

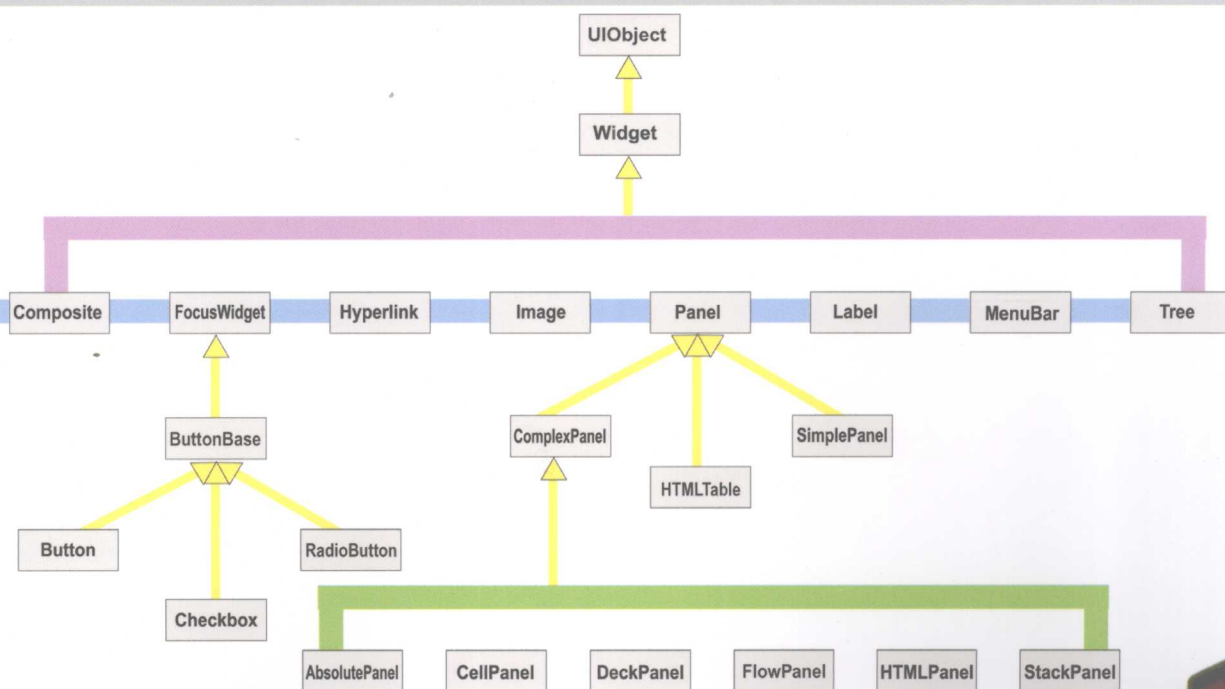
Google Web Toolkit

开发实战

Google Web Toolkit Solutions: More Cool & Useful Stuff

(美) David Geary Rob Gordon 著

李松峰 李维鑫 译



机械工业出版社
China Machine Press



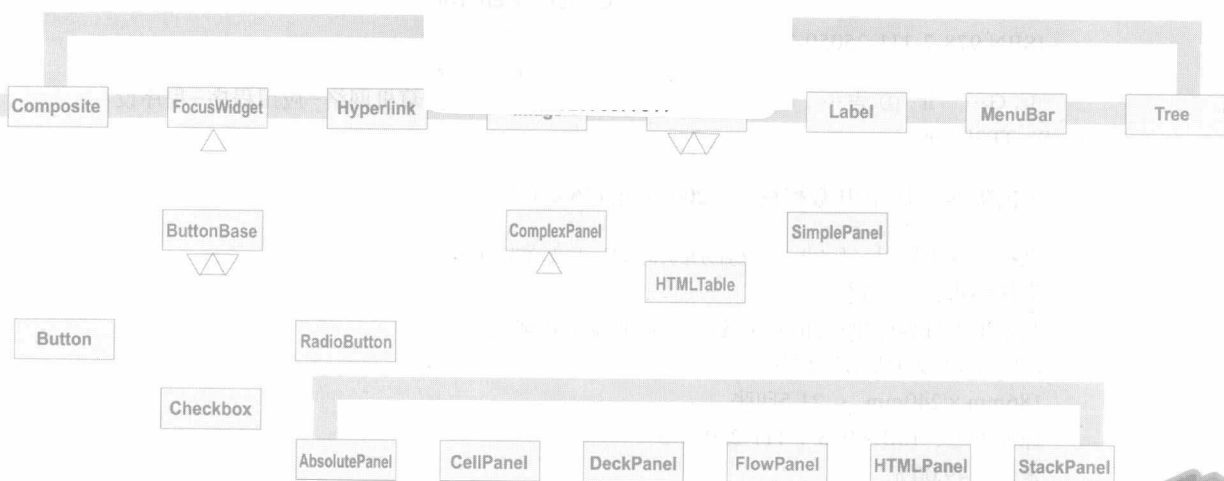
Google Web Toolkit

开发实战

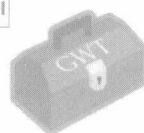
Google Web Toolkit Solutions: More Cool & Useful Stuff

(美) David Geary Rob Gordon 著

李松峰 李维鑫 译



机械工业出版社
China Machine Press



本书介绍了Google Web Toolkit各种前沿应用及其使用技巧和方法,例如实现拖放、整合JavaScript库,以及使用高级事件处理机制等。本书提供了精心准备的各种示例,其中不仅融合了常见的模式,甚至还揭示出许多可能的缺陷。书中每个解决方案都可以在实际项目中加以重用。

本书适用于GWT中级用户和Web站点的开发人员。

Simplified Chinese edition copyright © 2008 by Pearson Education Asia Limited and China Machine Press.

