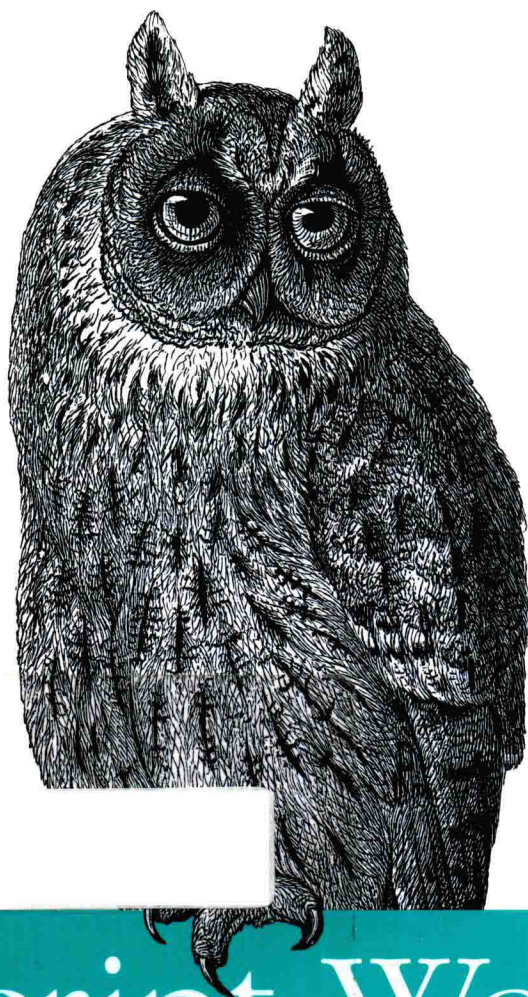


JavaScript Web Applications
面向jQuery开发者的客户端应用开发指南



基于MVC的

JavaScript Web 富应用开发

[美] Alex MacCaw 著

李晶 张散集 译

O'REILLY®

 中国工信出版集团

 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

O'REILLY®

基于MVC的 JavaScript Web富应用开发

JavaScript Web Applications

[美] Alex MacCaw 著
李晶 张敬集 译



电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

如今Web应用程序的开发已经越来越向传统软件开发靠拢了, Web和应用之间的界限也进一步模糊。传统编程语言中的设计模式、MVC、应用架构等理论也在慢慢地融入Web前端开发。这本书所涵盖的知识点非常全面, 从MVC的基本理论到网络协议、从模块解耦到异步编程模型、从HTML5/CSS3到Node.js、从软件测试到部署调试, 对于很多前端工程师来说, 这些知识正是突破自己的瓶颈所亟需的。

这本书将专注于讲述如何构建“优雅又不失高水准”(state of the art)的JavaScript应用, 包括软件架构、模板引擎、框架和库、同服务器的消息通信等内容。书中同样提供了大量的示例代码, 可以帮助你更深入地理解很多重要的概念。除此之外, 作者在MVC和架构方面的很多观点都很有启发性, 即使你不是一名JavaScript程序员, 读完本书后也会受益匪浅。

本书适合从事JavaScript开发, 寻求进阶的前端开发人员、Web架构师阅读。

978-1-449-30351-8 JavaScript Web Applications © 2011 by O'Reilly Media, Inc.

Simplified Chinese edition, jointly published by O'Reilly Media, Inc. and Publishing House of Electronics Industry, 2011. Authorized translation of the English edition, 2011 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.

