

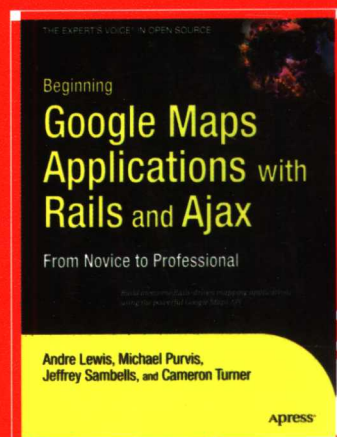
Google Maps 应用程序

Rails和Ajax开发指南

Beginning Google Maps Applications with
Rails and Ajax: From Novice to Professional

(美) Andre Lewis Michael Purvis Jeffrey Sambells Cameron Turner 著
龚波 吴耀宇 等译

使用强大的Google Maps API
构建惊人的Rails驱动地图应用程序



机械工业出版社
China Machine Press

TP393.09/179

2008

Web开发系列丛书

Google Maps 应用程序 Rails和Ajax开发指南

Beginning Google Maps Applications with
Rails and Ajax: From Novice to Professional

(美) Andre Lewis Michael Purvis Jeffrey Sambells Cameron Turner 著
龚波 吴耀宇 等译



机械工业出版社
China Machine Press

本书以实例方式介绍如何使用 Rails 和 Ajax 开发 Google Maps 应用程序。本书内容包括 Google Maps 和 Rails、与用户和服务端交互、地理编码地址、使用第三方数据、改进用户界面、优化和调整大型数据集、Google Maps API 预计改进之处、高级提示和技巧、线段、长度和区域以及高级的地理编码话题等。附录还列出了 Google Maps API，以方便在阅读正文内容时查询。本书不仅能够满足 Google Maps API 初学者的入门需求，也能给高级开发者提供很多重要的建议。

本书适合 Google Maps 应用程序开发人员阅读，也可供其他软件开发人员参考。

Andre Lewis, Michael Purvis, Jeffrey Sambells, and Cameron Turner: *Beginning Google Maps Applications with Rails and Ajax: From Novice to Professional* (ISBN: 1-59059-787-7) .

Original English language edition published by Apress L. P., 2560 Ninth Street, Suite 219, Berkeley, CA 94710 USA. Copyright © 2007 by Apress L. P. Simplified Chinese-language edition copyright © 2008 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。

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：01-2007-4212

图书在版编目 (CIP) 数据

Google Maps 应用程序：Rails 和 Ajax 开发指南/(美) 安德·勒维斯 (Andre Lewis) 著；龚波等译. —北京：机械工业出版社，2008.4

书名原文：Beginning Google Maps Applications with Rails and Ajax: From Novice to Professional (Web 开发系列丛书)

ISBN 978-7-111-23695-5

I. G… II. ①安… ②龚… 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 049527 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：周茂辉

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2008 年 4 月第 1 版第 1 次印刷

186mm × 240mm · 17.25 印张

标准书号：ISBN 978-7-111-23695-5

定价：38.00 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

Google Maps 的诞生标志着互联网地图服务的兴起。Google Maps API 是 Google 推出的编程 API, 可以让全世界对 Google Maps 有兴趣的程序员自行开发基于 Google Maps 的服务, 建立自己的地图网站, 从而降低开发地理应用程序的门槛, 把 Google 的地图信息和自己的数据整合起来以建立自己的地理信息服务。

2006 年 4 月 3 日, Google 推出了 Google Maps API 2, 第 2 版能对第 1 版提供 99% 的兼容性。目前, Google Maps API 的开发工作在始终不断进行着。随着 API 功能的逐渐增强, 全世界也涌现出越来越多“快捷和实用”的地理应用程序。

本书以实例方式介绍如何在 Ruby on Rails 平台上使用 Google Maps API。为什么要采用 Rails 来使用 Google Maps API 呢? 因为 Rails 是简单的、强大的, 同时也是免费的。使用 Rails 框架可以大幅度提升基于数据库的 Web 应用程序的开发效率。把 Rails 和 Google Maps API 集成起来, 可以开发绝妙的 Web 应用程序。借助于 Rails, 也可以很容易生成和使用 XML。借助于 XML, Google Maps 可以更广泛地得以应用。Rails 也提供对 JSON(JavaScript Object Notation)的内置支持。

本书是国内引入的第一本有关 Google Maps API 开发的专业图书。本书作者 Andre Lewis、Michael Purvis、Jeffrey Sambells 和 Cameron Turner 具有丰富的阅历, 对新技术孜孜以求。作者们积极参与 Google Maps API 成长进程, 具有丰富的应用开发经验。

本书的可读性和实践性都很强, 并不是简单陈列 API 方法或者事件, 而是向读者详细介绍: 如何基于现有免费资源, 提供全面的代码范例, 演示如何逐步建立地理应用程序。

