

Join the discussion @ p2p.wrox.com



Wrox Programmer to Programmer™



Professional XMPP

Programming with JavaScript and jQuery

XMPP高级编程

——使用JavaScript和jQuery



(美) Jack Moffitt 著
杨明军 译

清华大学出版社

XMPP 高级编程—— 使用 JavaScript 和 jQuery

(美) Jack Moffitt 著
杨明军 译

清华大学出版社

北 京

Jack Moffitt
Professional XMPP Programming with JavaScript and jQuery
EISBN: 978-0-470-54071-8
Copyright © 2010 by Wiley Publishing, Inc.
All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 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: 01-2010-5494

本书封面贴有 Wiley 公司防伪标签, 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

XMPP 高级编程——使用 JavaScript 和 jQuery/ (美) 莫菲特(Moffitt,J.) 著; 杨明军 译.
—北京: 清华大学出版社, 2011.6

书名原文: Professional XMPP Programming with JavaScript and jQuery
ISBN 978-7-302-25630-4

I. X… II. ①莫… ②杨… III. 网络通信—程序设计 IV. TN915

中国版本图书馆 CIP 数据核字(2011)第 096085 号

责任编辑: 王 军 于 平

装帧设计: 孔祥丰

责任校对: 胡雁翎

责任印制: 何 芊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

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

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

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京鑫丰华彩印有限公司

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 27.25 字 数: 697 千字

版 次: 2011 年 6 月第 1 版 印 次: 2011 年 6 月第 1 次印刷

印 数: 1~4000

定 价: 58.00 元

产品编号: 038094-01

前 言

采用 XMPP 的应用程序范围广泛，包括即时通信、多用户聊天、语音和视频会议、协作空间、实时游戏、数据同步，甚至包括搜索。虽然 XMPP 起初是作为专有即时通信系统(比如 ICQ 和 AOL Instant Messenger)的开放的标准化替代技术，但它已经成长为一个极为健全的、适用于各类扣人心弦的创新应用程序的协议。

Facebook 在其聊天系统中使用了 XMPP 技术。Google 则采用 XMPP 来架构 Google Talk 以及它的令人兴奋的新型 Google Wave 协议。Collecta 基于 XMPP 的发布-订阅系统构建了一个实时搜索引擎。有几款 Web 浏览器正在试验使用 XMPP 作为同步和共享系统的基础。还有许多公司在它们的 Web 应用程序中使用 XMPP 来提供增强的用户体验和实时交互。

XMPP 的核心是小型结构化信息块的交换。与 HTTP 类似，XMPP 是一种客户端-服务器协议，但它与 HTTP 的不同之处在于，它允许任何一端向另一端异步发送数据。XMPP 使用长连接，数据以推(而不是拉)的形式发送。

由于 XMPP 的不同，使得它成为 HTTP 极佳的补充协议。采用 XMPP 的 Web 应用程序将能够实现 AJAX 提供给静态网站的所有功能，而且它们将实现进一步的交互性和动态性。JavaScript 和动态 HTML 已经将桌面应用程序功能带到 Web 浏览器，而 XMPP 将为 Web 带来新型通信机制。

由于其即时通信传统，XMPP 内置了许多常见的社交 Web 功能。联系人列表和订阅机制建立了社交图，出席更新机制可帮助用户了解别人在做什么，而个人通信机制可以保护用户之间通信的私密性。XMPP 还拥有将近 300 种扩展，这为我们构建复杂应用程序提供了范围广泛的、实用的工具。只须使用核心协议以及这些扩展中的少数几个，我们就可以构建出神奇的应用程序。

本书讲解如何在自己的应用程序中利用 XMPP 来构建具有社交功能的、协作式的、实时的应用程序。我们将开发一系列逐渐变复杂的 XMPP 应用程序，从“Hello, World!”开始，最后将开发出一个协作式文本编辑器、一个共享的涂鸦板以及一个实时的、多玩家游戏。最终，您将拥有使用 XMPP 构建下一代应用程序或向当前应用程序中添加新型的实时、推送或社交功能所需的所有工具。

目标读者

本书是为那些对开发 XMPP 应用程序感兴趣的开发人员而编写的。读者不需要具备任何 XMPP 经验，但如果先前有过这方面的经验，那么当然会对学习有所帮助。本书假设读者已经对 XMPP 有所耳闻，并希望深入研究。

本书中的所有应用程序均采用 JavaScript 语言开发，这是因为它易于理解，而且许多程序员比较熟悉它，每台带有 Web 浏览器的计算机都支持它。虽然本书使用 JavaScript，但所有的

概念和应用程序都可以使用任何语言开发，绝大多数“核心部分”与编程语言、所用的库以及 Web 浏览器没有任何关系。理解和处理本书中的代码并不要求读者是一位 JavaScript 专家。

