web APIs 》的作者 Richardson 提出）来衡量，基于 JAX-RS 规范实现的 RESTful API 仅仅能够达到成熟度模型的第二级，即支持资源抽象、统一接口的“CRUD 式 Web 服务”。

可以这样说，JAX-RS 规范与真正的 REST 风格，覆盖的范围其实是不同的。JAX-RS 覆盖的是简单基于 HTTP 协议（没有使用 SOAP/WSDL）的各种远程调用需求，很多需求对于可伸缩性、松耦合的要求并不高，仅仅是希望使用 HTTP 本身来取代大 Web Service 作为一种轻量级、容易测试的远程调用协议。REST 架构风格的严格要求，在这些场合并不是非常重要。慵懒是人类的天性，大多数开发者写代码只是为了解决手头的问题，JAX-WS 并不好用，JAX-RS 解救了他们。

如果按照 Roy Fielding 博士的严格要求（REST APIs must be hyper-text driven），那么包括 JAX-RS 规范在内都不能算是真正的 RESTful。然而，从实战角度，我认为革命不分先后，只要能够达到 Richardson 成熟度模型第一级，即有清晰的资源抽象，就可以认为是 RESTful API 了。如果连第一级都达不到，所设计的架构根本就不是面向资源的，那八成还是 RPC 风格的，就没有必要非要往 RESTful API 阵营里面挤了。从来没有人说过 RPC 就是万恶的，RPC 在企业应用的大多数场合其实都非常有效，只是不适合面向互联网的大流量 Web 应用而已。

因此，能够完美支持 HATEOAS，攀登到成熟度模型第三级，是一种理想情况（当然也是值得追求的）。而通过部分拥抱 REST 风格的要求，来更好地解决手头的问题，是更多开发者所面对的现实情况。JAX-RS 反映的正是这种现实情况，从实战的角度，它是一套非常有用也很好用的规范。

韩陆兄的新著《 Java RESTful Web Service 实战》是 JAX-RS 规范方面的专著，也是国内第一本 REST 开发的原创著作。这本书的实战性非常强，全面介绍了 JAX-RS 2.0 的方方面面，可以作为一线 Java 分布式应用开发者的案头必备书。如同我在前面所指出的，JAX-RS 规范并不等于 REST 架构风格本身，它们有着不同的覆盖范围。在本书中，作者也介绍了很多设计 RESTful API 的最佳实践，这些内容假如读者不理解 REST，甚至在亲自阅读了 JAX-RS 规范之后也未必能够总结出来。读者在阅读本书的过程中，不应该仅仅满足于掌握了 JAX-RS 开发的基本方法、解决了手头的问题、用其完全取代 JAX-WS，更重要的是，读者还应该就 REST 架构风格本身做更多的学习。幸运的是，除了本书之外，目前 REST 设计

和开发方面的图书资料已经非常多了。

本书的内容非常严谨，有非常好的系统性，对于设计开发大流量 Web 服务会面临的各种问题都有涉及。特别是在自动化测试方面着墨颇多，在我看来是本书的一大亮点。RESTful API 的自动化测试非常重要，需要在设计之初就充分考虑到。本书是一本难得的原创佳作，值得所有 Java 分布式应用的开发者购买。

理想富丽丰满，现实贫瘠骨感，追求理想和注重解决现实问题其实并不矛盾。JAX-RS 规范的发展，反映出了 Java 社区在更好地开发 RESTful Web Service 方面的求索。尽管存在争议，在我看来，规范化是推动 RESTful Web Service 取得更大发展的必由之路。目前对于优秀的 RESTful API 有哪些判断标准，Web 开发者社区已经达成了高度共识，也积累了大量非常有价值的成果。JAX-RS 规范的发展，离不开 Web 开发者社区的这些成果。在未来的 JAX-RS 3.0 规范中，我们将会看到更多令人兴奋的成果被规范化。JAX-RS 2.0 已经做得不错了，但是在 RESTful Web Service 规范化的道路上，其实才刚刚起步，任重而道远。

李锟　于上海

Forword 第 1 版序二

半年前初识韩陆的时候，我们就聊到他正在写的这本书，当得知我从 2006 年就参与了 Apache CXF 开发，他立即邀请我为他的新书写序，我也就欣然答应了。

Apache CXF 作为 JAXWS 以及 JAX-RS 规范的实现框架，已经成为很多 Web 服务开发者必选的开发框架。作为这一框架的开发维护者之一，我的日常工作经常需要熟悉这些 JSR 规范，并实现 JSR 所定义的 API，解决最终用户的使用问题。

熟悉 Java 的人大多都听说过 JSR（Java Specification Requests）、JCP（Java Community Process），通过 JSR 可以就 Java 某一方面的应用定义一组标准的 API 或者服务。对于最终用户来说，他们的代码只需要调用 JSR 定义的标准 API，不做任何修改就可以调用不同的 JSR 实现。这里常见的例子就是 Java Servlet 应用，用户开发的 Web 应用可以不做任何修改就部署到 Tomcat、JBoss 等不同的 Web 容器中。