本书内容包括初识 Google Maps、跨越基础和高级地图功能和方法 3 部分, 还提供 Google Maps API 参考。本书不仅能够满足 Google Maps API 初学者的入门需求, 也给高级开发者提供了很多重要的建议。阅读本书之后, 读者会对 Google Maps API 有深入了解, 具备开发基于 Google Maps API 应用程序的能力。

原书的句子比较晦涩, 涉及的领域知识比较广泛。译者们竭力保证译文的正确、通顺和优雅, 但错误之处在所难免, 恳请广大读者多加指正。

本书主要由龚波、吴耀宇和徐雅丽翻译, 龚波负责最后的统稿。其他参与本书翻译和质量保证的人员还有张平、田丽韞、李志、任志宏、韩存兵、田振中、赵军锁、陈蓓、姜南。感谢大家的辛勤工作和专业精神。同时, 还要感谢出版社编辑们的辛勤工作!

译者

2008 年 3 月

关于作者



Andre Lewis

在开始创建一个简单的在线 Wi-Fi 咖啡厅时, Andre Lewis 就对 Google Maps 产生了浓厚的兴趣。这个项目最后演变为一个活跃的促进社团互动的站点 <http://hotspotr.com>。从那时开始, 他使用 Ruby on Rails 开发了大量的工具和技术, 应用于基于地图的应用程序。

Andre 喜欢多种多样的技术, 而面向地理的应用程序始终是他非常喜欢的主题。他曾经搭建支持每日百万次访问量的 Web 系统, 也喜欢使用 JavaScript 和 CSS 创建相应的 Web 页面。他现在是一位自由作家, 专注于 Web 2.0 技术和 Ruby on Rails 应用程序。他的博客是 <http://earthcode.com>。他还定期参加 Bay Area 技术组。

Andre 现在在美国 San Francisco (旧金山) 居住和工作。当不钻研客户端应用程序或者最新技术时, 他喜欢爬山、露营以及骑摩托车远行。



Michael Purvis

Michael Purvis 是一名学生, 目前就读于位于 Ontario (安大略省) 的 Waterloo (滑铁卢) 大学机械工程系。在开发 Web 脚本之前, 他忙于其他类型的项目, 诸如开发能够玩 Connect 4 游戏的 LEGO Mindstorms 工具包。在本书的 PHP 版本出版之后, 他被 Google 招聘, 在 New York City 办公室做了 4 个月的实习生。他在自己的同学之间始终维护着一个活跃的社区, 并逐步扩展为 PunBB 和 MediaWiki。

他曾经为 Position Is Everything 网站编写 CSS, 并且偶尔参与讨论有关 CSS 的邮件列表。他崇尚简约化, 但是对于那些涉及曲折和高深技巧的问题, 他也乐此不疲。在他的博客 <http://uwmike.com> 上, 偶尔会出现对这些疑难问题的讨论, 以及一些其他非技术话题。

在空闲时间里, 他喜欢烹饪、写作、骑车游玩以及跳舞。他曾经参与过 We-Create 公司的很多令人兴奋的 PHP 项目, 对独立的 Web 标准保持着浓厚的兴趣。



Jeffrey Sambells

Jeffrey Sambells 是一名图形设计者和自学成才的 Web 应用程序开发人员, 他具有把图形的可视化世界和代码的智力领域完美集成的能力, 这是非常了不起的。他是图形通信管理专业的学士, 并辅修了多媒体专业的课程。Jeffrey 最初受雇于一家传统的图书印刷企业, 但是, 他很快意识到自己真正的兴趣点是像素和代码。

在 1999 年, 他与别人一起创办了 We-Create 公司, 这是一家位于 Waterloo (Ontario) 的 Internet 软件公司, 自此开始了漫长的、富有挑战性和创造性的工作。现在, 作为 We-Create 公司研发部的负责人, Jeffrey 负责研究和跟踪最新的 Internet 技术, 然后使用遵循 Web 标准的方法来集成这些技术。在 2005 年的下半年, 他还成为了一名 Zend 认证工程师。

Jeffrey 有很多业余爱好, 包括摄影和干木工活。有时候, 他会驾驶一叶独木舟, 畅游在 Algonquin Provincial 公园的湖中; 或者, 与他的妻子一起拿着地图, 开着车, 到远方冒险。Jeffrey 也有自己的个人网站 <http://JeffreySambells.com>。在这个平台上, 他会与大家分享自己对 Web 技

术、摄影、设计等方面的想法、主意和观点。他现在与妻子 Stephanie、女儿 Addison 一起住在加拿大的 Ontario，他还喂养了一条小狗，名为 Milo。



Cameron Turner

自从在七岁拥有自己的第一台 VIC 20 时，Cameron Turner 就痴迷于计算机编程。从 1994 年开始，他就一直从事交互式网站的开发。1999 年，他与别人一起创办 We-Create 公司，专注于 Internet 软件开发。现在，他是该公司的首席技术官。Cameron 获得 Waterloo(滑铁卢)大学的计算机专业学士学位，擅长应用密码学、数据库设计以及计算机安全。

