

O'REILLY®



图灵程序设计丛书



全端Web开发

使用JavaScript与Java

Client-Server Web Apps with JavaScript and Java

[美] Casimir Saternos 著
王群锋 杜欢 译



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

全端Web开发 使用JavaScript与Java

Client-Server Web
Apps with JavaScript and Java

[美] Casimir Saternos 著

王群锋 杜欢 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权人民邮电出版社出版

人民邮电出版社

北 京

图书在版编目(CIP)数据

全端Web开发：使用JavaScript与Java / (美) 萨特诺斯(Saternos, C.) 著；王群锋，杜欢译. — 北京：人民邮电出版社，2015.8
(图灵程序设计丛书)
ISBN 978-7-115-39730-0

I. ①全… II. ①萨… ②王… ③杜… III. ①JAVA语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第144108号

内 容 提 要

JavaScript 和 Java 这两大生态系统之间如何协同，成为所有 Web 开发人员共同面临的问题。本书应运而生，全面又简练地为读者展示了最新的 C/S 应用开发范式。本书以 Java 和 JavaScript 这两种最流行的服务器与客户端开发环境为例，全面讲解了最新的 C/S 应用开发范式。作者不仅讲解了很多实用的 C/S 开发架构，还通过各种实例进一步强化了读者的认知。

这是一本写给 Java 程序员的完整的最新 C/S 应用开发范式的指南。

-
- ◆ 著 [美] Casimir Saternos
 - 译 王群锋 杜 欢
 - 责任编辑 毛倩倩
 - 执行编辑 毛倩倩 岳新欣
 - 责任印制 杨林杰
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 三河市海波印务有限公司印刷
 - ◆ 开本：800×1000 1/16
 - 印张：13.75
 - 字数：326千字 2015年8月第1版
 - 印数：1—3 500册 2015年8月河北第1次印刷
 - 著作权合同登记号 图字：01-2014-8385号
-

定价：59.00元

读者服务热线：(010)51095186转600 印装质量热线：(010)81055316

反盗版热线：(010)81055315

广告经营许可证：京崇工商广字第 0021 号

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

版权声明

© 2014 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2015. Authorized translation of the English edition, 2015 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版，2014。

简体中文版由人民邮电出版社出版，2015。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

O'Reilly Media, Inc.介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 Make 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创产业的革命性思想。作为技术人士获取信息的选择，O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar 博客有口皆碑。”

——Wired

“O'Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

——CRN

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去 Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

前言

“计算机科学领域只有两大难题：缓存失效和命名。”

——菲尔·卡尔顿

写一本关于缓存失效的书并不是难事，难的是如何给书取个合适的名字。本书的书名代表的是 Web 开发方方面面的变化，这些变化促成了一种新的 Web 应用设计方法的诞生。

诚然，Web 开发的许多方面都可以称为“新”的。不断升级的桌面浏览器、层出不穷的移动终端、一直在演变的编程语言、更快的处理器，还有越来越挑剔的用户以及他们不断增长的、对易用性和交互性的期望，都让开发者枕戈待旦。这些改变使开发者在为项目提供解决方案时必须持续创新。而其中的许多解决方案并不是孤立的，它们有更为深远的含义。

因此，为了响应这些创新，我选择了“客户端－服务器端”这一术语，它在许多方面都反映了 Web 开发中所发生的变化。时下流行的其他一些对现代化开发实践的描述不适用于我们的研究领域。这里所讲的 Web 应用开发是有关桌面浏览器的，另一个日渐相关的领域——移动应用开发，则并不包括在内。

人们使用术语单页面应用或单页面界面，来区分现代 Web 应用和早期的静态网页。这些术语准确地抓住了现代 Web 应用的特征，和早期应用相比，它们更动态，交互性更强。

当然，很多现代的动态网站并不是单页面应用，它们可能由多个页面组成。这些术语强调的是页面——一个应用的客户端。对于服务器端的开发，它们未作任何特别声明。还有与高度动态化的页面相关的各种 JavaScript 框架（Angular、Ember 和 Backbone 等），但是这些仍然是客户端的事。我希望本书包含的内容更广，不局限于前端的创新，还应涵盖服务器端的相关设计和 Web 服务信息传递。