Original English language title: Google Web Toolkit Solutions: More Cool & Useful Stuff (ISBN 978-0-13-234481-4) by David Geary, Rob Gordon, Copyright © 2008.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall.

本书封面贴有Pearson Education(培生教育出版集团)激光防伪标签,无标签者不得销售。

版权所有,侵权必究。

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:01-2008-1795

图书在版编目(CIP)数据

Google Web Toolkit开发实战 / (美) 吉尔里 (Geary, D.), (美) 高登 (Gordon, R.) 著; 李松峰, 李维鑫译. —北京: 机械工业出版社, 2009.1

(Google核心技术丛书)

书名原文: Google Web Toolkit Solutions: More Cool & Useful Stuff

ISBN 978-7-111-25059-3

I. G… II. ① 吉… ② 高… ③ 李… ④ 李… III. 计算机网络—应用程序—程序设计 IV. TP393.09

中国版本图书馆CIP数据核字(2008)第136503号

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: 李东震

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2009年1月第1版第1次印刷

186mm × 240mm · 21.5印张

标准书号: ISBN 978-7-111-25059-3

定价: 49.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换
本社购书热线 (010) 68326294

译者序

2008年9月3日，Google发布了开源的Chrome浏览器。这款浏览器基于开源的Webkit解释引擎开发，内置了全新的JavaScript引擎V8。经过几天的试用，译者个人感觉还是不错的。至少，Chrome酷似另一个“桌面”的标签式浏览外观与IE、Firefox、Opera，以及它的近亲Safari都有着明显不同。全新的界面不禁令人产生了对Google网络操作系统的期待，同时也让用时整整5年才将IE 6升级到IE 7的微软的这一“重大决策失误”（比尔·盖茨语）落到了实处。

随着Chrome浏览器的诞生，特别是它“创建应用程序快捷方式”的功能，让普通用户能够更容易体会出Web应用程序——特别是使用GWT创建的Web应用程序的含义。GWT是基于Java的一款Ajax应用程序开发框架。有了这个框架，Java（及其他面向对象语言的）开发人员可以不必学习JavaScript，就能轻松地创建Ajax应用程序。而且在开发过程中，不必重新编译即可在GWT的主机模式下实时查看代码变化。虽然在产品状态下所有代码都将被编译成JavaScript，但在开发期间这些代码则完全以字节码形态运行在Java虚拟机中。这就意味着开发Java程序时可以使用任何调试工具和手段（如设置断点和单步调试）对GWT代码也都完全适用。此外，GWT对编译和部署所作的优化，也能确保Web应用程序实现跨浏览器兼容。

本书是畅销书作者David Geary倾力打造的一册GWT实战指南。全书通过众多实例展示了GWT开发中涉及的方方面面，例如自定义GWT部件、整合Hibernate、扩展GWT内置类、整合Script.aculo.us框架等。所有实例由浅入深，层层递进，翔实、全面、深入地剖析了GWT开发的初、中、高级主题。

希望读者通过阅读本书能够有所收获。

本书由李松峰负责编译，参加翻译工作的还有李维鑫、李丽、宋会敏、封耀杰、贾爱华、杨文芹、左家安、付艳芬、宋连海、刘英、左洪静、陈国良、闫振波、孙淑霞等。

译者

序

Ajax的成功非常耐人寻味。既没有经过特别地发布，也没有产品或报道让人们知道有一种技术叫Ajax。好像是一夜之间，Ajax就出现了。尽管Jesse James Garret写了文章为这种技术取了名字，但该文也没有声称这是他的创造，而是指出它是一种值得关注的新趋势。结果，虽然我们都意识到了它的存在，但对于“Ajax”到底意味着什么却难以达成共识。对此，20位专家可能会有22种不同的解说。假如非要我概括一下他们的意见，那么我只能说它是过去15年来软件领域中最好的，同时也是最差的一种技术。然而，无论人们是否知道它的起源，它是什么，或者它的优点和缺点，但每个软件从业人员似乎都想在自己的下一个产品中加入平滑的新Ajax界面。从技术人员的角度看，这并不意味着是一件简单的事情。浏览器普遍存在兼容性问题，而且其不一致性也给开发人员带来了无尽的痛苦。于是，大家都从自己的角度提出了正确的意见，但反映的问题却让人无法乐观。