自从本书的 PHP 版本出版以来，Cameron 就开始给位于 Waterloo(Ontario)地区的公司和协会进行各种讲演和讲座，涉及内容广泛，包括 Google Maps、搜索引擎优化，以及涉及专业化 Web 软件开发的其它主题。

现在，Cam 与妻子 Tanya、儿子 Owen 和爱犬 Katie 一起，居住在加拿大的技术中心——Waterloo(滑铁卢)。他的业务爱好包括地理寻宝探险(geocaching)、骑自行车、滑水以及绘画。他的个人博客 <http://CamTurner.com>，专注于讨论各种非技术话题、思想、原理以及家庭生活。

关于技术审阅者



Sam Aaron

Sam Aaron 是英国 Newcastle 大学计算机科学系的博士生。他现在即将完成有关兴趣管理的论文。他同时是 Ruby 和 Rails 的痴迷者，积极使用和宣扬这些令人兴奋的技术。他自己创办和组织了当地的 Ruby and Rails User Group——ncl.rb，每月能吸引 20 多人参与讨论。现在他使用 Rails 为 Newcastle 大学的运输系开发一个基于 Web 的决策支持工具，并期望在其中尽可能加入更多的 Rails 技术。

现在，Sam 居住在英格兰 Newcastle-upon-Tyne 的码头附近。他喜欢站在窗前看着小鸟轻盈地飞越河流。在工作时间，他喜欢使用自己的 PowerBook 笔记本聆听怪异的电子乐。在休闲时间，他喜欢到郊外放松。他爱好露营、爬山以及力量锻炼。在家时，他经常弹钢琴或者玩桌球。他自己没有汽车，甚至还不会驾驶，但更喜欢骑着自行车到处游玩——尤其喜欢与自己的好朋友一起骑着自行车远行。

完成博士论文后，Sam 计划创办一家公司，涉足有关 Ruby 和 Rails 的 Web 开发、培训和咨询业务。现在，他正与 Newcastle 大学协商，期望能够在相关课程中加入有关 Ruby 和 Rails 的内容。如果想了解 Sam 的近况，还可以访问他的博客 <http://sam.aaron.name>。

目 录

译者序
关于作者

第一部分 初识 Google Maps

第 1 章 Google Maps 和 Rails	1	第 3 章 与用户和服务器交互	24
1.1 KML: 第一个地图	2	3.1 添加交互能力	24
1.2 Warfaring: 第二个地图程序	3	3.2 寻宝游戏	24
1.2.1 添加第一个停靠点	4	3.3 回顾应用程序结构	26
1.2.2 添加飞行路线	4	3.4 完善自己的应用程序	26
1.2.3 添加目的地停靠点	5	3.4.1 创建新的控制器	26
1.2.4 添加行车路线	6	3.4.2 创建标记模型和迁移	26
1.3 已经掌握 Rails 吗	6	3.4.3 创建数据库, 使用 Rails 连接数 据库, 并运行迁移	27
1.4 接下来的内容	7	3.4.4 创建地图视图	28
第 2 章 开始行动	8	3.5 创建地图和标记点	28
2.1 关于 JavaScript、帮助器和插件	8	3.6 监听用户事件	29
2.2 创建自己的 Rails 应用程序	9	3.7 使用信息窗口获取更多信息	31
2.3 第一个地图	9	3.7.1 在地图上创建信息窗口	31
2.3.1 密钥	9	3.7.2 在信息窗口中嵌入一个表单	32
2.3.2 讨论范例地图	11	3.7.3 避免模糊状态	35
2.3.3 指定一个新的位置	12	3.7.4 控制信息窗口的尺寸	37
2.3.4 分离代码和内容	13	3.8 实现 Ajax	38
2.3.5 内存清理工作	15	3.8.1 Google 的 GXmlHttp 与 Prototype 的 Ajax.Request	38
2.4 基本的用户交互	16	3.8.2 使用 Google 的 Ajax 对象	39
2.4.1 使用 Map 控件	16	3.8.3 使用 GXmlHttp 实现数据保存	39
2.4.2 创建标记	16	3.8.4 解析 JSON 结构	43
2.4.3 检测标记单击操作	18	3.9 从服务器检索标记	44
2.4.4 打开信息窗口	18	3.10 添加一些新风格	46
2.5 标记点的列表	19	3.11 Ajax 和 Prototype	48
2.5.1 使用数组和对象	19	3.12 小结	50
2.5.2 迭代	21	第 4 章 地理解码地址	51
2.6 小结	23	4.1 准备地址数据	51
		4.1.1 创建模型	51
		4.1.2 添加 full_address 方法	52
		4.1.3 填充数据库表	52

4.2 使用地理解码 Web 服务	54
4.2.1 使用地理解码服务的需求	54
4.2.2 Google Maps API 地理解码器	55
4.2.3 Google JavaScript 地理解码器	61
4.2.4 Yahoo Geocoding API	62
4.2.5 Geocoder.us	66
4.2.6 Geocoder.ca	68
4.2.7 Google 覆盖范围之外的其他 地理解码服务	69
4.3 持久保存查找结果	70
4.4 创建一个商店位置地图	71
4.5 小结	73

