

构建 实时Web应用

基于HTML5 WebSocket、 PHP和jQuery

Jason Lengstorf Phil Leggetter 著 肖智清 译

Realtime Web Apps
with HTML5 WebSocket, PHP, and jQuery

- 实时Web应用开发领域的经典著作，由实时Web技术领域的布道者和资深Web开发工程师撰写
- 系统讲解构建实时Web应用的过程、方法与核心技术，同时包含大量代码和设计示例，实战性极强

构建 实时Web应用

基于HTML5 WebSocket、
PHP和jQuery

Realtime Web Apps
with HTML5 WebSocket, PHP, and jQuery

Jason Lengstorf Phil Leggetter 著 肖智清 译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

构建实时Web应用: 基于HTML5 WebSocket、PHP和jQuery / (美) 灵格斯多夫 (Lengstorf, J.), (英) 勒格特 (Leggetter, P.) 著; 肖智清译. —北京: 机械工业出版社, 2013.9

(Web开发技术丛书)

书名原文: Realtime Web Apps: With HTML5 WebSocket, PHP, and jQuery

ISBN 978-7-111-43983-7

I. 构… II. ①灵… ②勒… ③肖… III. 网页制作工具 IV. TP393.092

中国版本图书馆CIP数据核字 (2013) 第212700号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2013-4814

Realtime Web Apps: With HTML5 Websocket, PHP, and jQuery by Jason Lengstorf, Phil Leggetter (ISBN: 978-1-4302-4620-6)

Original English language edition published by Apress L.P., 2560 Ninth Street, Suite 219, Berkeley, CA 94710 USA. Copyright © 2013 by Apress L.P. Simplified Chinese-language edition copyright © 2013 by China Machine Press. All rights reserved.

This edition is licensed for distribution and sale in the People's Republic of China only, excluding Hong Kong, Taiwan and Macao and may not be distributed and sold elsewhere.

本书原版由 Apress 出版社出版。

本书简体字中文版由 Apress 出版社授权机械工业出版社独家出版。未经出版者预先书面许可, 不得以任何方式复制或抄袭本书的任何部分。

此版本仅限在中华人民共和国境内 (不包括中国香港、台湾、澳门地区) 销售发行, 未经授权的本书出口将被视为违反版权法的行为。

实时 Web 应用开发领域的经典著作, 由实时 Web 技术领域的布道者和资深 Web 开发工程师撰写。不仅详细讲解了构建实时 Web 应用所需的各项技术, 还系统讲解了实时 Web 应用规划与设计的过程和方法, 为构建实时 Web 应用提供了翔实的指导。此外, 本书包含大量代码和设计示例, 实战性极强。

全书一共 10 章: 第 1 章介绍了什么是实时 Web 技术及其原理; 第 2 章详细讲解了构建实时 Web 应用需要哪些技术和工具; 第 3 章讲解了如何利用 Pusher 构建实时 Web 应用; 第 4 章介绍了如何根据需要在 Web 应用和原生应用之间做出选择; 第 5 章讲解了如何规划应用程序的功能和结构; 第 6 章讲解了实时 Web 应用的设计; 第 7 章讲解了如何为实时 Web 应用创建 HTML 和 CSS 标记; 第 8 章和第 9 章则非常详细地讲解了如何构建实时 Web 应用的后端程序; 第 10 章讲解了如何实现实时事件和 jQuery 效果。

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: 谢晓芳

三河市杨庄长鸣印刷装订厂印刷

2013年10月第1版第1次印刷

186mm×240mm·18印张

标准书号: ISBN 978-7-111-43983-7

定 价: 69.00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) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzsj@hzbook.com

译者序

当前，实时应用如雨后春笋般涌现出来。在各种智能手机的通知栏，以及Windows等操作系统的右下角弹窗区和通知区，常常能看到各类应用程序推送的消息。除了社交软件之外，甚至即时通信软件、安全软件等，都开始使用实时技术发布消息。实时技术能在新事件发生时，让用户即刻获得新的消息。本书对实时技术进行了详尽的介绍。

本书将理论与实际较好地结合了起来。第一部分着重介绍实时技术的来龙去脉，并做了一些准备工作。这让读者对整个开发流程和相关技术的理论基础有了更好的了解，使读者不仅“知其然”，而且“知其所以然”。第二部分专门介绍设计过程。俗话说，“磨刀不误砍柴工”，好的设计可以让开发事半功倍。第三部分则讲述开发过程。另外，附录介绍了一种用户认证的方法OAuth。这类用户认证方法不需要用户在新的网站上注册。在国内，可能出于对用户信息收集的考虑，这种用户认证方法使用范围比较小。事实上，译者认为，如果用法得当，这种认证方法能够更快且更好地收集用户信息。译者推荐使用这样的认证方式。