本书中文简体版专有出版权由O'Reilly Media, Inc.授予电子工业出版社。未经许可, 不得以任何方式复制或抄袭本书的任何部分。专有出版权受法律保护。

版权贸易合同登记号 图字: 01-2011-7104

图书在版编目(CIP)数据

基于MVC的JavaScript Web富应用开发 / (美) 亚历克斯·麦卡劳 (Alex MacCaw) 著; 李晶, 张散集译. —北京: 电子工业出版社, 2016.11

书名原文: JavaScript Web Applications

ISBN 978-7-121-30179-7

I. ①基… II. ①亚… ②李… ③张… III. ①JAVA语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2016)第255622号

责任编辑: 许 艳

封面设计: Karen Montgomery 张 健

印 刷: 三河市双峰印刷装订有限公司

装 订: 三河市双峰印刷装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路173信箱 邮编: 100036

开 本: 787×980 1/16 印张: 19.25 字数: 462千字

版 次: 2016年11月第1版

印 次: 2016年11月第1次印刷

定 价: 59.00元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至zlts@phei.com.cn, 盗版侵权举报请发邮件至dbqq@phei.com.cn。

本书咨询联系方式: 010-51260888-819 faq@phei.com.cn。

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, 共同描绘出开创新产业的革命性思想。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

译者序

从第一眼看到封面上这只憨憨的猫头鹰开始，我就深深地喜欢上了这本 *JavaScript Web Applications*，读了简介和目录之后就已经不能自拔了。这几年鲜有深入讲架构级 Web app 的好书，这让这本 *JavaScript Web Applications* 更加难得，作为 O'Reilly 第一本专注于纯高端 JavaScript 架构思想的书，凡是有一点“架构情节”的工程师都不应当错过。

如今 Web 应用程序的开发已经越来越向传统应用软件开发靠拢了，Web 和应用之间的界限也进一步模糊。传统编程语言中的设计模式、MVC、应用架构等理论也在慢慢地融入 Web 前端开发。随着服务器端 JavaScript 和移动终端的兴起，作为一名前端工程师，也深知自己正处在一个深刻变革的年代，面对眼花缭乱的新概念和新技术更应当把握本质、认清方向，勇于创新和实践，而这本 *JavaScript Web Applications* 的出现更是一阵及时雨，为我们工作中遇到的很多难题提供了解决方案和最佳实践。同时，这本书所涵盖的知识点非常全面，从 MVC 的基本理论到网络协议、从模块解耦到异步编程模型、从 HTML5/CSS3 到 Node.js、从软件测试到部署调试，对于很多前端工程师来说，这些知识正是突破自己的瓶颈所亟需的。

这本书将专注于讲述如何构建“优雅又不失高水准”（state of the art）的 JavaScript 应用，包括软件架构、模板引擎、框架和库、同服务器的消息通信等内容。书中同样提供了大量的示例代码，可以帮助你更深入地理解很多重要的概念。除此之外，作者在 MVC 和架构方面的很多观点都很有启发性，即使你不是一名 JavaScript 程序员，读完本书后也会受益匪浅。

本书作者 Alex MacCaw 是一名 Ruby/JavaScript 程序员，是 Spine 框架的开发者。在翻译本书的过程中，我深深体会到他作为一名优秀工程师所具备的扎实的计算机专业功底和让人敬佩的开源精神。尽管这本书包含大量的专业术语，但作者文笔轻松流畅，即使直接读原文也丝毫不会感到枯燥，所以我们在翻译过程中也是非常小心，生怕丢掉这种

轻松流畅的阅读感觉，尽力为大家原汁原味地呈现本书。当然由于专业知识所限，翻译过程中难免有所疏漏，还希望各位高手批评指正。

最后，我要感谢博文视点的张春雨在译书过程中给予我们的帮助和信任。感谢我的好友王保平（玉伯）对很多关键的技术性问题提出的宝贵意见，还要感谢可爱的同事杨振楠（栋寒）、杨翰文（地极）、李燕青（霸先）、车思慧（灵玉）、陈良（舒克）的细心校对，他们给译文提了很多中肯的建议。当然，最最需要感谢的是家中的“领导”，已经记不得多少次赶译稿加班太晚，得到的不是你的抱怨，而是你的鼓励，这让我至今备感温暖。