第二部分 跨越基础

第 5 章 使用第三方数据	75
5.1 使用可下载的文本文件	75
5.1.1 下载这个数据库	76
5.1.2 操作文件	78
5.1.3 关联和导入数据	79
5.1.4 使用新的数据库模式	82
5.2 界面抓取	89
5.2.1 我们的脚本工具: scrAPI	89
5.2.2 界面抓取需要考虑的问题	93
5.3 小结	94
第 6 章 改进用户界面	95
6.1 CSS: 改进自己的风格	95
6.1.1 最大化地图	97
6.1.2 添加悬浮的工具条	98
6.1.3 创建可收缩的侧面面板	101
6.2 脚本实现的样式	103
6.2.1 改变文档体的类属性	103
6.2.2 使用 JavaScript 代码来调整尺寸	104
6.2.3 填充侧面面板	107
6.2.4 获取侧面面板的反馈	108
6.3 过滤数据点	109
6.4 RJS 和可拖放的工具条	111
6.4.1 RJS 模板	111
6.4.2 可拖放的工具条	111

6.5 小结	112
第 7 章 优化和调整大型数据集	113
7.1 限制	113
7.2 服务器和客户端通信的串行化	114
7.3 优化服务器端处理过程	115
7.3.1 服务器端边界方法	116
7.3.2 服务器端公用点方法	119
7.3.3 服务器端集群	123
7.3.4 定制的细节覆盖层方法	127
7.3.5 定制图格方法	134
7.4 优化客户端的用户体验	141
7.4.1 客户端边界方法	141
7.4.2 客户端最靠近公用点方法	143
7.4.3 客户端集群	145
7.4.4 更多的客户端优化方法	149
7.5 小结	150

第 8 章 Google Maps API 预计

改进之处	151
8.1 运动方向	151
8.2 集成的 Google 服务	152
8.3 KML 数据	153
8.4 更多的数据层	154
8.5 满足企业级需求	155
8.6 界面改进	155
8.7 小结	156

第三部分 高级地图功能和方法

第 9 章 高级提示和技巧	157
9.1 调试地图	157
9.2 使用 API 与地图进行交互	158
9.2.1 帮助你找到自己的位置	158
9.2.2 使用 GEvent 强迫触发事件	159
9.2.3 创建自己的事件	160
9.3 使用 GOverlay 创建地图对象	161
9.3.1 选择覆盖层的窗格	161
9.3.2 创建一个快速的工具提示 覆盖层	162

9.4 创建定制的控制	165	10.2.1 大圆圈	200
9.4.1 创建控件对象	166	10.2.2 大圆圈长度	200
9.4.2 创建容器	167	10.2.3 球体表面的面积	202
9.4.3 容器定位	167	10.3 使用折线	205
9.4.4 使用这个控件	167	10.3.1 创建折线演示程序	205
9.5 向信息窗口添加标签	168	10.3.2 扩展折线演示程序	210
9.5.1 创建有标签的信息窗口	168	10.4 关于 UTM 坐标	211
9.5.2 收集信息窗口信息和修改标签	169	10.5 与日期变更线冲突	212
9.6 创建定制的信息窗口	170	10.6 小结	213
9.6.1 创建覆盖层对象和容器	174	第 11 章 高级地理解码话题	214
9.6.2 绘制 LittleInfoWindow	175	11.1 数据来自于什么地方	214
9.7 实现自己的地图类型、图格和投影	178	11.1.1 来自于政府数据源的数据	214
9.7.1 GMapType: 粘合在一起	179	11.1.2 原始的 GIS 数据源	218
9.7.2 GProjection: 定位事物 所处位置	179	11.2 基于邮政编码的地理解码服务	218
9.7.3 GTileLayer: 查看图像	184	11.3 使用 TIGER/Line 数据	222
9.7.4 Blue Marble 地图: 把所有事物 集成起来	186	11.3.1 理解和定义数据	222
9.8 小结	194	11.3.2 解析和导入数据	225
第 10 章 线段、长度和区域	196	11.3.3 创建地理解码服务	231
10.1 从平面开始	196	11.4 小结	235
10.1.1 长度和角度	196		
10.1.2 面积	197		
10.2 考虑球体环境	200		

附 录

附录 A 搜索需要的数据	237
附录 B Google Maps API	244

第一部分 初识 Google Maps

第 1 章 Google Maps 和 Rails

对于 Web 开发者而言, 去年也许是一段令人难以置信的兴奋时光。新工具层出不穷, 使得 Web 开发更加容易, 效率更高, 开发过程更加有趣。新的 API 使得开发者可以以更有趣的方式来集成数据和服务。作为开发者, 相对于以前, 我们能接触到更多新颖有趣的技术。