不过，值得欣慰的是Ajax并非只有技术人员在推动。还有另一股力量对技术人员结构之美的困惑起到了调和作用——即用户越来越强烈地认为软件应该简化生活，而不是让生活复杂化。虽然技术人员抱怨浏览器是软件开发中的混沌地带，但用户却因为自己熟悉和适应浏览器而对它推崇倍至。对用户而言，浏览器把Internet连接的全部复杂性简化为几个关键概念：地址、链接、前进、后退和搜索。这就是Web中的语言，而且用户对这种具有局限性的技术图景却乐在其中。他们为什么不应该如此呢？除了要熟悉各种浏览器之外，额外的约束也强迫软件开发人员更关注用户的需求。成功使用Ajax开发的一个重要标志，就是无论你使用什么技术，但结果必须对用户有价值。虽然这对于推动应用程序开发的创新明显具有积极意义，但软件开发人员必须为此付出的代价，则是要承受更多的痛苦和磨难。

在我听说的新项目中，“当然，它的UI要设计成非常Ajax的风格”这样的论调越来越多。一般情况下，通过他们对项目的深爱程度，我都能推测出项目进展。而那些刚刚参与到项目中的人，则会因为能在倍受关注的领域内工作而激动不已。他们会讨论选择某个框架的过程，有时也会粗略地聊一聊早期UI模型的情况。此时此刻，他们已经构建了一些小例子，一切好像都按部就班。不过，也很容易看出基于浏览器开发的困难对他们来说并没有那么夸张，反而对攻克技术难关的满足感倒是溢于言表。于是，他们决定品尝一种时尚的浓咖啡来奖励自己。几个月后，当我再遇到他们时，发觉他们对那个项目欲言又止。然而，他们最终还是向我透露，开始时对某些问题的难度估计不足，而事实证明挑战还不小。“我们本来是想让它实时更新，但谁知道加载要花那么多时间”。此时，我的初步判断就是他们的信心开始低落，但我还是尽力

鼓励他们。不过，我知道一场耐力赛才刚刚拉开帷幕。这时候，我通常会建议他们别再喝浓咖啡了，而是改喝茶。与咖啡因相比，茶更温和且具有持久力；咖啡却总爱在关键时刻掉链子。简单地说，要为好的Ajax应用程序画上完美的句号，必须有足够的时间。虽然他们都对接近真实挑战的终点持乐观态度，但却往往对我的建议一笑置之。

项目真正的灾难开始于出现如下问题：我们的用户在想什么？在试图回答这个问题时，多数人都想通过用户测验，或者通过发布早期的beta版来找到答案。另外还有一些人，——这些人大概吃水果之前都不会先洗一下，他们会基于自己认为的对用户需求的精确把握，提前发布完整版。我觉得这些人最可怜。事实上，在回答这个问题时，上述两类开发人员存在一些共同问题。首先，他们作出的避免涉及某些较难问题的决定，对用户而言实际上是错误的。其次，他们认为任何测试都比不上实际使用。目前，类似的认识并不仅仅存在于UI内置的Ajax上面。我不知道有哪个成功的项目可以绕过这些难题。但对Ajax应用程序而言，独有的一个特点就是这些问题越来越难以解决好。因为，需要闯过的关口实在是太多了。

首先要面对的恐怕就是浏览器间行为的不一致性。用户会报告说在一个浏览器中他们的菜单显示在了错误的位置上。在另外一个浏览器中，文本又发生了折行。但在第三种浏览器中，虽然不存在前两个问题，但却在加载应用程序15分钟后，由于某些内容的变化，导致整个界面出现了闪烁现象。其次，你可能会发现用户的使用模式不对头，而且UI中的某些部分还必须得改变——如果不是出现讨厌且莫名其妙的问题，这些改变通过JavaScript的灵活表达性来实现也确实会让人折服。我也曾在很多情况下尝试无缝地重构大型JavaScript代码基，但没有一次令人满意。在这些错综复杂的问题之上，只有一个硬道理：应用程序必须让用户满意。因此，除了刚刚介绍的这些直接困难之外，用户体验最大化这个主要目标仍然未变。而且，当应用程序运行后，这也是人们评估它的唯一标准。你的应用程序能够满足用户需要吗？

简单地说，这正是Google Web Toolkit的使命——让开发人员在面对这些问题时，能够更轻松地说“没问题”。我们越来越讨厌攻击令人头疼的Ajax开发，特别是忿忿地抱怨浏览器，因为这最终毫无结果，更得不到什么好处。不过，这样却会让你知道什么对用户最好，但你却无法真正实现——因为这就相当于连一把好用的锤子都没有，却幻想着建设一套房子。GWT充分利用了已有的工具。例如，软件工程中就有不少好用的“锤子”，而我们就非常奇怪，怎么就没有有人在Ajax开发中有效地使用它们呢。我们坚定地认为，确保Web应用程序持续改进的正确方法，就是利用优秀的软件工程工具和已有的最佳实践。GWT可以让开发人员通过Java来编写Ajax代码，也利用了UI开发人员耳熟能详的各种概念和模式；在开发中使用可验证的开发环境，包括像Eclipse这样具有代码完成和重构工具的IDE；在真实的浏览器中运行并调试应用程序，并使用可靠的调试器；然后，再通过编译器将全部Java代码转换成小型化但却高性能的JavaScript代码。而且，这些代码可以应对绝大多数浏览器问题，甚至让开发人员都毫无觉察。当然，也包括将各种从未想到过的操作平滑地转换为JavaScript。GWT无意挑起趋势和语言之