李晶（拔赤），张散集（一舟）

前言

随着 1995 年 Netscape 浏览器的发布，JavaScript 也作为它的组成部分进入公众的视野，之后 JavaScript 的发展道路尽管充满坎坷但成长飞速，如今得益于高性能的 JIT（just in time）解析引擎，（在浏览器端）JavaScript 已经无孔不入了。在几年以前，开发者还在使用 Ajax 写一些短小的代码或热衷于实现一些类似“黄色渐褪技术”的网页特效；而现在，复杂的 JavaScript 应用已经可以写成百上千行的代码了。

就在去年，互联网出现了一股追捧 JavaScript 应用的浪潮，很多人开始着迷于给 Web 应用加入很多桌面软件的交互元素，增强 Web 应用的用户体验，这种趋势犹如星星之火迅速蔓延至整个互联网。在过去，在浏览器性能不佳的情况下，用户在进行 Web 应用时每次交互都要刷新页面，而且页面加载很慢。而如今 JavaScript 引擎已经变得异常强大，我们可以将很多交互行为植入客户端，这样交互的响应就会非常及时，增强体验。

当然获得提升的不仅仅是 JavaScript 引擎的性能。尽管 CSS3 和 HTML5 规范现在仍在修订之中，也已经有很多现代浏览器广泛支持这些新特性了，比如 Safari、Chrome 和 Firefox，IE9 也在一定程度上支持这些新特性。利用这些特性可以花更少的时间做出更棒的视觉效果，而且不用花精力做图片的切割和拼合来模拟视觉效果。现在浏览器的升级也很快，对 HTML5 和 CSS3 的支持也一天比一天好。但你还是要定义一个浏览器测试基准（你的应用所支持的最低标准的客户端软件和版本），基于此才能更加合理地选择所需的技术。

将应用的重心从服务器迁移到客户端并不轻松，这和构建服务器应用的方法完全不一样。你需要想清楚架构、模板、与服务器端的通信、框架等，这些正是本书所涵盖的内容。我将手把手教你如何构建“优雅又不失高水准”的 JavaScript 应用。

本书的目标读者

本书不是为 JavaScript 初学者所写，如果你对 JavaScript 这门语言缺乏基本的了解和认识，我建议你先阅读一些更基础的书，比如 Douglas Crockford 著的《JavaScript 语言精粹》(<http://oreilly.com/catalog/9780596517748>) (O'Reilly)。本书更适合有一些 JavaScript 开发经验的开发者，比如使用 jQuery 类库的开发者，或者当你希望构建更复杂、更高级的 JavaScript 应用时，本书也是适合你的。此外，本书的很多章节，特别是附录，对于有经验的 JavaScript 开发者来说也是非常有帮助的。

本书的内容组织

第 1 章

本章从 JavaScript 的发展历程开始，介绍了 JavaScript 的发展现状和对互联网的巨大影响。然后轻描淡写地介绍了 MVC 的基本概念，随后又讲解了 JavaScript 的构造函数、原型继承及如何使用 JavaScript 创建一个类库。

第 2 章

本章主要介绍了浏览器的事件机制，包括事件机制的发展历史，API 设计和事件模型的行为和实现。然后讲解了如何基于 jQuery 绑定事件监听、使用代理，以及创建自定义事件。最后使用发布 / 订阅模式实现了“DOM 无关”事件。

第 3 章

本章讲解了如何在你的应用中使用 MVC 模型，包括加载和操作远程数据。我们将会提到为什么在构建 ORM 类库的时候使用 MVC 和命名空间是如此之重要，以及如何使用 ORM 类库来管理模型数据。接下来讲解了如何使用 JSONP 和跨域 Ajax 来加载远程数据。最后介绍了如何通过使用 HTML5 本地存储和将本地存储提交至 RESTful 服务器，来实现模型数据的持久化。

第 4 章

本章演示了如何使用控制器模式在客户端保持一个状态。我们将讨论如何将逻辑封装成模块、阻止全局命名空间的污染，然后介绍如何使用视图来进一步简化控制器的结构，以及怎样在视图中实现 DOM 事件监听。本章的最后将会讨论路由选择，包括使用 URL 中的 hash 片段，使用新的 HTML5 History API 等技术，以及确保解释两种方法的利弊。