本书专门介绍一种 API, 它已经给我们带来特别深远的影响, 那就是 Google Maps。一旦本书在手, 你就可能已经深深认识到 Google Maps 的重要性。但是, 我们还建议你经常访问 Google Maps Mania (<http://googlemapsmania.blogspot.com>), 以了解 Google Maps 给开发社区所带来的创新文化。现在已经出现了很多基于 Google Maps API 的 Web 应用程序。如果没有 Google Maps API, 要实现这些应用程序几乎是不可能的。

在本书中, 我们将在一个同样炙手可热的平台上使用 Google Maps API, 那就是 Ruby on Rails。Rails 框架有助于大幅度提升基于数据库的 Web 应用程序的开发效率。Rails 是简单的、强大的, 同时也是免费的。把 Rails 和 Google Maps API 集成起来, 开发者可以开发绝妙的 Web 应用程序, 要知道, 在两年前, 实现这一点是很难的, 甚至是不可能的。

学习了本书接下来几章之后, 你就会从实现仅涉及标识符和 geocode 的简单任务, 到进一步掌握更高深的主题, 诸如如何获取数据, 如何显示很多数据点, 以及如何设计一个有用的吸引人的交互式用户界面。

有很多理由可以说明为什么 Ruby on Rails 是使用 Google Maps 的理想平台。借助于 Rails, 可以很容易生成和使用 XML, 而借助于 XML, Google Maps 也可以更广泛地得以应用。Rails 也提供对 JSON(JavaScript Object Notation)的内置支持, 其中 JSON 是从服务器向浏览器传送结构化数据的简明格式。最后, Ruby 还支持界面部件的开发库, 我们在稍后章节会进一步说明。

我们假设, 在阅读本书之前, 你已经具备一定的 Rails 基础。也许, 你的开发环境已经安装 Ruby on Rails, 并且知道如何创建和运行应用程序。如果没有的话, 也不要担心: 本章末尾的提示栏“你刚学习 Ruby on Rails 吗?”列出一些有价值的资源, 有助于你熟悉 Rails。现在, 无论你的 Rails 水平如何, 阅读本书都不会阻碍你对内容的理解, 也能够让你轻松掌握地图设计, 以及实现杀手级地图应用程序所需的任何知识。借助于 Rails 框架、Ruby 语言以及 Google Maps API, 你就拥有了一个强大的开发工具包。

我们知道你期望早点接触地图项目, 但是在深入研究相关代码之前, 还是应该首先介绍创建 ultraquickie 地图的两种简单方式: 使用 KML(Keyhole Markup Language, Keyhole 标记语言)文件以及通过 Wayfaring 地图站点。

这些方法都是按部就班的, 我们使用它们作为进入 Google Maps 世界的敲门砖。在本书第 2

章，你将有机会深入分析代码，当然就能创建出更加灵活和复杂的应用程序。

1.1 KML：第一个地图

KML 是实现在 Google 地图上显示自己的标记和内容的最简单方法之一。正如你所期望的那样，简单化和灵活性是背道而驰的，但是 KML 仍旧不失为学习上手的好方法。2006 年 6 月，Google 宣称它的官方地图站点将支持 KML 文件。把一个 URL 输入到搜索框，Google Maps 将显示 KML 文件中该 URL 所指定的位置。我们不会深入讨论这个 KML，而是仅提供一个简单的范例来展示 KML 方法的强大功能。尽管这个范例比较简单，但也足以能说明问题。

注意 名字 Keyhole Markup Language 是应用程序 Keyhole 的 XML 结构和 Google Earth 继承层次的一个节点。在 2004 年底，Google 就开始使用 Keyhole。

我们已经创建一个名为 toronto.kml 的文件，代码如清单 1-1 所示。段落标识是从 Wikipedia 中借用的，通过从 Google Maps 手工找到相应的位置就可以发现相应的坐标。

代码清单 1-1 一个 KML 文件示例

```
<? xml version="1.0" encoding="UTF-8"? >
<kml xmlns="http://www.google.com/earth/kml/2">
<Document>
  <name>toronto.kml</name>
  <Placemark>
    <name>CN Tower</name>
    <description> The CN Tower (Canada's National Tower, Canadian National Tower),
    at 553.33 metres (1, 815 ft., 5 inches) is the tallest ➡
    freestanding structure on land.
    It is located in the city of Toronto, Ontario, Canada, and is considered the
    signature icon of the city. The CN Tower attracts close to two million visitors annual-
    ly.

    http://en.wikipedia.org/wiki/CN_Tower</description>
    <Point>
      <coordinates> -79.386864, 43.642426</coordinates>
    </Point>
  </Placemark>
</Document>
</kml>
```

在实际的文件(<http://book.earthcode.com/kml/toronto.kml>)中，我们使用更多 Placemark 元素，以指向多伦多市其他一些有名的建筑物。为了在 Google Maps 上查看某个建筑物，可以把先前的 URL 粘贴到 Google Maps 的搜索框中。同样，也可以访问如下网址来实现：<http://maps.google.com/maps?f=q&hl=en&q=http://book.earthcode.com/kml/toronto.kml>。