这就是流行的 REST（Representational State Transfer）技术，它建议了一种 Web 服务消息传递的风格。但是 REST 的作者罗伊·T. 菲尔丁（Roy Fielding）只给出了有限的定义，在个

人博客中，他列出了自称为“RESTful APIs”、却常常违反 REST 风格的几个错误 (<http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>)。还有人在 StackOverflow 上发帖提问 JSON API 是否是真正的 REST 式 (<http://stackoverflow.com/questions/9055197/splitting-hairs-with-rest-does-a-standard-json-rest-api-violate-hateoas>)，因为 JSON 并不完全满足 REST 结构风格的约束。对于 REST 服务，有一种渐进式描述 (<http://martinfowler.com/articles/richardsonMaturityModel.html>)，即只有当 API 满足相应程度的约束时才可称为 REST 式。REST 将客户端 – 服务器端架构、动词的使用和 URL 的命名规范也作为其约束。

使用 JavaScript 客户端处理实用的“REST 式”API 产生的消息是开发方法中重要的一环，那么服务器端的情况是怎么样的？

Java 企业版 (JEE) 包含了 JAX-RS API (<http://docs.oracle.com/javaee/6/tutorial/doc/giepu.html>)，它是一种 Java 风格的 REST (但没有严格遵循 REST 规范)，并使用了 Jersey 作为一种参考实现予以展示。如果仅限于使用 JAX-RS 开发 Web 应用，则会忽略一些现有的框架和其他一些基于 JVM 的编程语言所提供的解决方案 (尤其擅长快速搭建原型)。

为一本书起一个简单明了、引人入胜的书名并不容易，多亏了 James Ward (www.jamesward.com)，他在 2012 年的 OSCON 大会上发表了一篇题为“Client-Server Web Applications with HTML5 and Java”的演讲。在这篇演讲中，他列举了一种正在变得流行的 Web 应用开发方法的优点，在近些年的一些项目中，我也使用了这种方法开发 Web 应用。其中，术语“客户端 – 服务器端”是理解这种开发方法的关键，它概括了软件架构的根本性改变，对客户端和服务端做出了明确区分，并将两者放在了同样重要的位置上。

Web 应用的这种客户端 – 服务器端架构带来了程序员工作方式的转变 (有时候，甚至是天翻地覆的变化)。本书就是为了帮助开发者适应这种革命而写的，特别是对于如何构建现代 Web 应用的最新方法方面，也做出了恰当的评价。

目标读者

本书的目标读者是那些熟悉 Java、HTML、JavaScript 和 CSS 的 Web 应用开发者。它是为那些喜欢在实践中学习，喜欢将新技术集成进标准工具中构建示例应用的读者而量身定制的。如果你了解 JavaScript 的最新发展，以及和使用 Java 在开发流程上的异同，那么请阅读本书。

阅读本书，你需要在两方面之间做平衡。一方面，读者从本书中能得到的最大收获是从宏观上掌握技术转变的影响和趋势；另一方面，理解技术的最好途径往往是具体的例子。如果你的兴趣在于从宏观上掌握各种技术之间是如何融合在一起，那么定能从本书中受益。

我写作此书的目的，就是帮助读者做出明智的决定。好的决定可以帮助你在新项目中采用正确的技术，避免因采用不相容的技术造成的陷阱，或者对给出的决定形成错误的预期。

它还能帮助你在正开发的项目里快速上手，更好地支持已有代码。简言之，明智的决定可以让程序员的工作效率更高，它能让你在当前，或者是未来工作中研究感兴趣的领域时高效地利用时间。

本书内容

第 1 章概览了客户端 – 服务器端的 Web 应用架构，介绍了 Web 应用开发的历史，并为 Web 开发范式的迁移提供了充足的证据。由此引出了接下来的三章内容，分别描述了开发过程中各种工具的使用。