第 5 章

本章介绍了视图和 JavaScript 模板，给出了多种动态渲染视图的方式，以及很多模

板类库和存储模板的方式（使用行内形式存储模板、使用 script 标签，以及远程加载）。接下来，你会接触到数据绑定的一些内容，包括使模型控制器、视图与模型数据、视图数据动态同步连接。

第 6 章

本章详细介绍了使用 CommonJS 模块系统来做 JavaScript 的依赖管理。开始会介绍 CommonJS 背后的历史和思想，接下来会讲解如何在浏览器端使用 CommonJS 模块，包括介绍一些模块加载器类库，比如 Yabble 和 RequireJS。然后，我们讨论了如何自动在服务器端包装模块，从而提高性能、节省时间。本章的最后会介绍 CommonJS 的一些替代方案，比如 Sprockets 和 LABjs。

第 7 章

这里将会讲到 HTML5 带给我们的一些好处：文件操作 API。本章将会涵盖文件操作 API 的浏览器支持情况、多文件上传、拖曳上传文件及使用剪贴板事件。接下来会介绍使用二进制大文件和文件切割来读文件，同时将读取的结果在浏览器中输出。然后讲解使用 XHR（XMLHttpRequest）Level 2 规范来实现在后台上传文件，最后向大家展示一个使用 jQuery Ajax API 实现文件上传进度指示的例子。

第 8 章

本章主要关注实时应用和 WebSocket 技术的一些令人兴奋的发展趋势。首先介绍实时应用的发展历史及各种实现技术的浏览器兼容性情况。然后更详细地介绍 WebSocket 和基于它的更高级的实现，包括浏览器兼容性和 JavaScript API。接下来展示一个使用 WebSocket 实现的简单的 RPC 服务，看一下如何在客户端和服务端之间建立连接。然后介绍 Socket.IO 和如何搭建实时架构，最后介绍用户体验方面的一些考量。

第 9 章

本章主要讲解测试和调试的内容，这些内容是 JavaScript 网络应用开发过程中的关键环节。我们的话题将围绕跨浏览器测试的主题进行展开，介绍浏览器基准的选择、单元测试和测试类库，比如 QUnit 和 Jasmine。接下来，介绍自动化测试和持续集成服务器，比如 Selenium。然后讲解调试相关的内容，研究了 Firefox 和 WebKit 网络监测器、主控台，以及使用 JavaScript 调试器。

第 10 章

本章介绍了另外一个非常重要却又极易被忽略的内容——JavaScript 网络应用的部署。我们主要考虑性能方面，以及如何使用缓存、代码压缩、Gzip 压缩及其他减少应用初始化加载时间的技术。最后简单讲解了如何使用 CDN 服务器来让我们的工作事半功倍，以及如何使用浏览器内置的策略来提升你站点的性能。

第 11 章

接下来的 3 章主要介绍了一些流行的 JavaScript 类库，这些类库常用来做 JavaScript 应用开发。Spine 是一个轻量级的 MVC-compliant 类库，这个类库使用了本书中讲到的很多概念。本章将会为你介绍类库的核心部分：类、事件、模型和控制器。最后本章用一个管理应用的例子来展示本章所讲到的知识点。

第 12 章

Backbone 是一个非常流行的类库，使用这个类库可以非常高效地构建 JavaScript 应用，本章主要介绍这个类库。本章会涵盖 Backbone 的核心观念和类，比如模型、集合、控制器和视图等。接下来会介绍使用 RESTful JSON 请求从服务器同步获取模型数据，以及如何在服务器端响应 Backbone。最后我们给出一个待办事项列表应用的例子，来向大家展示如何使用这个类库。

第 13 章

本章主要介绍了 JavaScriptMVC 类库，这是一个流行的基于 jQuery 的框架，用来构建 JavaScript 网络应用。在本章中你将会学到 JavaScriptMVC 的一些基础知识，比如类、模型和控制器，同时还包含客户端的模板及渲染视图。本章的最后会给出一个实际的 CRUD 列表的例子，给读者展示使用 JavaScriptMVC 创建抽象的、可重用的、节省内存的组件是多么简单。