争，它只是一种实用的、能够解决问题的方案。

为此，当我看到David Geary的这本书如其名的著作时，感到非常高兴，因为本书侧重于具体而实用的解决方案。本书与GWT的精神完全合拍。如果抛开用户关心的问题不谈，那么无论设计模式还是优雅的代码都将无的放矢。David在本书中不仅提供了精心准备的各种示例，同时也向读者展示了更高级的内容。而且，书中每个示例都能完美地适应两个层面的应用。首先，通过这些示例可以学习到使用GWT构建用户界面的过程，其中不仅融合了常见的模式，甚至还揭示出许多可能的缺陷。其次，书中每个解决方案都可以在实际项目中照样子加以重用。因此，如果将来涌现的GWT应用程序中出现这些示例的影子，我一点也不会感到惊讶。我相信读者也会认同这种高效率的方法。事实上，读者对书中解决方案的评判标准，最好能像我对GWT给出的评判标准一样：看手中的这本书对您创建满足用户需求的应用程序是否有用。我想对于本书答案应该是：有用。

Google公司Kelly Norton

前言

Java开发人员已经超过600万了！而且，这其中的大多数人都有使用Swing等桌面应用程序框架的经验。无论在NFJS (No Fluff Just Stuff, nofluffjuststuff.com) tour[⊖]、Java Users Groups，还是在其他大会上例行介绍GWT (Google Web Toolkit) 时，我要做的第一件事，就是询问有多少与会者使用过AWT (Abstract Window Toolkit)、Swing以及SWT。得到的反馈显示，大约有95%的人用过以上框架。

AWT、Swing和SWT与Google Web Toolkit有什么关系？从许多方面来看，GWT就是用于开发Web应用程序的Swing，但无需借助启用技术（如Web Start）。通过GWT，Java开发人员可以利用熟悉的Swing、AWT或SWT开发经验，开发出能够在浏览器中运行的应用程序。因此，我在会上所做的第二件事，就是告诉与会者“如果你举手了，那么就说明你会用GWT”。当然，这些开发人员还必须再学习一种新框架及其API（为此，他们无疑要有一本好书）；不过，从能力构成上讲，他们已经懂得如何实现在浏览器中运行的Ajax驱动的应用程序了。如果曾经使用过AWT、Swing或SWT，那么可以肯定地说，您一定有通过在匿名内部类中实现事件处理程序接口，编写过GWT的事件处理程序的经历——这句话的意思，相信您能理解。而且，如果我告诉您GWT提供了带有那些接口空实现的适配器类，而您可以根据需要有选择地覆盖该类中的方法，那么您一定会有一种到了家的感觉，这种感觉建立在费尽心机地使用JavaScript、HTML、CSS和XMLHttpRequest对象，经过长途跋涉，最终创造出简单Ajax功能的艰辛之上。可以说，能够让那么多开发者立即上手使用，就是GWT最大的亮点。

在Rob和我搞懂了GWT是什么，以及它能做什么之后，我们就对其潜力充满信心。事实上，我们正因为如此有信心才编写了本书。过去10年，我曾编写过7本有关Java的技术书，虽然写多少本书好像也不算什么，但要再写另外一本书仍然需要付出大量的劳动。因此，如果这本书的主题不是具有成为“下一件大事”的潜力，我也不会觉得有必要为它写一本书。相信我把希望寄托在GWT之上，并且花上整整6个月时间把本书呈现给读者应该是值得的。

但是，GWT还不只是能够构建Ajax驱动的Web应用程序。确切地说，它能构建运行在浏览器中的桌面般的应用程序。在本书中，Rob和我向您展示了很多应用程序的实现过程及技巧，比如说在浏览器中打开多个窗口，其中每个窗口都包含基于提供给Yahoo! Maps Web服务的地址返回的地图；用户可以在浏览器中拖动每个窗口、调整窗口大小，并且能在各自窗口中拖动地图。用户还可以操作悬浮在窗口中的GWT部件，实现对地图的放大和缩小。此外，通过快速

⊖ 全球四大Java研讨会之一。——译者注

地拖动一幅地图，用户还能够启用动态滚动，其效果就像Apple iPhone中滚动联系人列表一样。具体一点说，就是只要拖动地图时间少于半秒钟，应用程序就会开启地图的动态滚动功能，一方面沿着拖动方向滚动，另一方面则基于拖过的像素距离调整滚动的速度。上述功能在其他Web应用程序框架中（例如Struts，没错，甚至在Ruby on Rails中）如果不是编写大量JavaScript代码并整合到相应框架中，几乎都是不可能实现的。

也就是说，GWT与其他Web应用程序框架的区别，就是提供了对实现桌面般的、能够在浏览器中运行的应用程序的支持。但是，如果把GWT想像成为一个简单地捆绑了Ajax的Web应用程序框架，那可就错了。事实上，GWT和Flash、Flex一样，为开发人员实现桌面应用程序中各式各样的丰富特性提供了强大支持。

最后，本书不是一本GWT的入门书。如果不熟悉GWT，第1章的解决方案中简单介绍了一些GWT的基础知识。但是，从第2章开始，我们就不再谈基础了，而是立即深入到诱人的实例剖析当中。因此，我们希望您通过网络或其他GWT入门图书掌握了必要的基础知识。通过本书，我们想展示各种前沿应用，以及使用GWT的技巧和方法。好吧，现在就翻到第1章，让我们开始神奇之旅！