第 2 章介绍了 JavaScript 语言，以及使用 JavaScript 进行开发时用到的工具。

第 3 章介绍了如何设计 Web API、REST，以及开发基于 HTTP 的 REST 式应用时用到的工具。

第 4 章介绍了 Java，以及本书剩余章节内容需要用到软件。

本书的第二部分探讨了更高级的结构（比如客户端类库和应用服务器），以及它们如何做到良好的分离，从而方便快速开发。

第 5 章介绍了主流的客户端 JavaScript 框架。

第 6 章介绍了 Java API 服务器和服务。

第 7 章介绍了快速开发实践。

第 8 章深入讨论了 API 设计。

对这些类库和快速开发原型的流程有所了解之后，在接下来的几章中，我们将这些知识应用于具体的项目，使用各种基于 JVM 的语言和框架。在接下来的两章中，我们使用了轻量级的 Web 服务器和微框架，而没有采用传统的 Java Web 应用所需的打包方式和服务器。

第 9 章介绍了一个使用 jQuery 和 Jython 的项目。

第 10 章介绍了一个使用 JRuby 和 Angular 开发的项目。

本书的最后几章详细介绍了几个使用传统的 Java Web 应用服务器和类库的项目。

第 11 章介绍了 Java 生态系统中现有的 Web 应用打包方式和部署选项。

第 12 章介绍了虚拟化和在管理大规模服务器环境中涌现出的新技术。

第 13 章将焦点放在了测试和文档化上。

第 14 章对全书做总结，并对互联网相关技术和软件开发中各种纷繁的变化给出了看法。

附录 A 介绍了如何方便地与 Java 类交互。

排版约定

本书使用的排版约定如下所示。

- 楷体
表示新的术语。
- 等宽字体 (`Constant width`)
表示程序片段，也用于在正文中表示程序中使用的变量、方法名、命令行代码等元素。
- 等宽粗体 (**`Constant width bold`**)
表示示例中的新代码。
- 等宽斜体 (*`Constant width italic`*)
表示应该由用户输入的值或根据上下文确定的值替换的文本。



这个图标表示提示、建议或重要说明。



这个图标表示警告或提醒。

使用代码示例

你可以在这里下载本书随附的资料（项目、代码示例等）：<https://github.com/javajavascript/client-server-web-apps>。你可以在线查看，或者下载其压缩文件以在本地使用。其中资源是按章划分的。

本书示例代码仅作展示特定功能之用，不求全面展示一个功能完整的应用，具体区别如下。

产品级的系统要对数据类型、验证规则、异常处理例程和日志做很大的改进。

大多数产品级系统都会在后台包含若干个数据存储。为了限制讨论范围，大多数示例都未牵扯数据库。

现代 Web 应用有很大一部分面向移动设备，而且要考虑浏览器兼容性。除非讨论到这些主题，否则我们会避免响应式设计（http://en.wikipedia.org/wiki/Responsive_web_design），而只提供一个最简化的设计。

不可见 JavaScript（unobtrusive JavaScript，参见 http://en.wikipedia.org/wiki/Unobtrusive_JavaScript）将 CSS、JavaScript 和 HTML 分开，是一种被广泛接受的最佳实践。然而在本书的示例代码中，它们常常被混在一起，这是因为对一个应用来说，仅从一个文件就可窥视其奥秘。

单元测试和测试代码示例只在和所讨论主题直接相关时才被包括进来。产品级的系统往往包含大量的测试。

让本书助你一臂之力。也许你需要在自己的程序或文档中用到本书中的代码，但除非大幅地使用，否则不必与我们联系取得授权。例如，用本书中的几段代码编写程序无需请求许可，但是销售或者发布 O'Reilly 图书中代码的光盘则必须事先获得授权；引用书中内容或代码来回答问题也无需获得授权，但将大量示例代码整合到你自己的产品文档中则必须经过许可。

使用我们的代码时，希望你能标明它的出处，但我们并不强求。出处信息一般包括书名、作者、出版商和书号，例如：*Client-Server Web Apps with JavaScript and Java*, Casimir Saternos (O'Reilly). Copyright 2014 EzGraphs, LLC., 978-1-449-36933-0。