如果你认真阅读全书，将对以下技术有所了解：HTML（包括HTML5）、CSS（包括CSS3）、JavaScript（以及jQuery）、PHP、MySQL（与SQL语言）、Pusher（及其API）、OAuth（及Facebook的对应API），另外，还涉及Photoshop的一些使用技巧。

本书源代码可以在<http://www.apress.com/9781430246206>中的Source Code/Download部分下载。源代码都以完整的项目给出。在配置好开发环境后，这些项目可以直接运行。

本书是一本开发技术进阶书籍。适合阅读本书的读者有：

- 对编程有一定的了解，对Web技术有所耳闻，想要更深入地学习Web开发技术的读者。
- 有若干年的开发经验，正在或将要从事Web 2.0、社交网络开发的开发人员。
- 想要学习最新Web开发技术（如HTML5、CSS3）的大学师生与研究人员。
- 需要对网络开发流程和涉及的技术有全面了解的技术管理人员或创业者。

本书中文版由清华大学肖智清翻译。在翻译过程中，译者将以帮助读者理解并掌握开发技术为根本宗旨，力求保证术语的准确和语言的通顺，力争做到“信、达、雅”。限于译者水平，加上时间仓促，书中难免存在错误和不足，恳请广大读者和同行批评指正。译者的电子邮箱是：xzq.xiaozhiqing@gmail.com。

这里要感谢为本书中文版出版做出贡献的所有工作人员。其中，机械工业出版社的谢晓芳是本书的责任编辑，她对本书进行了全面细致的审校，并提出了许多建设性意见。同时，还要感谢机械工业出版社的其他编辑为提升本书质量做出的审稿工作，与他们合作是一个愉快的过程。最后，还要感谢我的亲友，特别是我的爸爸和妈妈，他们在本书翻译期间给予我极大的支持。

感谢您选择本书。祝您学习快乐！

肖智清
于清华大学

前言

几年前，我参加了一个名为“Keeping It Realtime”的会议。在那个会议中，许多演讲者都是实时行业中的资深人士，他们正在解决人们闻所未闻的问题。

在那时，技术的力量正在推进，实时技术的应用领域非常令人惊讶。当时，我想了解更多的信息，所以我就立即开始了解实时技术。那么，我是怎样开始在自己的应用程序中使用这个神奇的新东西呢？

我走进了距离我最近的一个分会场，与其中的听众坐在一起，我马上就被其中的内容迷住了。在讲台上，一个留着小胡子的人使用他的笔记本电脑，一边对着麦克风说话，一边用惊人的速度在Vim中编写代码。在那时，我能看出他在初始化socket.io，他已经完成了应用程序的一半。

我马上就沉迷其中了。同时，我开始思考，这个令人惊叹的技术是不是只能供极少数的精英开发人使用？如果我没有那个教授实时技术的人那么厉害，我是否还应该构建自己的应用程序？

如果你曾让一个聪明的开发人员做一些事，你也许会有这样的感觉：当某人的聪明程度到了一定境界后，他们有时就不会告诉我们他们曾经用过的技术。这意味着我们可能需要研究海量复杂的代码、标准和文档，或只能放弃。

本书旨在帮助我们揭开实时编程的神秘面纱，使得所有具有中等PHP和JavaScript水平的开发人员都能够进行实时编程。如果你想即刻将实时技术用在实际的项目中，那么你并不需要知道构建Flash polyfill或维护Node.js的方法，你只需要学习本书即可。

我们相信，虽然理论是有趣和必要的，但是实际开发更激动人心。实际开发将理论运用于实际，实现了理论的价值。在最后，将以非常简单的方式设置本书使用的技术，而不需要你学习新的编程语言或框架。本书基于的Web技术与那些最流行的应用、网站、内容管理系统使用的技术相同。

实时技术应该属于广大的群众。现在，为自己倒一杯咖啡，或沏一杯茶，开始学习实时技术，你能够在实时技术没落前编写出具有实时功能的程序。

致谢

本书的诞生要感谢Phil Leggetter，他使进度不断推进。他添加了大量与技术有关的背景信息，并就添加详细的解释和例子做了很多工作。没有他的帮助，本书可能就半途夭折了。