致谢

一本书的写作绝不是单打独斗，本书当然也不例外。在本书从构思到上架期间，许多人都伸出过热情的援手。

首先，我们要感谢Google公司的Kelly Norton。Kelly是参与GWT开发的Google工程师，他对本书的贡献不可抹杀。Kelly给我们提出了极好的评审意见和建议，这些意见和建议也只有GWT工程师才能提出来。根据他提供的可行性意见和建议，我们修改了示例中的原始代码，最终体现出最佳实践并利用了GWT的特长。事实上，本书中的许多提示都要直接归功于Kelly的意见。

其次，我们也要感谢其他审稿人，他们给出的宝贵反馈涉及内容主旨，让我们更全面地认识到代码中存在的问题，并提升了本书的代码质量。这些审稿人是Henry Crutcher、Ido Green、Thad Humphries、Jeff Kurtz、Shailesh Mangal、Ted Neward、Sang Shin和Dick Wall。此外，Robert Kuhar和WingDings Book Study Group的朋友们也给我们提供了很好的建议。

再次，我们要对Anthony Francavilla和Dan Moore致以深切谢意。Anthony是Colorado Homefinder的CTO，他维护着一个实时房产网站（www.cohomefinder.com）。该网站是一个Struts应用程序，Dan在使用Struts实现它时也在其中嵌入了GWT部件。Anthony和Dan非常友好，他们把按揭计算器的代码交给我们，而本书最后一章整合GWT与遗留代码的解决方案中，就使用了他们提供的代码。

同样，Mary Lou Nohr仍然是我们的技术编辑，而且和以前一样，她不仅润色了我们的书稿，也提醒我们下次写作时该如何行文运笔。谢谢你，Mary Lou。

最后，我们要感谢参与本书编辑制作流程中的Prentice Hall的朋友们。感谢版权编辑(Signing Editor) Greg Doench出色的整合工作；感谢策划编辑Christopher J. Zahn博士对本书规划提供的帮助；感谢高级项目编辑Kristy Hart对本书制作付出的努力。

作者简介

David Geary先后编著了8本Java技术书，包括最畅销的*Graphic Java 2 Series*、*Advanced JavaServer Pages*和*Core JavaServer Faces*（全部由Prentice-Hall出版）。David曾是JSTL（JavaServer Pages Standard Template Library，JSP标准模板库）和JSF（JavaServer Faces）的Sun专家组成员。他还是Apache Struts项目执行委员和Struts Template Library（基于JSP片段构建网页的流行开源框架Tiles的前身）的发明者。此外，David还为Sun的Web Developer Certification Exam编写过试题。目前，David担任Clarity Training Inc.公司董事长职务，该公司是一家专注于服务器端Java技术咨询的机构。

Rob Gordon是一位独立咨询师，擅长企业系统的设计和实现。他曾是Sun的开发人员，也是*Essential JNI*的作者和*Essential JMF*的合著者。

目 录

译者序
序
前言

第1章 GWT入门与进阶	1
1.1 综述	1
1.2 本章要点	3
1.3 GWT部件简介	4
1.4 GWT应用程序分析	6
1.5 通过GWT面板构建用户界面	7
1.5.1 根面板	9
1.5.2 user.client.ui API	10
1.6 通过CSS为部件添加样式	10
1.7 GWT与事件驱动的程序设计	13
1.8 国际化与本地化	16
1.8.1 实现GWT国际化的4个步骤	17
1.8.2 使用GWT的i18nCreator命令	19
1.9 GWT Web应用程序中的新“视图”	21
1.10 GWT的历史机制	23
1.10.1 添加历史类和历史侦听器	24
1.10.2 处理浏览器后退按钮	26
1.11 GWT的对等实体：DOM元素	28
1.12 使用延迟命令赋予部件键盘焦点	28
1.13 实现远程过程调用	30
1.14 使用自定义部件	36
1.15 在GWT模块中封装自定义部件	38
1.16 完整的登录应用程序	42
1.17 本章回顾	46
第2章 JavaScript整合技术	48
2.1 本章要点	48
2.2 整合JavaScript库Script.aculo.us	48
2.2.1 在HTML文件中导入JavaScript 文件	55

2.2.2 在GWT配置文件中导入JavaScript 文件	56
2.3 整合JavaScript日历	58
2.4 本章回顾	62
第3章 自定义部件	63
3.1 本章要点	63
3.2 复合的自定义部件	64
3.2.1 复合部件和传递方法	78
3.2.2 事件处理程序	79
3.2.3 触发变化事件	80
3.2.4 实现HasText和HasName接口	81
3.2.5 整合Script.aculo.us的Blind Down 效果	81
3.3 低级自定义部件	84
3.3.1 在低级自定义部件中使用DOM类	88
3.3.2 为自定义部件设置默认CSS样式	88
3.3.3 在自定义部件中通过参数控制Script. aculo.us效果	88
3.4 本章回顾	89
第4章 视口与地图	91
4.1 本章要点	91
4.2 通用视口部件	93
4.3 地图应用程序的文件和目录	94
4.4 实现视口	94
4.4.1 通过AbsolutePanel实例在像素级别 上放置部件	97
4.4.2 在视口中拖动视图	98
4.4.3 通过焦点面板响应鼠标事件	98
4.4.4 通过事件预览禁止浏览器对鼠标 事件的默认响应	98
4.4.5 捕获事件	99
4.5 深入理解焦点面板在视口中的应用	102