如果还有关于使用代码的未尽事宜，请随时与我们联系：permissions@oreilly.com。

对长命令的格式化

对于长命令，我们要做适当的调整以方便大家阅读。在操作系统里有一个约定俗成的规则，即使用反斜杠插入新行。比如，下面的两个命令是等价的，在 bash 中执行结果是一样的。（bash 是一种标准的操作系统 shell，当你使用命令行访问 Linux 服务器或 Mac OS X 系统时就会用到。）

```
ls -l *someVeryLongName*
...
ls -l \
*someVeryLongName*
```

这种规则在其他用到操作系统命令的环境下也适用，比如 Dockerfiles。

同样，合法的 JSON 字符串也可以分成多行：

```
o={"name": "really long string here and includes many words"}

// 如大家期望的那样，下面的语句结果为真
JSON.stringify(o)=='{"name":"really long string here and includes many words"}'
```

```
// 同样的字符串，分成多行的结果是一样的，因此下面的结果也为真
JSON.stringify(o)=='{"name":"' +
    "some really long " +
    'JSON string is here' +
    ' and includes many, many words"}'
```

Safari® Books Online



Safari Books Online (<http://www.safaribooksonline.com>) 是应需而变的数字图书馆。它同时以图书和视频的形式出版世界顶级技术和商务作家的专业作品。

Safari Books Online 是技术专家、软件开发人员、Web 设计师、商务人士和创意人士开展调研、解决问题、学习和认证培训的第一手资料。

对于组织团体、政府机构和个人，Safari Books Online 提供各种产品组合和灵活的定价策略。用户可通过一个功能完备的数据库检索系统访问 O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 以及其他几十家出版社的上千种图书、培训视频和正式出版之前的书稿。要了解 Safari Books Online 的更多信息，我们网上见。

联系我们

请把对本书的意见和疑问发送给出版社。

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室 (100035)
奥莱利技术咨询(北京)有限公司

O'Reilly 的每一本书都有专属网页，你可以在那儿找到本书的相关信息，包括勘误表、示例代码以及其他信息。本书的网站地址是：

<http://oreil.ly/client-server-web-apps-js>

对于本书的评论和技术性问题，请发送电子邮件到：bookquestions@oreilly.com。

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息，请访问以下网站：

<http://www.oreilly.com>

我们在 Facebook 的地址如下：<http://facebook.com/oreilly>

请关注我们的 Twitter 动态：<http://twitter.com/oreillymedia>

我们的 YouTube 视频地址如下：<http://www.youtube.com/oreillymedia>

致谢

感谢人员名单如下：

- Meg、Ally、Simon 和 O'Reilly 公司的朋友们，感谢你们让我有幸写作此书。
- 感谢我的兄弟 Neal Saternos 和 James Femister 博士，他们在我学习编程早期给我很多中肯的建议。
- 感谢 Michael Bellomo、Don Deasey 和 Scott Miller 花费了宝贵的时间审阅本书。
- 感谢 Charles Leo Saternos 从使用 Lua 开发游戏中抽身出来，帮我绘制了一些图形并完成了设计工作。
- 感谢 Caleb Lewis Saternos，从他身上我学到了什么是坚持（有人坚持晨跑吗？），他完成了本书的编辑工作。
- 感谢 David Amidon 给了我从事软件开发的机会，感谢 Doug Pelletier 让我第一次开发了一个基于 Java 的 Web 应用。
- 感谢那些在项目中并肩作战的兄弟们，他们是经理 Wayne Hefner、Tony Powell、Dave Berry、Jay Colson 和 Pat Doran，当然还有首席软件架构师 Eric Fedok 和 Michael Bellomo。
- 感谢 PluralSight (<http://pluralsight.com/training>) 的 Geoffrey Grosenbach、Webucator (<http://www.webucator.com/>) 的 Nat Dunn，还有 Caroline Kvitka（和来自 Oracle 和 *Java Magazine* 的朋友），感谢他们在过去几年中给了我技术写作的机会，正是因此我才得以在今天写出本书。
- 感谢我的父母 Leo 和 Clara Saternos 在温馨的环境中将我抚养成大，感谢他们在家用 PC 还很罕见时为我购置了一台 Radio Shack Color Computer。还要感谢我的妹妹 Lori，她时刻提醒我生活中还有很多重要的事是和计算机编程无关的。