Ben和Ana：非常感谢你们的耐心。

Alison：感谢你在我每天晚上通宵写书的情形下没有弃我而去。

Nate：感谢你提供友好的竞争环境，在我完成每一个目标的情况下提醒我还需进行下一步的工作。

我的父亲：感谢您一直支持我。我的妈妈：感谢这些年来您与爸爸一起将我哺育成人。

Drew：感谢你来到Copter Labs，使得我有足够的时间来完成这个工作。Alex、Anne、Jason、Kevin、Rob、Roger和Wes：感谢你们加入Copter团队。

由于Richelle、Troy、Taunja、Chris和其他朋友的帮助，使我避免成为从不打扫卫生的“隐士”，感谢你们继续邀请我做一些事情。

——Jason Lengstorf

目 录

译者序

前言

第一部分 熟悉必备技术

第1章 什么是实时 / 2

1.1 传媒的演化 / 2

1.1.1 是网站而不是Web应用 / 3

1.1.2 HTTP解决方案 / 4

1.1.3 一个先要解决的问题：实时
究竟意味着什么 / 5

1.1.4 AJAX / 5

1.1.5 轮询 / 5

1.1.6 HTTP长轮询 / 7

1.1.7 HTTP流 / 8

1.1.8 在Web浏览器中使用基于HTTP
的解决方案的其他问题 / 9

1.2 解决方案：WebSocket / 11

1.3 为什么要学习实时Web技术 / 13

1.4 请即刻在你的应用中使用实时
Web技术 / 14

1.5 小结 / 14

第2章 工具 / 15

2.1 我们要构建什么 / 15

2.2 选择工具 / 16

2.2.1 HTML5 / 16

2.2.2 CSS3 / 19

2.2.3 JavaScript和jQuery / 22

2.2.4 PHP / 26

2.2.5 MySQL / 28

2.2.6 HTML5的WebSocket技术和
Pusher / 31

2.2.7 OAuth / 33

2.3 小结 / 35

第3章 Pusher / 36

- 3.1 Pusher简史 / 36
- 3.2 为什么要使用Pusher / 36
 - 3.2.1 扩展性 / 37
 - 3.2.2 WebSocket、旧技术支持和自动重连接 / 37
 - 3.2.3 其他客户端库 / 37
 - 3.2.4 REST API / 37
 - 3.2.5 服务器库 / 37
 - 3.2.6 开发人员工具 / 38
 - 3.2.7 文档 / 38
- 3.3 Pusher中的术语 / 38
- 3.4 开始使用Pusher / 39
- 3.5 使用Pusher发送事件 / 45
- 3.6 调试Pusher应用程序 / 54
- 3.7 小结 / 55

第二部分 规划应用

第4章 选择Web应用 / 58

- 4.1 为什么要在Web应用与原生应用间做抉择 / 58
- 4.2 要考虑的因素 / 58
 - 4.2.1 了解用户 / 59
 - 4.2.2 市场推广 / 59
 - 4.2.3 销售 / 61
 - 4.2.4 发布应用程序 / 61

- 4.2.5 外观和性能 / 62

- 4.2.6 开发 / 63

- 4.3 根据需要进行选择 / 66

- 4.3.1 选择Web应用而不是原生应用 / 67

- 4.3.2 最终的决定：构建Web应用程序 / 67

- 4.4 小结 / 68

第5章 确定应用的功能和结构 / 69

- 5.1 应用要做什么 / 69

- 5.2 应用不做什么 / 69

- 5.3 用户扮演的角色 / 70

- 5.3.1 主持人 / 70

- 5.3.2 参与者 / 70

- 5.4 前端规划 / 71

- 5.4.1 要使用的技术 / 71

- 5.4.2 使用HTML5 / 71

- 5.4.3 CSS3、媒体查询以及它们如何影响设计和HTML / 74

- 5.4.4 效果和动画 / 78

- 5.5 后端规划 / 79

- 5.6 将所有这些整合入线框图中 / 85

- 5.6.1 筹划主页 / 85

- 5.6.2 筹划参与者的问答页面 / 85

- 5.6.3 筹划主持人的问答页面 / 86

- 5.7 小结 / 87

第三部分 构建基本内容

第6章 设计应用 / 90

- 6.1 为设计设置目标 / 90
- 6.2 定义颜色面板 / 91
- 6.3 选择字体 / 91
- 6.4 设计常见的页面元素 / 93
 - 6.4.1 创建页眉 / 93
 - 6.4.2 创建页脚 / 95
 - 6.4.3 表单元素 / 96
- 6.5 设计主页视图 / 100
 - 6.5.1 创建房间的表单 / 100
 - 6.5.2 加入房间的表单 / 102
- 6.6 设计房间视图 / 104
 - 6.6.1 设计参与者视图 / 104
 - 6.6.2 设计关闭的房间视图 / 105
 - 6.6.3 设计主持人视图 / 105
- 6.7 小屏幕布局 / 107
- 6.8 小结 / 108