JAXRS 是 JCP 为 Java RESTful Web Service 定义的一套 API。由于 Web 服务的描述模型与 Java 类和接口有一定的差距，JAX-RS 定义了很多 annotation，通过这些 annotation 我们可以很方便地将 Java 类描述成为相关的 REST 服务。由于 RESTful Web Service 通常需要部署到 Web 容器中，JAX-RS 也定义了相关服务来发现部署到容器中的 JAX-RS 应用。

读过 JSR 规范的朋友或多或少都会有这样的体会，JSR 作为规范文档，其目标是将 API 定义以及实现功能描述清楚、完备，其目标读者是相关 API 的实现人员，或者是相关 API 的高级使用人员。如果读者对相关的背景知识还不熟悉的话，JSR 文档读起来会比较晦涩而且难以理解。加之绝大部分 JSR 文档都没有相关的中文翻译，对于绝大多数初学者来说，通过阅读 JSR 文档来学习相关的 API 的知识是一个艰难的过程。

如果我们想要对 JAX-RS 规范有一个比较快速并且全面的了解应该怎么办呢？一般来我们可以通过 JSR 的相关参考实现入手，我们不但可以通过运行相关的参考实现的例子快速入门，还可以通过跟踪相关的代码对实现细节有一个全面的了解。韩陆的这本新作以 JAX-

RS 的参考实现 Jersey 为蓝本，由浅入深地向大家介绍了 JAX-RS 的由来，以及与 RESTful Web 服务开发的相关 API，并结合实例分享了作者的实战经验。

好了，现在打开你熟悉的 IDE 工具，加载 Jersey 代码库，沿着本书的指引去探索 Java RESTful Web Services 开发世界吧。

RedHat 姜宁

Preface 前　言

本书第 1 版发行后，Jersey 版本从 2.9 更新到了 2.22.2，此间 REST 服务得到了更广泛的认可和使用。与此同时，Java 8、Spring Boot 和 Docker 的爆发式发展，使得 Java 领域的 RESTful 开发有了新的发展。

第 2 版变更

迫不及待，这是我想为读者更新 REST 服务新发展的心情，遂有此第 2 版。首先，我们要拥抱 Java 8。lambda 表达式在大数据处理，尤其在 Spark 中是默认的语法表达；Java 8 带给我们的不只是“语法糖”，而是开发和执行效率的提升。我从实践中得到了其中的好处，也希望读者能跟上时代的步伐。其次是 Spring Boot，这是 Java 领域实现微服务的事实标准框架。我已经无法回去适应部署 war 到 Tomcat 的时代，请保守的读者原谅我的情不自禁。再次是 Docker，我希望读者具备使用 Docker 完成开发自测阶段的一切，也希望读者能运用 Docker 实现微服务的部署和可伸缩实践。

从第 1 版第 1 次印刷至今，我始终关注着读者的反馈。邮件都做了认真的回复。根据读者的反馈，我在第 2 版中重新梳理了章节的结构，删除了第 1 版中反馈不好的第 9 章和第 11 章，调整后的章节与第 1 版的对应关系如下。

- 第 1 章合并了第 1 版的第 1 章和第 2 章。
- 第 2 章对应第 1 版第 3 章。
- 第 3 章对应第 1 版第 4 章。
- 第 4 章包含了第 1 版的第 8 章。
- 第 5 章在第 1 版的基础上做了更新。
- 第 6 章包含了第 1 版的第 7 章，并升级了第 1 版 2.5 节的示例。
- 第 7 章和第 8 章是新增章节。

❑ 第 9 章对应第 1 版第 10 章。

❑ 第 10 章包含了第 1 版的第 6 章。

与许多技术作者一样，写书的时间是挤出来的。如果精力尚可，每晚 7 点到 9 点、11 点到凌晨 2 点是我动笔的时间，偶尔，早上 6 点到 8 点我也会赶赶。写书成为我梳理、总结和思考的最佳方式。

于此过程，我总结了 3 句话与读者共享。搞技术的人，是停不下来的。时而要开疆拓土，学习和研究新的知识点，弥补自己的技术债；时而要运筹帷幄，将知识点梳理成线，编织成网；时而要深耕细作，面对当下要攻坚的业务所对应的知识点，深入研究、反复实践、勤于思考、勇于交流。只有这样，我们才可以坦然地用手推一下眼镜，谦虚地告诉别人，“其实我是个程序员”。

源代码

本书提供源代码下载，地址是 https://github.com/feuyeux/jax-rs2-guide-II。

勘误和交流

本书的勘误会在 https://github.com/feuyeux/jax-rs2-guide-II/wiki 发布，欢迎读者批评指正。

❑ 我的邮箱：feuyeux@163.com

❑ 我的新浪微博：六爷 1_1

致谢

感谢我的妻子 Caroline 和女儿 Doris 一直以来的关心和陪伴。

感谢华章公司的杨福川对我的专业指导。感谢华章公司编辑高婧雅、李艺专业和耐心的审阅和指正。

感谢阿里巴巴速卖通中间件团队在微服务、容器化上对我的影响。感谢雷卷、许晓斌在 DDD、Spring Boot 和 Docker 上对我的帮助。感谢 Technicolor 的敏捷团队、阿里巴巴国际站测试架构团队，前者带我悟得 Jersey，后者给我深入实践的机会。

最后我要感谢阿里巴巴阿里云事业群大安全的各位兄弟对我的支持。我正在这里，与大家一天天、一步步将微服务和容器化落地生花。

Contents 目　录