4.6 带动态手势的视口视图	103	7.3.3 使用GWT的伸缩表格并为其单元格 添加样式	188
4.7 完成后的视口	106	7.3.4 接收事件	189
4.7.1 加入拖动手势	114	7.3.5 操作部件DOM元素的Z轴堆叠 次序	191
4.7.2 响应拖动手势的动态滚动	114	7.3.6 调整弹出面板的大小	192
4.7.3 与拖动手势保持同方向动态滚动	115	7.3.7 防止意外的浏览器效果	195
4.7.4 改变动态滚动的速度	115	7.4 窗口标题栏	198
4.8 本章回顾	116	7.4.1 在水平面板的单元格内对齐部件	201
第5章 访问在线Web服务	117	7.4.2 通过鼠标侦听器和图像模拟3D 按钮	201
5.1 本章要点	117	7.4.3 移动弹出面板	202
5.2 Yahoo! Maps Web服务	118	7.4.4 在窗口标题栏中防止意外的浏览器 效果	205
5.2.1 示例应用程序分析	118	7.5 本章回顾	207
5.2.2 地图应用程序的文件和目录	122	第8章 伸缩表格	208
5.2.3 实现YahooMaps应用程序	122	8.1 本章要点	208
5.3 访问Yahoo!的Web服务	129	8.2 伸缩表格简介	209
5.4 缩放	134	8.3 简化伸缩表格的操作	216
5.5 本章回顾	145	8.4 删除伸缩表格的行	220
第6章 拖放技术	146	8.5 编辑伸缩表格的单元格	223
6.1 本章要点	146	8.6 动态调整伸缩表格的列	226
6.2 拖放示例应用程序	146	8.6.1 通过GWT构建事件侦听器和触发 事件	228
6.3 拖放模块	148	8.6.2 构建ResizableCellPanel部件	232
6.4 实现拖放应用程序	150	8.6.3 将ResizableCellPanel部件加入伸缩 表格	237
6.4.1 使用音乐播放器和购物车面板 组件	152	8.7 通过伸缩表格实现数据分页	240
6.4.2 使用拖动源和放置目标	154	8.8 使用Pager部件	246
6.4.3 定义CSS类	158	8.9 本章回顾	249
6.5 在GWT模块中实现拖放	159	第9章 文件上传	250
6.5.1 模块配置文件	160	9.1 本章要点	250
6.5.2 抽象拖动源和放置目标类	160	9.2 GWT的文件上传部件	250
6.5.3 鼠标侦听器	165	9.2.1 创建用户界面	257
6.6 本章回顾	173	9.2.2 解析JSON	258
第7章 简单的窗口	174	9.3 Apache Commons fileUpload	262
7.1 本章要点	175	9.4 本章回顾	267
7.2 基于伸缩表格构建窗口	178		
7.3 简单的窗口	179		
7.3.1 使用弹出面板	184		
7.3.2 重写弹出面板默认的事件处理 行为	186		

第10章 整合Hibernate	268	第12章 GWT与遗留程序	305
10.1 本章要点	268	12.1 本章要点	305
10.2 Hibernate示例应用程序	268	12.2 Struts/GWT混合应用程序实例	305
10.3 服务器端Hibernate	280	12.3 简单的Struts应用程序	307
10.3.1 Hibernate配置文件	280	12.3.1 Struts登录应用程序代码	309
10.3.2 POJO Java类	281	12.3.2 JSP页面	309
10.3.3 联系人映射文件	283	12.3.3 支撑Bean、表单Bean和动作Bean	312
10.3.4 RPC Servlet实现	283	12.4 GWT/Struts混合应用程序	314
10.3.5 Eclipse配置	287	12.4.1 混合应用程序的文件和目录	316
10.4 本章回顾	288	12.4.2 修改Ant构建文件并加入Struts	317
第11章 外部服务器部署	289	12.4.3 混合应用程序的代码	318
11.1 本章要点	289	12.5 JSON在配置GWT部件中的应用	322
11.2 外部服务器目录结构	289	12.6 遗留代码和GWT部件的协同	327
11.3 使用Ant实现自动构建	293	12.7 通过GWT视图替换HTML及JSP 视图	328
11.4 开发与部署	301	12.8 深度框架整合	329
11.5 部署后的应用程序调试	302	12.9 本章回顾	330
11.6 本章回顾	304		

第1章 GWT入门与进阶

近几年来，基于Java的Web应用程序框架有了充分发展，颇受推崇的Apache Struts就是其中第一代框架的代表。

随后，第二代框架Tapestry和JavaServer Faces又粉墨登场。这些框架带来了可喜的进步，尤其是能够直接将输入字段连接到JavaBean组件属性^①。不过，第二代框架的定义主要以支持组件为核心。通过使用组件，开发人员能够创建自己的组件，并且将这些组件用于网页中就能看到立竿见影的效果。