第7章 创建HTML和CSS标记 / 109

- 7.1 开始构建基本部分：设置HTML5文档 / 109
- 7.2 获得需要的字体 / 110
- 7.3 常见的元素 / 113
 - 7.3.1 页眉标记 / 113
 - 7.3.2 页脚标记 / 114

7.3.3 样式 / 115

7.3.4 使得页眉和页脚具有响应 / 119

7.4 开发主页视图 / 120

7.4.1 编写标记 / 120

7.4.2 添加媒体查询 / 126

7.5 开发参与者的活动房间视图 / 127

7.5.1 编写标记 / 128

7.5.2 实现CSS / 131

7.5.3 添加媒体查询 / 136

7.6 开发参与者的关闭的房间的视图 / 139

7.6.1 尽可能少引入新标记 / 139

7.6.2 添加样式 / 140

7.6.3 关于媒体查询 / 140

7.7 开发主持人的房间视图 / 141

7.7.1 修改现有的标记 / 141

7.7.2 更新CSS / 143

7.7.3 更新媒体查询 / 144

7.8 小结 / 146

第8章 构建后端：第1部分 / 147

8.1 计划简单的MVC框架 / 147

8.1.1 确定文件夹结构 / 147

8.1.2 为所有的请求设置路由 / 148

8.1.3 设置实用工具函数 / 152

8.1.4 结束路由的编写 / 161

8.1.5 设置核心类 / 163

8.1.6 创建抽象模型类 / 170

8.2 增加页眉标记和页脚标记 / 171

8.3 构建主页 / 174

8.3.1 创建主页控制器 / 174

8.3.2 创建主页视图 / 175

8.4 添加错误处理程序 / 178

8.4.1 创建错误控制器 / 178

8.4.2 创建错误视图 / 179

8.4.3 添加与错误有关的样式 / 180

8.4.4 测试错误页面 / 180

8.5 构建数据库 / 181

8.6 处理表单提交 / 182

8.6.1 计划表单提交工作流程 / 182

8.6.2 设置并检查有效的动作 / 183

8.6.3 防止重复提交和欺骗性的

提交 / 184

8.6.4 编写表单处理方法 / 185

8.7 小结 / 189

第9章 构建后端：第2部分 / 190

9.1 构建问题 / 190

9.1.1 构建Question控制器 / 190

9.1.2 添加问题视图 / 192

9.1.3 完成视图编写 / 193

9.1.4 添加提出问题表单 / 200

9.1.5 构建问题模型 / 201

9.1.6 为控制器添加表单处理程序和

数据访问方法 / 205

9.2 构建房间 / 209

9.2.1 增加Room控制器 / 209

9.2.2 构建房间模型 / 216

9.2.3 向Room控制器添加表单处理
程序 / 221

9.3 测试所有代码 / 225

9.3.1 创建第一个房间 / 225

9.3.2 关闭房间 / 226

9.3.3 重新打开房间 / 227

9.3.4 加入房间 / 227

9.3.5 提出第一个问题 / 228

9.3.6 为问题投票 / 228

9.3.7 回答问题 / 229

9.4 小结 / 231

**第10章 实现实时事件和jQuery
效果 / 232**

10.1 添加需要的证书和库 / 232

10.1.1 获得Pusher的API证书 / 232

10.1.2 下载Pusher的PHP API

包装 / 234

10.1.3 载入Pusher的JavaScript API

包装 / 234

10.1.4 载入jQuery / 235

10.2 在后端实现实时 / 235

10.2.1 创建事件 / 235

10.2.2 测试实时事件 / 236

10.3 在前端实现实时 / 238

10.3.1 订阅通道 / 238

- 10.3.2 绑定事件 / 239
 - 10.4 增加效果 / 240
 - 10.4.1 处理房间事件 / 241
 - 10.4.2 为增加新问题添加动画 / 242
 - 10.4.3 为问题增加投票 / 242
 - 10.4.4 回答问题中的动画和问题重排 / 246
 - 10.5 小结 / 248
- 附录A 深入理解OAuth / 249**