本书假设读者理解基本的前端 Web 技术，即 CSS 和 HTML。如果您曾经手工编写过少量的 HTML 并修改过了 CSS 样式属性，那么您已经具有足够的经验了。

本书还使用两个库，即 jQuery 和 Strophe。如果以前曾经使用过 jQuery，那么会有所帮助，但如果没用过，那么可以参见附录 A 提供的一个简短入门指南。在开发应用程序的过程中，我们将充分讲解 Strophe 库。

涵盖内容

XMPP 协议及其扩展涵盖面非常广泛。本书关注那些得到广泛应用的 XMPP 部分。以下主题得到更多关注：

- XMPP 的即时通信功能，比如花名册、出席和订阅以及个人聊天
- XMPP 节、节错误消息、客户端协议语法和语义
- 扩展 XMPP 节
- 服务发现(XEP-0030)
- 数据表单(XEP-0004)
- 多人聊天(XEP-0045)
- 发布-订阅(XEP-0060)

尽管这些主题全部位于客户端，但几乎所有这些主题也同样适用于 XMPP 机器人或服务器组件和插件。

本书还涵盖 XMPP 编程相关的主题，比如应用程序设计、事件处理以及将简单的协议元素组合成一个更强大的整体。在这个过程中，我们还将讨论几个 Web 编程主题，比如 Canvas API。

XMPP 问世已经超过 10 年，现在已经相当成熟。本书内容涵盖核心协议的 1.0 版。对于 XMPP 协议的 future 版本，本书的 XMPP 协议部分应该仍然有效，就像 HTTP 1.0 客户端能够很容易与 HTTP 1.1 服务器通信一样。

XMPP 有很多扩展，本书也同样讨论了其中的一些扩展。本书所关注的大多是处于稳定成熟状态的扩展。对于每一个用到的扩展，给出了它们的文档编号，这样如果有疑问就可以检查该扩展的最新版本，看看是否已经改变或已被取代。

本书中的应用程序采用 jQuery 的 1.3 系列版和 jQuery UI 的 1.7 系列版。通常这些库在很大程度上保持向后兼容。此外本书使用了 Strophe 库的 1.0 版，但未来的 1.X 版也应该能够运行。

本书结构

本书内容主要按照一系列示例 XMPP 应用程序的指导教程来进行组织。每个应用程序在难度上逐渐增加，分别用来展示 XMPP 协议及其扩展的一个或多个有用部分。出于清晰考虑，这些应用程序经过缩减，但它们确实是 XMPP 开发人员每天建立的应用程序类型。

全书划分为 III 个部分。

第 I 部分介绍 XMPP 协议、它的用途以及 XMPP 应用程序的设计。第 1 章讲解了 XMPP 的用例、该协议的历史以及它的组成部分。第 2 章解释 XMPP 适用于哪些场合并深入研究了 XMPP 应用程序的工作原理，特别是在 Web 环境下。

第 II 部分是本书的重点，它包含了 9 个解决不同问题的 XMPP 应用程序。每个应用程序均要比前一个应用程序更加复杂，并且构建在前几个应用程序的概念之上。从第 3 章的简单的“Hello, World!” 示例开始，最终在第 11 章中构建一个实时的、多玩家游戏。

第 III 部分讲解了几个高级的但重要的主题。第 12 章讨论会话接入，这是一项有关安全、优化和持久化方面的技术。第 13 章深入研究如何最佳地部署和扩展基于 XMPP 的应用程序。第 14 章讲解如何使用 Strophe 的插件系统以及如何创建自己的插件。

使用本书的前提条件

本书使用 Web 技术，因此几乎不需要任何特殊的工具。我们可以在任何平台上使用、构建和运行本书中的应用程序。第 3 章讲解这些应用程序所需的库，而且大多数均可以使用而无须下载任何代码。

还将需要一种提供网页服务的方式，比如本地 Web 服务器或托管账号。如果不方便采用这些方式，那么可以使用 Tape 程序来提供文件服务。Tape 是一种简单的 Web 服务器，我们将在附录 B 中讲解。由于浏览器安全策略的需求，我们无法轻易地直接从本地文件系统中运行这些应用程序。

为了运行这些应用程序，还需要一个 XMPP 账号(如果要自行测试代码，那么在某些情况下还需要多个账号)。虽然可以借助任何公共 XMPP 服务器来完成这个任务，但还需要确保该服务器支持发布-订阅和多用户聊天功能，大多数服务器均支持这些功能。此外，还可以下载并运行自己的 XMPP 服务器，但本书并没有讲解这方面的内容。

第 12 章还需要在服务器端执行一些操作。该示例使用 Python 编程语言以及 Django 框架来完成这些操作。该章属于高级主题，本书中的普通应用程序并不需要它。