如今，以Google Web Toolkit (GWT) 为标志的第三代框架又出现了。同第二代框架类似，GWT也提供组件，而且，这些组件在GWT中被称为部件 (Widget)。但是，与第二代框架有所不同，GWT代表的不是改进，而是一场真正的革命，因为它完全打破了Web应用程序的既有模式。

与此前的框架不同，GWT并不是建立于经典的、基于表单的应用程序之上的。当然，如果读者愿意，通过GWT创建这种过气的应用程序也没有任何问题。确切地讲，GWT是第一个主流的、基于Java的Web应用程序框架，通过它能够在浏览器中构建类似桌面应用般的程序。换句话说，有了GWT，那些由陈旧网页构成的Web应用程序将淡出我们的视野，而进入我们视野的则是能够在一个浏览器窗口中运行，并且与我们使用AWT、Swing或SWT开发的桌面程序类似的应用程序。

1.1 综述

本书将展示如何面向常见浏览器开发一些功能，这些功能在过去是不可能实现的。如图1-1所示，其中展示了本书将要介绍的一些代表性功能，例如：浏览器中可移动可调整大小的窗口、支持根据用户手势自动滚屏的滚动视口 (viewport) ——同Apple的iPhone中的自动滚屏类似、可以让用户在弹出式面板 (panel) 中编辑条目的Rolodex (通信录或名片簿) 以及拖放的界面等。

但是，我们编写本书的目的，并非仅仅为了向读者展示一些既酷又实用的例子，以及如何实现这些例子。比如说，通过实现拖放的例子，我们还会揭示出许多有用的GWT技术。像捕获事件、抑制浏览器对鼠标事件的响应，以及临时改变某个部件的光标等，这些技术不仅在拖放

^① 有关Struts的更多信息，请参考<http://struts.apache.org>。有关JavaServer Faces的更多信息，请参考<http://java.sun.com/jaee/jaserverfaces>。

功能中可以派上用场，在其他情形下同样也非常有用。

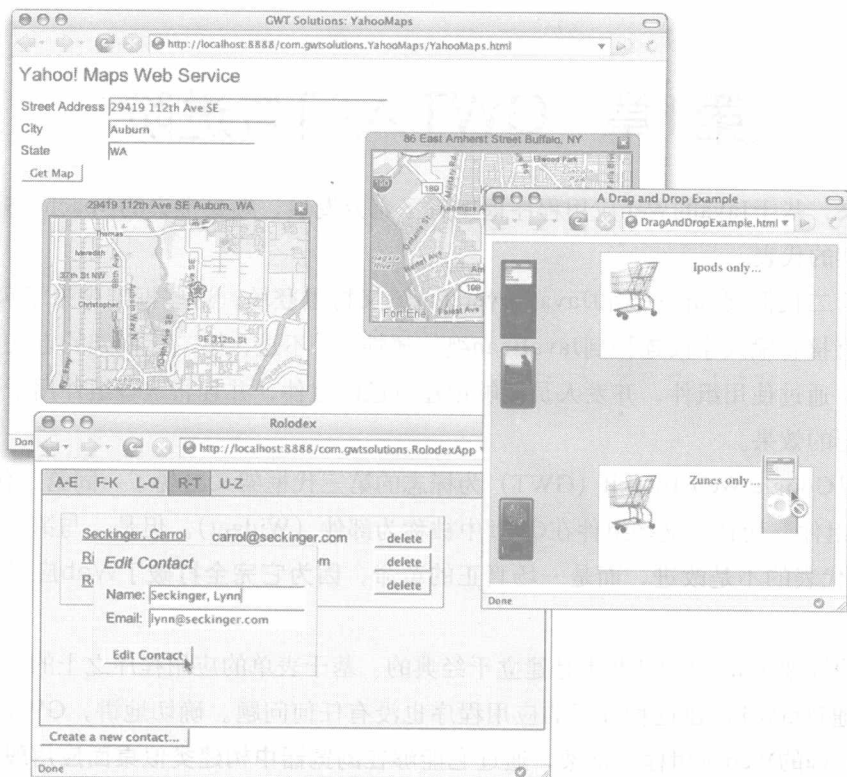


图1-1 一些既酷又实用的例子

本书通过构造一系列既酷又实用的功能向读者展示诸多实用技术，同时也致力于使本书成为一本无可争辩的高级GWT技术书。

读者在本书中将看不到如何安装GWT的介绍，我们也不会谈论诸如水平和垂直面板之间差异的基础知识，更不会让Hello World示例程序浪费读者的时间。假如读者想了解上述内容，请您参考其他GWT书籍，或者到Web上查找相关资料。在理解了以上内容后，您可以再回来跟我们一同驶入快车道。

话虽这么讲，但我们也知道并非所有人都熟知GWT的基础。有鉴于此，本章的解决方案将通过展示一个完善的登录应用程序，来使读者掌握GWT中的基本概念。作为本书第一个解决方案，我们开宗明义，先列出这个解决方案中将要涉及的内容要点，以后各章的体例也同样如此。

提示：GWT比Ruby on Rails更好吗？

2005年，以Ruby开发的Web应用程序框架Ruby on Rails精彩亮相。Rails不仅是一个极富生成效率的框架，而且还内置了对Ajax的支持。此外，这个框架还能够自动连接视图与后台数据库，因而提供了一种端到端的解决方案。然而，从根本上来讲，Rails（从许多方