第一部分

熟悉必备技术

构建Web应用程序并不是一件单纯的事情。现代Web开发人员需要综合利用多种技术来构建满足用户需求的应用程序。

在本书的这一部分，你将熟悉构建第一个实时Web应用程序所需要的技术。由于该项目会利用在本书编写时更常使用的Web技术，因此你也许会对书中该部分的许多内容感到熟悉。如果你觉得不用复习这些内容，那么可以跳过它们。



第1章 什么是实时

如果你关注最近一两年Web开发的趋势，那么你肯定会发现“实时”（realtime）这个术语无处不在。那么，什么是“实时”？它与当前的Web技术有何不同？为什么我们要使用实时技术？

为了更好地理解“实时”的含义，以及实时技术如何改变我们已经了解的Internet，让我们来看看实时技术试图解决的问题的历史。实时技术要解决的问题是：我们如何在不需要用户部分做任何动作的情况下影响客户端上Web应用程序的状态？

1.1 传媒的演化

实事求是地说，当谈到信息时，我们会期望首先得到最新的消息。这种期望可以归因于一种与生俱来的知识渴求，期望第一时间获悉某个消息，或者仅仅因为我们想成为消息灵通人士。如图1-1所示，在某些情况下，成为第一个获得消息的人甚至比消息本身是什么更重要。（这正是消息灵通人士存在的全部理由。）我们想要首先知道消息，这意味着我们希望信息立即出现。

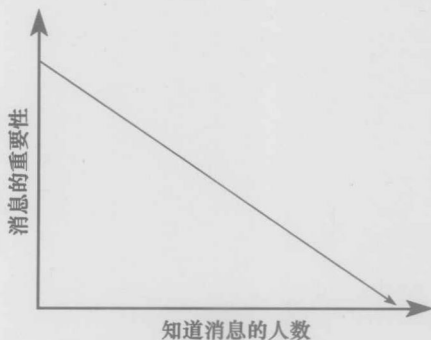


图1-1 某一特定类型的信息感知价值随着该信息的广泛流传而减小

对最新消息无休止地追求使我们成为现在这个样子：我们不再满足于洞穴里的壁画、厚重的手写巨著、书或宣传单这样的出版物，我们还期待更多的东西；虽然报纸和期刊可以快到每天上午更新一次，但是那些更新的内容都是昨天发生的；广播和电视可以给我们

传递信息，但是那需要数小时，在最好的情况下也要数分钟。

Internet让我们能够与全球的用户共享信息。但是，在其中发现信息仍然需要花费较长的时间。同时，我们会依赖像电子邮件或论坛这样的东西来传递消息。Google通过让数据更容易被发现，改变了这一切。即使这样，页面索引的速度意味着我们仍然需要在搜索时等待待发现的数据。实时更新博客的发明意味着如果我们知道从何处寻找更新，那么我们就更能频繁地接受更新。而更新发生的地方常常是常见的传媒品牌。

一些社会传媒发动大众，创建了一个全球的网络。在这个网络中，任何人都可以分享消息。对一些像2011埃及革命^①这样的事件，Twitter这样的服务是我们信息的主要来源。但是，第一个实时Web游戏规则的改变者是这样的一种应用，它第一次实现了新消息在发布后就即刻可以通过搜索发现。这开始说明了通过Internet访问新信息的价值，并增加用户对实时内容的期望，甚至导致了著名科技评论家Robert Scoble发出“实时Web是否对Google造成威胁”的疑问^②。

社交媒体平台逐渐转变为实时通信平台。在一个人发出状态更新后不久，他就会得到一个或多个用户的回复。我们中的大多数人都觉得这种快速的、交互式的反馈十分新颖。在这之前，我们可能只是玩基于Flash的游戏，习惯于Internet应用程序提供相对静态的单一用户体验。这种新的多用户交互功能会使用户的粘性更强，有效提升用户体验。

现在，传媒已经从原先有延时的、静态的内容演进为更丰富、更生动、更具交互性的内容。用户已经看到了这些改变，他们对Internet应用程序的期望大大增加。

Internet和社会媒体演示了这些即时的满足感。即使这样，许多来源仍然不能给我们以实时内容的形式提供新闻，或不能给我们提供交互性的、有趣的体验。为什么这些媒体不能提供这样的体验呢？

1.1.1 是网站而不是Web应用

在传统情况下，Internet用于分享静态内容。网站只是一些静态实体结构，这些实体属于一个静态集合。网站的主要焦点在于显示内容。现在的网站仍然以“内容为王”^③。即使使用那些创建“动态内容”的技术，实际上也只是说服务器现在能够基于一组不同的但