源代码

在研读本书示例的过程中，可以选择采用手工方式录入所有代码，也可以选择本书所附的源代码文件。本书中所用的所有源代码均可以从 <http://www.wrox.com> 和 <http://www.tupwk.com.cn/downpag> 下载。在该网站上，只要搜索本书的书名(可以通过搜索栏或使用书名列表)，然后在本书详细信息页面上单击 Download Code 链接来获取本书的源代码。

因为许多书籍都有着相似的书名，所以最简单的方式是按照 ISBN 搜索。本书的 ISBN 是 978-0-470-54071-8。

在下载代码之后，使用解压缩工具解压。或者，可以打开 Wrox 主代码下载页面 <http://www.wrox.com/dynamic/books/download.aspx>，查看本书以及所有其他 Wrox 书籍的代码。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，这还有助于提供更高质量的信息。

请给 wkservice@vip.163.com 发电子邮件，我们会检查您的反馈信息，如果是正确的，我们将在本书的后续版本中采用。

要在网站上找到本书的勘误表，可以登录 <http://www.wrox.com>，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上，单击 Book Errata 链接。在这个页面上可以查看 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是 <http://www.wrox.com/misc-pages/booklist.shtml>。

p2p.wrox.com

P2P 邮件列表是为作者和读者之间的讨论而建立的。读者可以在 p2p.wrox.com 上加入 P2P 论坛。该论坛是一个基于 Web 的系统，用于传送与 Wrox 图书相关的信息和相关技术，与其他读者和技术用户交流。该论坛提供了订阅功能，当论坛上有了新帖子时，会给您发送您选择的主题。Wrox 作者、编辑和其他业界专家和读者都会在这个论坛上进行讨论。

在 <http://p2p.wrox.com> 上有许多不同的论坛，帮助读者阅读本书，在读者开发自己的应用程序时，也可以从这个论坛中获益。要加入这个论坛，必须执行下面的步骤：

- (1) 进入 p2p.wrox.com，单击 Register 链接。
- (2) 阅读其内容，单击 Agree 按钮。
- (3) 提供加入论坛所需的信息及愿意提供的可选信息，单击 Submit 按钮。
- (4) 然后就可以收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。

提示：

不加入 P2P 也可以阅读论坛上的信息，但只有加入论坛后，才能发送自己的信息。

加入论坛后，就可以发送新信息，回应其他用户的帖子。可以随时在 Web 上阅读信息。如果希望某个论坛给自己发送新信息，可以在论坛列表中单击该论坛对应的 **Subscribe to this Forum** 图标。

对于如何使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作原理，以及许多针对 P2P 和 Wrox 图书的常见问题解答。要阅读 FAQ，可以单击任意 P2P 页面上的 FAQ 链接。

目 录

第 I 部分 XMPP 协议和架构	
第 1 章 了解 XMPP 协议.....3	
1.1 什么是 XMPP.....3	
1.2 XMPP 简史.....5	
1.3 XMPP 网络.....6	
1.3.1 服务器.....6	
1.3.2 客户端.....6	
1.3.3 组件.....6	
1.3.4 插件.....7	
1.4 XMPP 寻址.....7	
1.5 XMPP 节.....8	
1.5.1 通用属性.....9	
1.5.2 presence 节.....10	
1.5.3 message 节.....12	
1.5.4 IQ 节.....14	
1.5.5 error 节.....15	
1.6 连接生命周期.....16	
1.6.1 连接.....17	
1.6.2 流的建立.....17	
1.6.3 身份验证.....18	
1.6.4 连接断开.....18	
1.7 小结.....19	
第 2 章 设计 XMPP 应用程序.....21	
2.1 他山之石.....21	
2.2 XMPP 与 HTTP 的比较.....24	
2.2.1 XMPP 的优势.....24	
2.2.2 XMPP 的不足.....25	
2.3 桥接 XMPP 与 Web.....27	
2.3.1 长轮询.....27	
2.3.2 管理连接.....28	
2.3.3 让 JavaScript 理解 XMPP 协议.....28	
2.4 构建 XMPP 应用程序.....29	
2.4.1 浏览器平台.....29	
2.4.2 基本的基础设施.....29	
2.4.3 协议设计.....30	
2.5 小结.....31	
第 II 部分 应用程序	
第 3 章 Hello World: 第一个应用程序.....35	
3.1 应用程序预览.....35	
3.2 Hello 应用程序设计.....36	
3.3 准备.....36	
3.3.1 jQuery 与 jQuery UI.....37	
3.3.2 Strophe.....38	
3.3.3 flXHR.....38	
3.3.4 XMPP 账户.....38	
3.4 开始构建第一个应用程序.....39	
3.4.1 用户界面.....39	
3.4.2 应用程序代码.....40	
3.5 建立连接.....43	
3.5.1 连接生命周期.....43	
3.5.2 创建连接.....43	
3.5.3 连接 Hello.....44	
3.5.4 运行应用程序.....48	
3.6 创建节.....48	
3.6.1 Strophe 构建器.....49	
3.6.2 打招呼.....51	
3.7 处理事件.....52	
3.7.1 添加和删除处理程序.....52	
3.7.2 节匹配.....52	
3.7.3 节处理程序函数.....53	