图 1-1 是一个界面截图。

现在，上面就是所需的处理结果吗？确实，如果你的目的仅仅是显示相匹配的位置点，上面的显示结果就能满足你的需求。如果希望把所在地和喜欢的钓鱼场连接起来，你需要创建一个新的 KML 文件。

但是上述实现方式并没有多大意思，不是吗？尽管 KML 地图程序的功能非常强大，但是并

没有提供用户交互功能。实际上，第2章中大部分范例仅仅再现了 Google 此处提供的功能。但是一旦阅读到第3章，当全面掌握 Google Maps API 的威力之后，你就会知道可以实现的功能决不仅限于此。

在继续阅读之前，我们先探讨快速实现地图在线的另一种方式。

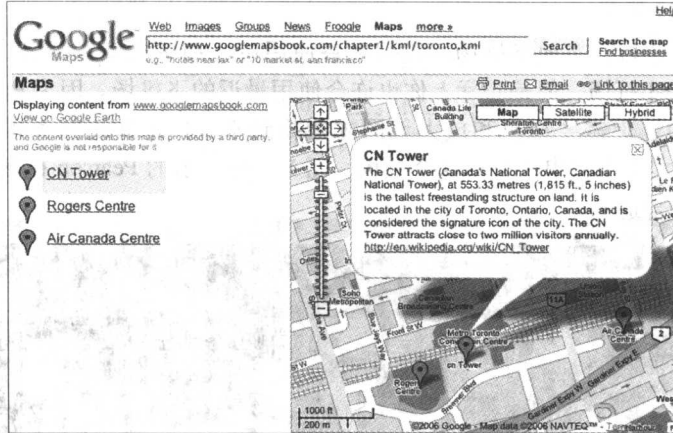


图 1-1 maps.google.com 网站显示的定制的 KML 数据文件

1.2 Warfaring: 第二个地图程序

现在有很多服务可以实现把手工设计的数据快速发布为免费地图。图 1-2 演示的 Wayfaring 就是这类服务的一种。由于具有优质设计和社区特征（比如评论和位置共享），Wayfaring 目前很受重视和推崇。Wayfaring 也是使用 Ruby on Rails 创建的。

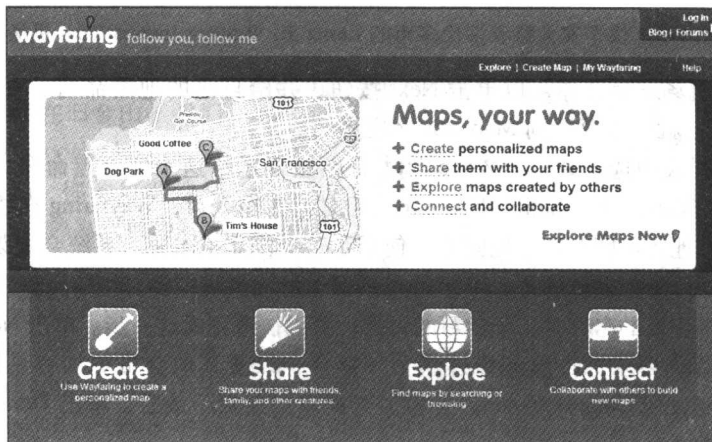


图 1-2 Wayfaring 主页

Wayfaring 是使用 Google Maps API 实现的一种地图服务，允许用户快速创建符合自己需求的地图。比如，有些人希望创建旅游地图；有些人希望能够标识住处或者城市所在地附近的兴趣区。

1551386/66

如果设想去位于加利福尼亚 Mountain View 的 Googleplex 旅行，我们一起来创建一个便捷的地图。

打开主页 <http://www.wayfaring.com>，按照链接说明申请一个账号(单击 Log In 就会看到创建新账号的选项)。在创建和激活自己的账号之后，可以通过单击页面右上角的 Create Map 链接来创建自己的地图。

1.2.1 添加第一个停靠点

首先，添加假想旅行的本地飞机场。我们准备使用位于加拿大安大略省、多伦多市的 Lester B. Pearson International 机场，但是实际上你也许会使用最近的飞机场。因为 Pearson 是家国际型机场(在美国本土之外)，你需要拖拉和放大地图才能找到它。如果你地处美国，可以使用方便的 Jump To 特性，根据文本字符串来查找这个机场。图 1-3 所显示的 Pearson 机场从缩放等级和显示位置等角度来讲都是合适的。