附录 A

附录 A 中是对 jQuery 的简要介绍，如果你想温习类库内容，则这部分内容对你会非常有帮助。本书中大部分示例代码都是基于 jQuery 的，首先熟悉 jQuery 是很重要的。这一部分会讲到大部分核心的 API，比如 DOM 操作、DOM 查询和遍历，以及事件绑定、触发和事件代理。接下来会讲解 jQuery 的 Ajax API，包括 POST、GET 和 JSON 请求。随后将介绍 jQuery 扩展，如何使用 jQuery 来封装一个插件，让你的代码更具通用性。最后展示了一个实际的例子：创建一个 Growl jQuery 插件。

附录 B

附录 B 主要讲解 Less，Less 是 CSS 的超集，它使用变量、混合、操作符和优雅的规则扩展了 CSS 本身的语法。利用这些规则可以极大地减少你所写的 CSS 代码量，特别是使用 CSS3 效果更佳。附录 B 包含 Less 的主要的增强的语法，以及如何使用命令行工具和 JavaScript 类库来将 Less 文件编译成 CSS。

附录 C

附录 C 主要讲解了 CSS3。首先介绍了一些 CSS3 的背景知识、浏览器厂商的前缀，然后介绍了 CSS3 的主要内容，从主要附件到规格说明。这里介绍的 CSS 特性

主要包括：圆角、`rgba` 颜色、阴影、渐变、动画和变换。附录的最后讨论了使用 Modernizr 实现的优雅降级，并展示了一个实际的使用 `box-sizing` 规范的例子。

本书的约定

本书使用下列排版约定：

斜体 *Italic*

用于表示新术语、URL、电子邮件地址、文件名、文件扩展名和事件。

等宽字体 `Constant width`

用来表示计算机代码片段，包括命令、数组、元素、语句、操作符、变量、属性、关键字、函数、类型、类、命名空间、方法、形参、实参、值、对象、事件处理程序、XML 标签、HTML 标签、宏指令、文件内容及命令行的输出等。

等宽加粗字体 **Constant width bold**

用来表示命令或者其他用户输入的文本。

等宽斜体 *Constant width italic*

用来表示可被替换的字符或文本，这些字符在合适的场景和特定的条件下会被替换成其他的值。



这个图标表示一种提示、建议或一般的消息提醒。



这个图标表示一种警告。

中文版书中切口处的“”表示原书页码，便于读者与原英文版图书对照阅读，本书的索引所列页码为原英文版页码。

附加文件

本书的附加文件都存放在 GitHub 上 (<https://github.com/maccman/book-assets>)，可以直接在 GitHub 上查看，也可以下载压缩包 (<https://github.com/maccman/book-assets/zipball/master>)。所有这些示例代码都以章节为单位存放，都已经包含了各自所需的类库，本书中用到的大多数示例代码同样在单独的文件中。

在本书中凡是引用这些附加文件的地方，都会以这种形式表述：`assets/chapter_number/name`。

代码约定

本书中我们以 `assert()` 和 `assertEqual()` 函数来展示变量的值或者函数调用的结果。`assert()` 是一种快捷表述方式，用来表示一个特定的变量是真值。这在自动化测试中是一种非常常见的模式。`assert()` 可以接收两个参数：一个值和一个可选的消息。如果运行结果不是真值，这个函数将抛出一个异常：

```
var assert = function(value, msg) {
    if ( !value )
        throw(msg || (value + " does not equal true"));
};
```

`assertEqual()` 是表示一个值等于另外一个值的另一种表述。它和 `assert()` 类似，但接收两个值。如果这两个值不相等，则这个断言失败：

```
var assertEqual = function(val1, val2, msg) {
    if (val1 !== val2)
        throw(msg || (val1 + " does not equal " + val2));
};
```