3.7.4 处理 Hello 响应.....	54	第 6 章 与好友交谈：一对一聊天.....	99
3.8 给 Hello 程序添加新功能.....	57	6.1 应用程序预览.....	99
3.9 小结.....	57	6.2 Gab 的设计.....	100
第 4 章 探索 XMPP 协议：一个调试		6.2.1 出席.....	100
控制台.....	59	6.2.2 消息.....	101
4.1 应用程序预览.....	59	6.2.3 聊天区域.....	101
4.2 设计 Peek.....	60	6.2.4 花名册区域.....	101
4.3 构建控制台.....	61	6.3 制作界面.....	101
4.3.1 用户界面.....	61	6.4 构建花名册.....	105
4.3.2 显示流量.....	63	6.4.1 请求花名册.....	106
4.3.3 美化 XML.....	66	6.4.2 处理 IQ.....	109
4.3.4 处理 XML 输入.....	69	6.4.3 更新出席状态.....	110
4.3.5 简化输入.....	71	6.4.4 添加新联系人.....	111
4.4 研究 XMPP.....	76	6.4.5 响应花名册变化.....	112
4.4.1 控制出席.....	76	6.4.6 处理订阅请求.....	114
4.4.2 探测版本.....	77	6.5 构建聊天对话.....	117
4.4.3 处理错误.....	78	6.5.1 处理标签页.....	117
4.5 更好的调试.....	80	6.5.2 创建新的聊天对话.....	118
4.6 小结.....	80	6.5.3 发送消息.....	119
第 5 章 实时微博：一个 Identica		6.6 即时通信最佳实践.....	122
客户端.....	83	6.6.1 理解消息路由.....	122
5.1 应用程序预览.....	83	6.6.2 更好地寻址消息.....	123
5.2 Arthur 的设计.....	84	6.7 添加活动通知.....	125
5.3 Identica 微博.....	84	6.7.1 理解聊天状态.....	125
5.3.1 建立账户.....	85	6.7.2 发送通知.....	126
5.3.2 开启 XMPP.....	85	6.7.3 接收通知.....	127
5.4 构建 Arthur.....	86	6.8 收尾工作.....	128
5.4.1 开始.....	86	6.9 更多 Gab 功能.....	138
5.4.2 接收消息.....	89	6.10 小结.....	138
5.5 XHTML-IM.....	90	第 7 章 探索服务：服务发现与浏览.....	141
5.5.1 将 XHTML-IM 添加到 Arthur.....	91	7.1 应用程序预览.....	141
5.5.2 发送消息.....	91	7.2 Dig 的设计.....	142
5.6 离线消息.....	93	7.3 查找信息.....	142
5.7 创建更好的微博客户端.....	96	7.3.1 disco#info 查询.....	142
5.8 小结.....	97	7.3.2 disco#items 查询.....	144
		7.3.3 disco 节点.....	144