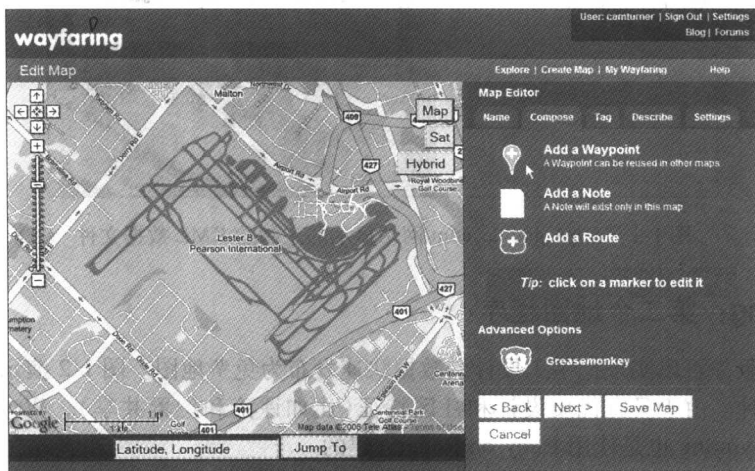


图 1-3 位于安大略省多伦多市的 Lester B. Pearson International 机场

在找到这个飞机场之后，你可以单击 Next 按钮，并给自己的地图命名。再次单击 Next 按钮(在地图命名之后)，就可以返回到 Map Editor 主界面。

从右边的选项列表中选择 Add a Waypoint，你会被提示给这个停靠点命名。我们把这个停靠点命名为 Lester B. Pearson International Airport。但是，在输入时，Wayfaring 会提示你输入准确的名称。这意味着，其他人可能在其他地图上也使用了这个停靠点名称，系统会建议你对这个停靠点使用自己的命名方式。令人欣慰的是，Wayfaring 所覆盖的飞机场很多，相信能够满足你的出行要求，可以放心地使用系统建议的名称。出于学习的目的，我们快速完成这一步。单击 Next 按钮继续。

接下来的两个界面要求你拖拉和描述这个停靠点，以便其他会员更加容易搜索到你的地图。我们添加“airport Toronto Ontario Canada”标签，并简单描述它。最后，单击 Done 把这个停靠点提交给系统，再返回到 Map Editor 主界面。

1.2.2 添加飞行路线

接下来需要向地图系统添加的对象是路线(route)。一条路线是由很多你喜欢的停靠点组成的旅游线路。在本例中，我们使用两条路线。第一条路线是两个飞机场之间的直线，有利于粗略评

估飞机把我们运到 Google 总部需要飞行的大致距离；第二条路线用于描出从旧金山 (San Francisco) 飞机场到 Googleplex 的行车路线。

首先，单击 Add a Route 按钮，给这个路线命名(比如 airplane trip)，然后单击自己希望使用的飞机场。这时候，一个小白色圆点就会出现在所单击的位置。这就是这条路线的第一个停靠点。现在，缩小地图，焦点转移到加利福尼亚(California)，并在旧金山(San Francisco)位置放大地图，你会发现飞机场就在海湾的西边。再次单击这个飞机场。如图 1-4 所示，这个飞机场处也出现一个白色圆点，并且一条蓝色的线把这两个白色圆点连接起来。在屏幕右边路线标签的下面，会看到这条路线的距离。这段路线的距离超过 2000 英里。在上述操作过程中，如果无意间创建其他的停靠点，可以使用 Undo Last 选项取消所做的操作。如果确信上述操作没有问题，单击 Save 按钮就可以保存这条路线。

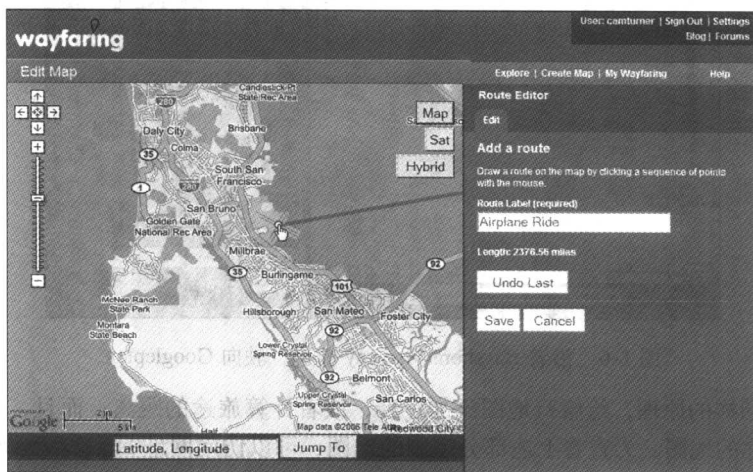


图 1-4 到达旧金山国际机场的飞行路线

1.2.3 添加目的地停靠点

现在假设你已经在旧金山，我们考虑如何直接到达 Googleplex。单击 Add a Waypoint 按钮，你的目的地是 Google，所以这个新的停靠点就命名为 The Googleplex。使用地址框特性，可以直接把地图界面切换到 1600 Amphitheatre Parkway, Mountain View, California, 94043。Wayfaring 使用一个名为地理解码 (geocoding) 的进程来确定地址的纬度和经度，本书第 4 章会更详细讨论。