将我诚挚的爱和谢意献给我最棒的妻子 Christina 和孩子们：Clara Jean、Charles Leo、Caleb Lewis 和 Charlotte Olivia。在本书写作过程中，他们给予了我无尽的爱与支持，他们总是很有耐心，不断给我鼓励。

最后，巴赫在各个层面上都给予了我灵感。特别是他在开始工作时的专注，我愿和他一同呐喊：荣耀归于上帝（http://en.wikipedia.org/wiki/Soli_Deo_gloria）！

目录

前言	xiii
第 1 章 因变而变	1
1.1 Web 用户	2
1.2 技术	2
1.3 软件开发	3
1.4 哪些没变	4
1.4.1 Web 的本质	5
1.4.2 为什么说服务器驱动的 Web 开发有害	6
1.5 为什么需要客户端—服务器端的 Web 应用	7
1.5.1 代码组织结构 / 软件架构	7
1.5.2 “设计的灵活性”与“使用开源 API”	7
1.5.3 原型	7
1.5.4 开发者的效率	8
1.5.5 应用性能	8
1.6 小结	9
第 2 章 JavaScript 和 JavaScript 工具	11
2.1 学习 JavaScript	12
2.2 JavaScript 的历史	13
2.3 一门函数式语言	14
2.3.1 作用域	15
2.3.2 一级函数	16
2.3.3 函数声明和表达式	17
2.3.4 函数调用	19

2.3.5	函数参数	19
2.3.6	对象	20
2.4	面向 Java 开发者的 JavaScript	20
2.4.1	HelloWord.java	20
2.4.2	带变量的 HelloWord.java	23
2.5	最佳开发实践	25
2.5.1	编码规范和约定	25
2.5.2	浏览器	26
2.5.3	集成开发环境	26
2.5.4	单元测试	27
2.5.5	文档	27
2.6	项目	27
第 3 章	REST 和 JSON	33
3.1	什么是 REST	34
3.1.1	资源	34
3.1.2	动词 (HTTP 请求)	34
3.1.3	统一资源标识符	35
3.2	REST 约束	36
3.2.1	客户端-服务器端	36
3.2.2	无状态	36
3.2.3	可缓存	37
3.2.4	统一接口	37
3.2.5	分层	38
3.2.6	按需交付代码	38
3.3	HTTP 响应代码	38
3.4	JSON	39
3.5	HATEOAS	40
3.6	API 衡量和分类	43
3.7	函数式编程和 REST	43
3.8	项目	44
3.9	其他 Web API 工具	48
3.10	约束回顾	48
第 4 章	Java 工具	49
4.1	Java 语言	49
4.2	Java 虚拟机	50
4.3	Java 工具	51
4.4	构建工具	52
4.4.1	Maven 的优点	54

4.4.2	Maven 的功能	54
4.4.3	版本控制	55
4.4.4	单元测试	56
4.5	处理 JSON 的 Java 类库	56
4.6	项目	57
4.6.1	用 Java 处理 JSON	57
4.6.2	用 JVM 上的脚本语言处理 JSON	59
4.7	小结	62
第 5 章	客户端框架	65
5.1	概述	65
5.2	起点一：响应式 Web 设计	67
5.2.1	HTML5 Boilerplate	68
5.2.2	Bootstrap	68
5.3	起点二：JavaScript 库和框架	69
5.3.1	浏览器兼容性	69
5.3.2	框架	69
5.3.3	功能	70
5.3.4	流行程度	70
5.4	获取起始项目	71
5.4.1	直接从仓库下载	71
5.4.2	从入门网站下载	71
5.4.3	IDE 生成的起始项目	72
5.5	前端工程师的崛起	72
5.5.1	客户端模板	72
5.5.2	资源管道	73
5.5.3	开发流程	74
5.6	项目	74
5.7	小结	76
第 6 章	Java Web API 服务器	77
6.1	更简单的服务器端解决方案	77
6.2	基于 Java 的服务器	79
6.2.1	Java HTTP 服务器	79
6.2.2	Jetty 嵌入式服务器	81
6.2.3	Restlet	82
6.2.4	Roo	83
6.2.5	Netty 嵌入式服务器	87
6.2.6	Play 服务器	89
6.2.7	其他轻量级服务器	92