7.4	创建 Dig	145	9.2.1	一切都是 pubsub	196
7.4.1	初始 disco 查询	148	9.2.2	展示者的流程	197
7.4.2	浏览 disco 树	151	9.2.3	观众的流程	197
7.5	挖掘服务	156	9.3	填写表单	198
7.5.1	查找代理服务	156	9.3.1	Data Forms 扩展	198
7.5.2	发现功能	157	9.3.2	表单元素、字段和类型	199
7.5.3	寻找聊天对话	157	9.3.3	标准化的表单字段	201
7.6	服务发现的更多功能	158	9.4	处理 pubsub 节点	202
7.7	小结	158	9.4.1	创建节点	202
第 8 章	群聊: 多人聊天客户端	159	9.4.2	配置节点	204
8.1	应用程序预览	159	9.4.3	pubsub 事件	206
8.2	Groupie 的设计	160	9.4.4	发布到节点	206
8.3	公开发言	161	9.4.5	订阅和退订	207
8.3.1	群聊服务	161	9.4.6	检索订阅情况	210
8.3.2	进入和离开房间	161	9.4.7	获取项	210
8.3.3	发送和接收消息	164	9.4.8	订阅管理	212
8.3.4	匿名性	165	9.5	使用 pubsub 广播绘图	213
8.3.5	创建房间	165	9.5.1	构建用户界面	214
8.3.6	理解角色和岗位	166	9.5.2	使用 Canvas 绘制草图	216
8.4	构建界面	168	9.5.3	登录并建立节点	219
8.5	加入房间	172	9.5.4	发布和接收绘图事件	225
8.6	处理出席和消息	175	9.6	改进 SketchPad	240
8.6.1	处理房间消息	176	9.7	小结	240
8.6.2	跟踪出席状态变化	178	第 10 章	与好友一同写作: 协作式	
8.6.3	聊天历史	179		文本编辑器	243
8.6.4	保持私密性	180	10.1	应用程序预览	243
8.6.5	描述动作	182	10.2	NetPad 的设计	244
8.7	管理房间	184	10.3	操作转换	245
8.7.1	更换主题	184	10.3.1	基本原理	245
8.7.2	处理麻烦制造者	185	10.3.2	算法细节	246
8.7.3	招募管理员	187	10.3.3	实现	248
8.8	改进 Groupie	194	10.4	扩展 XMPP 协议	258
8.9	小结	194	10.4.1	忽略未知数据	258
第 9 章	发布与订阅: 共享画板简介	195	10.4.2	XML 命名空间	258
9.1	SketchCast 预览	196	10.4.3	扩展元素	259
9.2	SketchCast 的设计	196	10.4.4	扩展属性	261

10.4.5	贡献扩展	261
10.5	设计协议	262
10.5.1	测试支持	262
10.5.2	请求和控制会话	263
10.5.3	编辑操作	264
10.6	构建编辑器	265
10.6.1	初始骨架	265
10.6.2	启动编辑会话	268
10.6.3	谈论工作	275
10.6.4	进行编辑	278
10.7	扩展 NetPad	288
10.8	小结	288
第 11 章	玩游戏：面对面的	
	Tic-Tac-Toe	291
11.1	应用程序预览	291
11.2	Toetem 的设计	293
11.3	设计游戏协议	294
11.3.1	跟踪用户	294
11.3.2	管理玩家	295
11.3.3	管理游戏	297
11.3.4	玩游戏和观看游戏	299
11.4	Toetem 起步	302
11.5	实现会话和等待列表	308
11.5.1	裁判员(版本 1)	308
11.5.2	Toetem 客户端(版本 1)	313
11.6	实现游戏管理	316
11.6.1	裁判员(版本 2)	316
11.6.2	Toetem 客户端(版本 2)	324
11.7	实现游戏逻辑	329
11.7.1	Tic-Tac-Toe 库	329
11.7.2	裁判员(版本 3)	333
11.7.3	Toetem 客户端(版本 3)	336
11.8	让游戏更有趣	363
11.9	小结	363