这两个函数非常简单，正如你在示例代码中所看到的。如果断言失败，你会在浏览器的控制台中看到一个错误消息：

```
assert( true );

// 和 assertEqual() 等价
assert( false === false );

assertEqual( 1, 1 );
```

我们可以从代码中看出，对象比较会失败，除非两个对象是指向同一块内存的引用。解决办法是深比较，在 *assets/ch00/deep_equality.html* 这个例子中可以看到完整的代码。

jQuery 示例代码

本书的大部分示例代码都是基于 jQuery (<http://jquery.com>) 的，jQuery 是现在最流行的 JavaScript 类库，它对事件、DOM 遍历、DOM 操作和 Ajax 都做了封装。这里我选用 jQuery 是出于几个原因的考虑，最主要的原因是 jQuery 可以让代码变得非常简洁，而且当下大部分人对 jQuery 都非常熟悉，一看即懂。

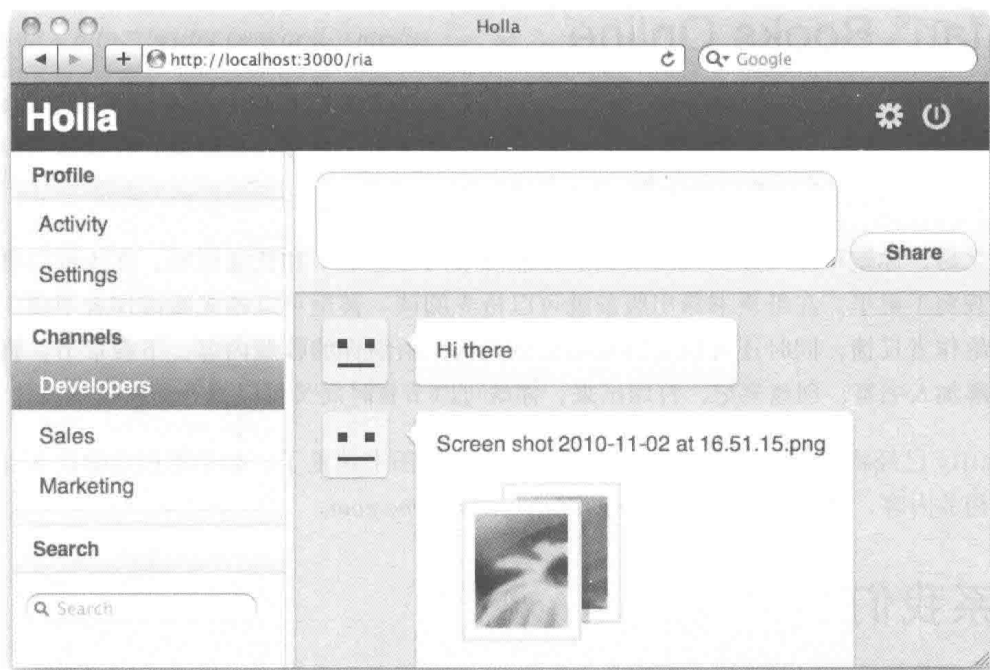
如果你没有使用过 jQuery，我强烈推荐你首先看一下 jQuery 的文档。它的 API 非常不错，为 DOM 提供了一组非常棒的抽象的接口。可以在附录 A 中查阅到简短的 jQuery 入门内容。

Holla

Holla (<http://github.com/maccman/holla>) 贯穿本书始终，它是一个 JS 群聊应用。Holla 是一个非常不错的示例应用，因为它和本书中大多数章节和内容都有交集。除了正文章节中对 Holla 的讲述之外，Holla 为我们展示了：

- 使用 CSS3 和 HTML5 来构建美观的界面。
- 拖曳上传文件。
- 使用 Sprockets 和 Less 来编写代码。
- 使用 WebSocket 将数据发送给客户端。
- 创建带有状态的 JavaScript 应用。

可以从 Holla 的 GitHub 的代码库 (<http://github.com/maccman/holla>) 中将代码复制下来，研读一下它的代码。本书中用到的很多例子都来自 Holla 的源代码，Holla 的界面如图 P-1 所示。