① http://en.wikipedia.org/wiki/2011_Egyptian_revolution

② <http://scobleizer.com/2009/02/09/is-the-real-time-web-a-threat-to-google-search/>

③ http://en.wikipedia.org/wiki/Web_content#Content_is_king

是定义过的参数和值来动态地生成静态内容。

我们使用那些应用程序来关注Internet或Web浏览器上的实体。这些应用程序主要聚焦于保证它满足最新的需求。这些需求包括下载和呈现HTML和图像，理解如何跟踪那些链接。在开始时，这就足够了。

传媒的形式被迫不断演进，网站也是如此。因为我们想要让我们的网站更好看，所以我们引入了CSS。因为我们想要网站对用户输入的交互性更好（你相信人们曾经需要为DHTML库付费吗？例如，下拉菜单），所以引入了JavaScript（让我们忘记VBScript的存在吧）。这些技术增强了Web浏览器的能力，但是，它们集中于增强网站的页面。

一些先驱者看到静态网站以外的机会。他们开始思考动态的Web应用程序。在Web应用程序中，关注点从服务器转移到了客户端。客户端需要做更多的事情：它需要动态地检索和载入内容，它可以基于用户反馈改变用户界面（User Interface，UI），UI能够以一种与桌面应用程序关联的方式呈现。它不再那么关注页面的重新载入，不再那么关注一般意义上的页面。在Web应用程序中，内容也变得不会那么基于文本，我们开始实现了更加吸引人的、更具交互性的表示。

1.1.2 HTTP解决方案

开发人员是技术的先驱。在越来越多的开发人员开始构建Web应用程序时，对Web浏览器的需求也增加了。这时，性能变成了问题。这种问题不仅由于Web浏览器应用程序，还与浏览器运行的机器有关。它们实实在在地推动网络技术和Web应用程序的发展，但是也遇到了一个重大的绊脚石：HTTP^①。

HTTP是一种协议。在设计之初，客户端可以使用该协议请求获得数据并接受响应。但是，一些Web应用程序开始要求信息从服务器发送到客户端，所以，我们需要主动进入客户端。主动进入客户端需要非标准的、复杂的解决方案。考虑到不同的Web浏览器支持的特性不同，你可以想象，要同时支持这么多种情况，问题必然比较复杂。（后文会涉及其中一些问题。）

当今流行的Twitter和Facebook采用了一些解决方案，这些解决方案可以用来演示实时

① en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

网络技术带来体验的好处和需求。在这种需求的推动下，实时网络技术得到了极大的改进，可用性大大增强。

1.1.3 一个先要解决的问题：实时究竟意味着什么

术语“实时”指的是在事件发生和我们意识到事件发生之间是即时的。从事件发生到事件传递的时间差实际上取决于事件。如果事件是用脚踩踏汽车的刹车，则在要用脚踩踏和刹车起作用之间的时间需要尽可能短。但是，如果事件是在一个足球论坛中发送一条聊天消息，使该消息显示给其他用户，则延迟几秒钟不会有太大影响。从根本上看，对于那些需要在足够短的时间里传递的事件，传递事件的时间还是有影响的；延时要使得在应用消息时消息的意义仍然在上下文中有效。不妨想象扇耳光的场景：在扇耳光和感到疼痛之间是没有延时的，这就是实时。如果在这个场景中有延时，就会令人相当困惑。

然而，在一开始为所有情况添加任何类型的实时体验特性并不是那么简单。但是，开发人员并没有被轻易打败。他们提出了一些聪明的变通方法，想出了解决服务器与客户端之间通信故障的方法。

注意：这里省略了一些与服务器双向通信的早期的方法，这是因为它们现在并不常使用。

1.1.4 AJAX

随着JavaScript开始流行，开发人员开始利用XMLHttpRequest对象^①来异步地（asynchronously）发送HTTP请求，而不需要重新载入当前页面。这称为异步JavaScript和XML（Asynchronous JavaScript and XML，AJAX）。

对于往网页应用中添加用户触发的功能而言，这种方法非常好用。从这个角度看，因为AJAX仍然典型地依赖于浏览器中的事件（例如，单击事件），所以它并没有真正解决关于请求内容实时更新的任何问题。

1.1.5 轮询

在使用AJAX后，离浏览器自动触发事件并处理新信息只有一步之遥了。开发人员设

① www.w3.org/TR/XMLHttpRequest/