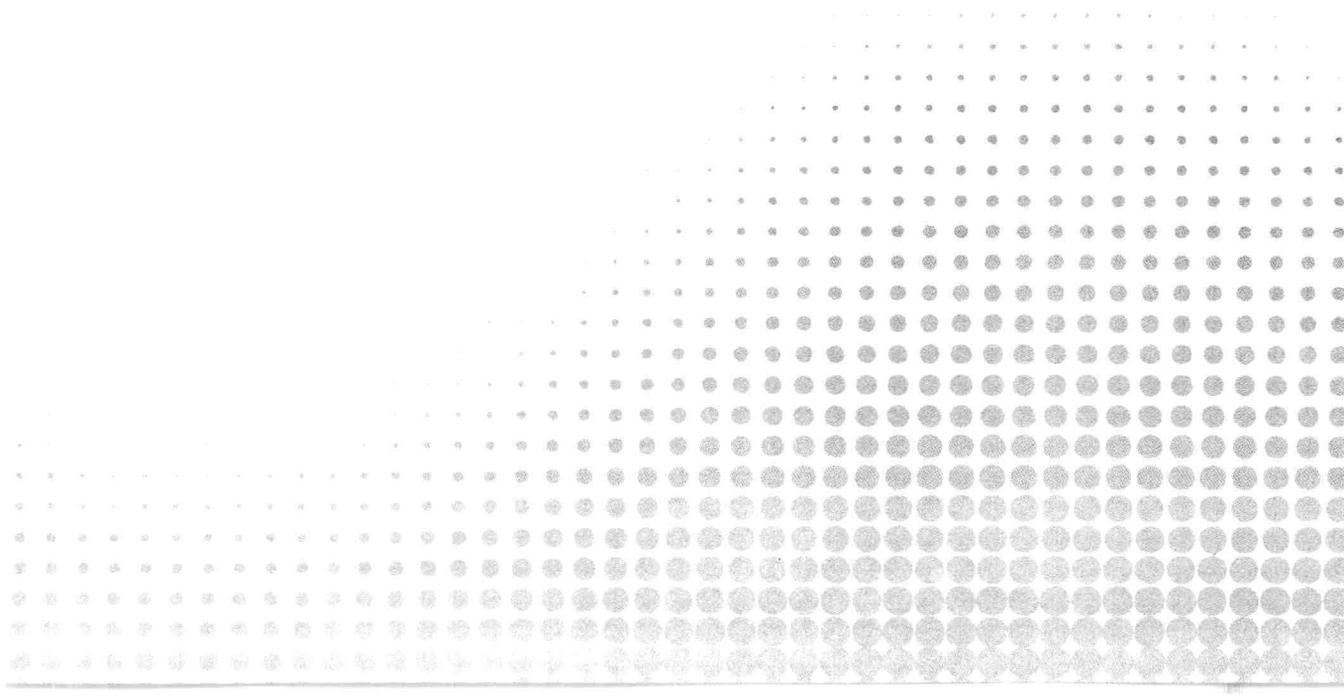
第 III 部分 高级主题分构

第 12 章	加入已有会话：引导 BOSH	367
12.1	会话接入	367
12.1.1	会话技术	368
12.1.2	用例	369
12.2	利用会话接入实现自动登录	370
12.3	小结	374
第 13 章	部署 XMPP 应用程序	375
13.1	横向扩展	375
13.1.1	多个连接管理器	376
13.1.2	XMPP 服务器集群	378
13.1.3	扩展组件	380
13.1.4	内部联合	380
13.1.5	成为服务器	380
13.2	纵向扩展	381
13.2.1	减少延迟	381
13.2.2	尽量减少 XML 序列化	383
13.2.3	优化 DOM 操作	384
13.3	小结	385
第 14 章	编写 Strophe 插件	387
14.1	使用插件	388
14.1.1	加载插件	388
14.1.2	访问插件功能	388
14.2	构建插件	389
14.3	创建花名册插件	390
14.3.1	保存联系人	390
14.3.2	获取并维护花名册	392
14.3.3	操纵花名册	397
14.4	试用插件	398
14.5	改进花名册插件	402
14.6	小结	402
附录 A	jQuery 入门	405
附录 B	设置 BOSH 连接管理器	415

第 I 部分

XMPP协议和架构

- 第 1 章 了解 XMPP 协议
- 第 2 章 设计 XMPP 应用程序



第 1 章

了解 XMPP 协议

本章内容

- XMPP 历史
- XMPP 网络和连接
- XMPP 的三种构造块节

从最基本的层次来说，XMPP(eXtensible Messaging and Presence Protocol，可扩展消息处理和现场协议)是一种用来在两个地点之间传递小型的结构化数据段的协议。在此基础上，XMPP 已被用来构建大规模即时通信系统、Internet 游戏平台、搜索引擎、协作空间以及语音和视频会议系统。每天都会出现更独特的应用程序，这进一步说明了 XMPP 协议功能是如此多样和强大。

XMPP 由几个小的构造块组成，并已经在这些原语的基础之上构建了许多更大的构造。在 XMPP 中有众多系统：构建发布-订阅服务、多人聊天、表单检索和处理、服务发现、实时数据传输、隐私控制以及远程过程调用。XMPP 程序员通常会创建自己的、独特的、完全为手头上问题量身定做的构造。

大多数社交媒体构造(它们已经将网站(比如 Facebook、MySpace 和 Twitter)推向最前端)也都采用 XMPP 协议。在 XMPP 内部，可以找到由联系人组成的花名册，可以用它们来创建定向或无向社交图。当联系人上线或下线时自动发送出席通知，而个人消息和公共消息是 XMPP 系统的基础应用。有时候开发人员之所以选择 XMPP 作为底层技术层只是因为它已经为开发人员提供了许多现成的社交功能，他们只需要关注自己的应用程序的一些独特方面即可。

XMPP 为我们带来了无限的可能，但在开始之前，我们还需要了解 XMPP 的各种组成部分以及它们如何有机地构成一个整体。

1.1 什么是 XMPP

与所有协议一样，XMPP 定义了两个或更多通信实体之间传递数据所采用的格式。对于 XMPP，实体通常是客户端和服务端，但它也允许两台服务器或两个客户端之间进行对等通信。Internet 上有许多 XMPP 服务器，彼此之间能够相互访问，从而构成了一个由互联系统组成的

联合网络。

在 XMPP 上交换的是 XML 数据，这让通信具有丰富的、可扩展的结构。为了获得更具人类可读性(因此更易于调试)这项更实用的功能，许多现代协议都放弃了二进制编码带来的带宽节省的优势。XMPP 选择使用 XML，这意味着，它能够充分利用大量与处理 XML 有关的知识和支持软件。