图P-1：Holla聊天应用程序运行界面

作者附言

本书是我在环游世界的时候完成的，这花费了我一年的时间。这一年我游历了很多地方，这本书一部分是我在非洲时编写的，那里没有电也没有网络，还有一部分是在日本古朴幽静的寺院中和凝霞烂漫的樱花树下完成的，还有一些内容是在遥远美丽的哥伦比亚岛屿上完成的。我非常享受这段时光，希望我的这种美妙的体验能通过我的文字传达给每一位读者。

这里我要特别感谢一些人。感谢 Stuart Eccles、Tim Malbon、Ben Griffins 和 Sean O’Halpin，是他们给了我这个机会，让我重新找寻到埋藏在心底的激情。同样要感谢 James Adam、Paul Battley 和 Jonah Fox，他们是我值得尊敬的导师，谆谆之言让我获益良多。

同样要感谢出版社的编辑们，他们严格的审校保证了本书的质量：Henrik Joreteg、Justin Meyer、Lea Verou、Addy Osmani、Alex Barbara、Max Williams 和 Julio Cesar Ody。

当然最需要感谢的是我的父母，他们的默默支持是我坚实的后盾。

Safari® Books Online



Safari 在线图书是一个数字图书馆，读者可以在这个图书馆里自选图书，在这里可以搜索到超过 7500 本技术相关的书籍创作和视频，在这里可以迅速找到你想要的内容。

订阅之后，你就可以阅读在线图书馆的任意图书的任意章节和任意视频。你还可以将图书下载到手机里。在纸质书籍出版前就可以抢先阅读，甚至可以抢先阅读作者手稿，并实时给作者反馈。同时还可以复制粘贴实例代码、组织你的收藏内容、下载章节、将关键段落加入书签、创建笔记、打印出来，你既可以节省时间又可以提升阅读效率。

O’Reilly 已经将本书（英文版）上传至 Safari 在线图书馆里了。如果想在线阅读本书和其他相关内容，请免费登录 <http://my.safaribooksonline.com>。

联系我们

对于本书的评论或问题请联系出版商：

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

我们为本书制作了一个 Web 页面，页面中包含了简介、样章，以及其他信息。可以从这里访问这个页面：

<http://www.oreilly.com/catalog/9781449303518>

<http://www.oreilly.com.cn>

如果要留言或者提交关于本书的技术问题的反馈，请发邮件至：

bookquestions@oreilly.com

本书的更多信息、资源、参考文献和新闻，请登录出版社官网：*<http://www.oreilly.com>* 或者 *<http://www.oreilly.com.cn/>*。

Facebook：*<http://facebook.com/oreilly>*

Twitter：*<http://twitter.com/oreillymedia>*

YouTube：*<http://www.youtube.com/oreillymedia>*

目录

第1章 MVC和类	1
最初	1
增加结构	2
什么是 MVC	2
模型	3
视图	4
控制器	5
向模块化进军，创建类	6
给类添加函数	7
给“类”库添加方法	8
基于原型的类继承	10
给“类”库添加继承	11
函数调用	12
控制“类”库的作用域	15
添加私有函数	17
“类”库	18
第2章 事件和监听	21
监听事件	21
事件顺序	22

取消事件	23
事件对象	23
事件库	25
切换上下文	26
委托事件	26
自定义事件	27
自定义事件和 jQuery 插件	28
DOM 无关事件	30

第3章 模型和数据 33

MVC 和命名空间	33
构建对象关系映射 (ORM)	34
原型继承	35
添加 ORM 属性	36
持久化记录	37
增加 ID 支持	39
寻址引用	40
装载数据	41
直接嵌套数据	42
通过 Ajax 载入数据	42
JSONP	46
跨域请求的安全性	46
向 ORM 中添加记录	47
本地存储数据	47
给 ORM 添加本地存储	49
将新记录提交给服务器	51

第4章 控制器和状态 53

模块模式	54
全局导入	54
全局导出	54
添加少量上下文	55
抽象出库	56