面都可以说)是一种经典的、基于表单的Web应用程序框架。该框架通过rhtml文件来模拟JavaServer Page (JSP) 创建网页的方式。

与Ruby on Rails相对, GWT (除了它对RPC的支持) 则可以说是一种纯粹的客户端框架。不同于Rails, GWT不是一种经典的Web应用程序框架, 它的出现是为了让开发者在浏览器中实现类似桌面应用般的程序。

综上所述, 对这个问题的回答如下: 如果要构建经典的、有数据库支持的、基于表单的应用程序, 其他情况类似, 那么选择Rails要胜过GWT——原因就是Rails具有对数据库的整合。但是, 如果想超越经典的Web应用程序, 在浏览器中实现桌面应用程序般的功能, 那么GWT则是首选。

1.2 本章要点

本章的解决方案将探索GWT的如下方面:

- GWT部件简介
- GWT应用程序分析
- 通过GWT面板构建用户界面
- GWT的根面板部件
- 使用CSS为GWT部件添加样式
- 事件驱动框架中的程序设计
- 对GWT应用程序进行国际化和本地化
- 使用GWT的`i18nCreator`命令
- 在GWT应用程序中加载新视图
- 使用GWT的历史机制
- 整合浏览器的后退和前进按钮
- GWT中的对等实体 (peer-based) 系统
- 使用GWT的延迟命令
- 实现远程过程调用
- 使用自定义部件
- 在GWT模块中封装自定义部件

如果读者此前使用过GWT, 可能会熟悉前面列出的一些内容, 例如使用GWT面板构建用户界面。不过, 读者很可能不知道, GWT与AWT类似, 也是一个基于对等实体的系统。在本解决方案中, 我们也会讨论与对等实体系统设计相关的, 以及其他一些非基础性的内容——例如, 整合浏览器的后退按钮和在可重用的GWT模块中封装自定义部件。

在本书提供的所有解决方案中, 读者可以随意挑选自己感兴趣的部分仔细研究; 或者, 也

可以从头到尾通读每个解决方案。无论怎样，只要您坚持看完本书，一定会成为更加优秀的GWT开发者。

1.3 GWT部件简介

同AWT（Abstract Window Toolkit，抽象窗口工具箱）及Swing类似，GWT也构建于部件之上。换句话说，要创建用户界面，必须实例化部件，然后将它们添加到面板中。最后，再将面板添加到应用程序的根面板。所谓根面板，就是在特定视图中包含所有部件的顶级容器面板。GWT中包含的许多部件都具有继承性层次关系。图1-2展示了其中一些部件。

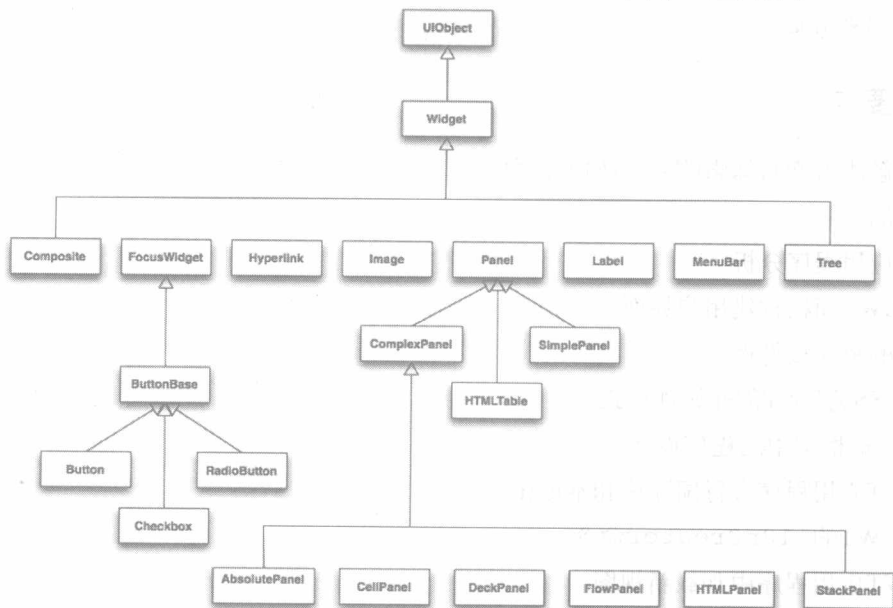


图1-2 部分GWT部件层次图

当然，GWT所提供的部件远不止图1-2中所示的这些。但是，其中有一部分是最常用的，参见表1-1。

表1-1 常用的GWT部件（所有类都来自com.google.gwt.user.client.ui）

部件类	实现的接口	说明
AbsolutePanel	EventListener, HasWidgets	用于包含其他部件，对其部件的定位可以精确到像素级
Button	EventListener, HasFocus, HasHTML, HasText, SourcesClickEvents, SourcesFocusEvents, SourcesKeyboardEvents	用户可以单击的按钮
CheckBox	EventListener, HasFocus, HasHTML, HasName, SourcesClickEvents, SourcesFocusEvents, SourcesKeyboardEvents	用户可以选中的复选框。可同时选中多个复选框