XMPP 通过使用 XML 而获得的一项主要功能就是 XML 的可扩展性。向 XMPP 协议中添加能同时保持向前和向后兼容的新功能是一件极其简单的事情。在已向 XMPP Standards Foundation 注册的 200 个协议扩展中，这种可扩展性得到了大量运用，它为开发人员提供了丰富的、实际上没有限制的工具。

XML 主要以文档格式闻名，但在 XMPP 中，XML 数据被组织成一对流，每个流分别用于通信的一个方向。每个 XML 流均由一个开始元素、后跟 XMPP 节和其他顶级元素、然后是一个结束元素组成。每个 XMPP 节(可带有子元素和属性)均是该流的一级子元素。在 XMPP 连接末尾，这两个流形成了一对有效的 XML 文档。

XMPP 节构成了该协议的核心部分，而 XMPP 应用程序则关注如何发送和响应各种类型的节。节可能包含网络上其他实体的信息、类似于电子邮件的个人消息或为计算机处理而设计的结构化通信数据。下面给出了一个示例节。

```
<message to='elizabeth@longbourn.lit'  
        from='darcy@pemberley.lit/dance'  
        type='chat'>  
  <body>What think you of books?</body>  
</message>
```

在一个典型的客户端-服务器 XMPP 会话中，一个与此类似的由 Elizabeth 发给 Darcy 的节将从 Elizabeth 的客户端传送到她的服务器。她的服务器将注意到该节的目的地是某个远程服务器上的一个实体，因此它与该远程服务器建立 XMPP 连接，并将该消息转发到该处。服务器之间的通信与电子邮件网络类似，但与电子邮件服务器不同的是，XMPP 服务器之间总是能够直接通信，而不需要借助中间服务器。

这种直接通信省去了垃圾消息和未经授权消息的常见传播媒介。这也是众多 XMPP 安全设计中的一环。它还支持通信端点之间通过使用 TLS(Transport Layer Security, 传输层安全)来加密通信并通过 SASL(Simple Authentication and Security Layers, 简单身份验证和安全层)实现强有力的身份验证机制。

XMPP 是为交换短小信息片段而设计的，而不是针对大型二进制数据块。但 XMPP 能够用来协商并建立可在端点之间传递大型数据块的带外或带内传输。对于这种传送而言，XMPP 相当于信令层。由于专注于小型结构化数据块，XMPP 协议的延迟极短，这使得它极其适用于实时应用程序。在开发人员试验实时 Web 时，这些应用程序(包括协作空间、游戏以及同步)正驱动着 XMPP 变得越来越流行。

我们将从本书中看到建立实时 Web 应用程序是多么简单的事情。当读完本书，读者应该对为什么如此多的人都被 XMPP 的强大功能所折服有一个透彻的理解。

1.2 XMPP 简史

XMPP 协议问世已经超过十年时间，从最初的简陋功能一路发展到今天，它已经取得了很大的进展。XMPP 的大部分设计均缘于 XMPP 创建时所处的环境，而 XMPP 的历史为我们了解开放协议如何被采用和创新提供了一个有趣的案例。

1996 年 Mirabilis 发布了 ICQ，这使得 Internet 用户之间能够进行快速的个人通信。ICQ 得以快速传播，而当时并没有其他公司发布类似的产品。1997 年 AOL 发布了 AOL Instant Messenger。Yahoo 也紧随其后于 1998 年发布了 Yahoo Pager(最终改名为 Yahoo Messenger)，而微软也最终于 1999 年凭借 MSN Messenger(现在的 Windows Live Messenger)加入竞争行列中。

所有这些即时通信应用程序都各自绑定到它们公司运营的专有协议和网络。ICQ 的用户不能与 Yahoo 用户交谈，反过来也一样。因此，常常出现这种情况：用户运行多种即时通信应用程序，这样才能够与自己的所有联系人聊天，因为尚无一家公司宣称占据 100% 的市场份额。

不久之后，开发人员期望为这些专有 IM 网络编写自己的客户端。有些开发人员希望编写能够联合两种或更多种 IM 网络的多协议客户端，还有一些开发人员希望将这些应用程序移植到微软的 Windows 以及 Apple 的 Mac OS 之外的操作系统。这些开发人员遇到了许多障碍，他们必须对这些没有文档资料的协议进行逆向工程，而 IM 网络则主动频繁地修改协议以挫败第三方开发人员的努力。

开放的、去中心化(decentralized)的 IM 网络和协议的思想正是在这样的气候中诞生了。

Jeremie Miller 于 1999 年 1 月发布了 Jabber 项目。Jabber 是一种基于 XML 的去中心化的即时通信协议，同时也是一个名为 jabberd 的服务器实现。立即形成了一个围绕该协议和实现的社区，并不断衍生出越来越多的客户端和新想法。在 2000 年 5 月，Jabber 的核心协议得以稳定，而 jabberd 被正式发布。

JSF(Jabber Software Foundation, Jabber 软件基金会)于 2001 年成立，旨在协调围绕 Jabber 协议及其实现而展开的各项工作。在 2002 年年底，JSF 已经向 IETF 标准化进程提交了核心协议规范，并成立了一个 IETF 工作小组。2004 年 10 月，这个标准化进程产生了 Jabber 协议的改进版，改名为 XMPP，其文档成为 RFC 标准，编号分别为 3920、3921、3922 和 3923。

在该协议的早期，开发人员持续通过向 JSF 提交协议扩展的方式来扩展它的可能性。这些扩展被称为 JEP(Jabber Extension Proposal, Jabber 扩展提议)。最终 JSF 及其扩展遵循 Jabber 到 XMPP 的名称改变，变成了 XSF(XMPP Standards Foundation, XMPP 标准基金会)和 XEP(XMPP Extension Proposal, XMPP 扩展提议)。

2005 年 XMPP 技术的大规模部署已经展开，比较突出的就是 Google Talk(Google 基于 XMPP 的 IM 服务)的发布。

现今 XMPP 生态系统已经相当庞大。已经有接近 300 种扩展被接受作为 XEP，并且创建了数十种客户端和服务器实现，既有商业产品也有开放源代码产品。实际上使用任何编程语言的软件开发人员都能够找到一个库来加速 XMPP 应用程序的开发进程。

XMPP 应用程序起初是以 IM 为中心的，这反映出它的出身，但开发人员已经发现 XMPP 能轻松胜任多种最初未能预见的应用程序，包括搜索引擎和同步软件。这种效用确实证明了开放系统和开放标准化进程的强大威力。

最近, IETF 已经组成了一个新的 XMPP 工作组来准备 XMPP 规范的下一版本, 把自从最初的 RFC 发布以来人们获取的所有认识整合起来。XMPP 不断得到精炼和扩展, 这样应用程序开发人员和 Internet 用户将总能够拥有一个开放的、去中心化的通信协议。

1.3 XMPP 网络

任何 XMPP 网络都是由若干角色组成的。这些角色可分为服务器、客户端、组件和服务器插件。XMPP 开发人员将编写代码来创建或修改这些角色类型中的一种。每种角色均在 XMPP 网络中发挥着各自的作用。

1.3.1 服务器

XMPP 服务器(更精确地说是使用服务器-服务器协议的 XMPP 实体或客户端-服务器协议中的服务器端)是任何 XMPP 网络的循环系统。服务器的任务就是为 XMPP 节提供路由, 无论这些节是从内部的一个用户发往另一个用户, 还是从本地用户发往远程服务器。

一组能够相互进行通信的 XMPP 服务器构成了一个 XMPP 网络。一组公开的 XMPP 服务器形成了全局的 XMPP 联合网络。如果某个服务器不能使用服务器-服务器协议, 那么它将变成一个孤岛, 不能与外部服务器通信。

XMPP 服务器总是允许用户连接到自己。但也可以编写直接使用服务器-服务器协议的应用程序或服务, 通过减轻路由开销来提高效率。

任何人都能够运行 XMPP 服务器, 而且几乎所有平台都有全功能的服务器可用。Ejabberd、Openfire 和 Tigase 是三种能够在 Windows、Mac OS X 和 Linux 系统上运行的流行的开放源代码服务器。还有几个商业 XMPP 服务器可用, 包括 M-Link 和 Jabber XCP。

1.3.2 客户端

大多数 XMPP 实体均是客户端, 它们通过客户端-服务器协议连接到 XMPP 服务器。这些实体中许多都是由人类驱动的、传统的 IM 用户, 但也有一些以机器人形式运行的自动化服务。

客户端必须向某个地方的 XMPP 服务器进行身份验证。服务器将该客户端发送的所有节路由到适当的目的地。服务器还负责管理客户端会话的其他几个方面, 包括它们的花名册以及裸地址, 稍后我们就会看到。

本书中的所有应用程序均被编写为客户端应用程序。这通常是大多数 XMPP 开发的起点。对于不需要用户参与或者有着苛刻要求的应用程序, 更好的做法通常是创建一个不同类型的实体, 比如服务器组件。

1.3.3 组件

并不只有客户端能够连接到 XMPP 服务器, 大多数服务器还支持外部服务器组件。这些组件通过添加某种新服务来增强服务器的行为。这些组件在服务器内部有各自的身份和地址, 但在外部运行并通过组件协议通信。

组件协议(由 XEP-0114 定义)可以让开发人员以一种服务器不可知的方式创建服务器扩展。