为确保自己所在的位置是正确

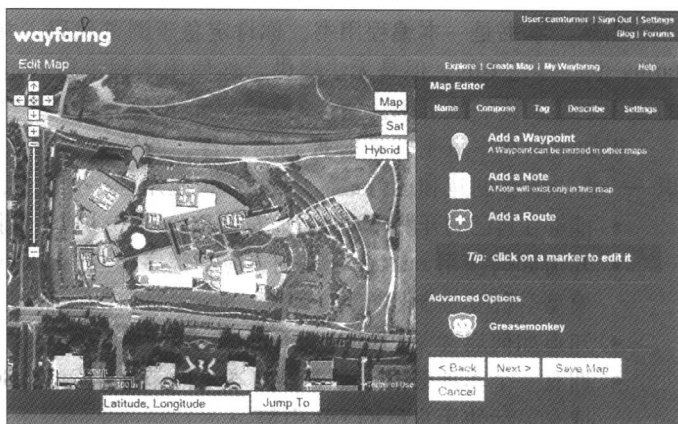


图 1-5 The Googleplex

的, 单击界面右上角的 Sat 按钮, 把地图界面切换为卫星模式。显示界面类似于图 1-5。

1.2.4 添加行车路线

接下来, 我们计算行车路线的距离。在 Wayfaring 地图界面上, 很难同时看到路线的起始点和终止点, 所以我们从 Googleplex 开始往回走。重新切换到 Map 模式或者 Hybrid 模式, 这样道路看起来会更加清晰。在 Map Editor 界面中, 选择 Add a Route, 并单击刚才添加的白色圆点。使用 10 个到 20 个左右的圆点, 这样可以清楚地追溯从 Mountain View, 沿着 Bayshore Freeway (美国 101 公路), 到达飞机场的路径。图 1-6 所显示的界面就是你驾车行驶了 23 公里时的地图界面。

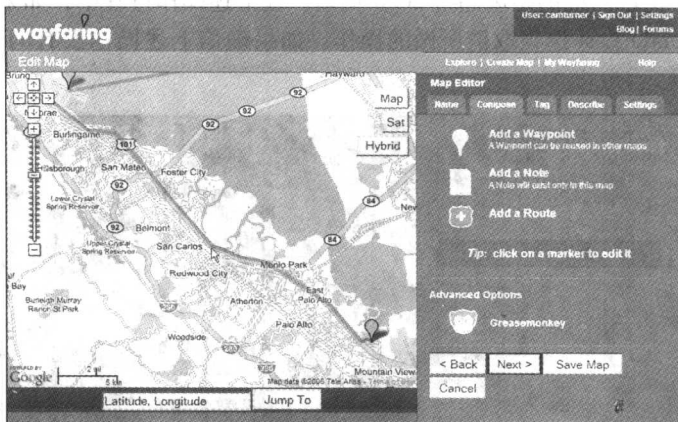


图 1-6 沿着 Bayshore Freeway 前进, 驶向 Googleplex

你可以使用同样的原理, 给旅行地图添加注释, 或者计算旅途的距离; 而且, 可以非常容易地与其他会员共享你的地图。为查看上述部分定制的地图, 可以访问网址 <http://www.wayfaring.com/maps/show/17131>。

1.3 已经掌握 Rails 吗

当然, 本书是一本编程用书, 你肯定急于浏览代码, 并试图创建自己独特的应用程序。Wayfaring 也许是完美的, 但是无法提供给你使用编程手段改造它的灵活性。为了了解有关 Maps API 的编程之道, 我们接下来开始讨论 Ruby on Rails, 这是一个服务器端框架。

正如我们在本章刚开始时提到的那样, 阅读本书时假定你已经具备 Ruby 和 Rails 的有关知识——至少能够创建和运行最简单的 Rails 应用程序。除此之外, 你还应该:

- 能够在自己的开发环境机器上安装 Ruby (1.8.4 或者更新版本) 和 Rails (1.1.5 或者更新版本)。
- 知道如何使用 rails 命令来创建应用程序构架 (skeleton), 并且能够在 Rails 应用程序中使用模型/视图/控制器布局。
- 知道如何启动 WEBrick、Mongrel 或者其他服务器, 以在 Web 浏览器中查看自己的页面。
- 拥有一个本地数据库服务器。本书的范例使用 MySQL, 但是你应该能够根据自己的实际情况灵活使用自己的 DB 引擎。
- 拥有一个自己非常熟悉和喜欢的开发环境——比如 TextMate、RadRails 或者 Vim。

在阅读本书时, 不必掌握以下内容:

- 不必了解 prototype.js JavaScript 库的细节内容。很多开发者很容易被 Prototype 搞混，估计其中主要原因是 Prototype 缺乏有效的说明文档。在编写本书时，我们做出精明决策，有意识地避免过于依赖 prototype.js(当然偶尔也会使用到它)。
- 不必学习 RJS。RJS 是一种 Rails 模板类型(类似于 RHTML)。借助它，在使用极少 JavaScript 代码的前提下，就可以创建具有丰富功能的 Web 页面。因为 Google Maps API 是使用 JavaScript 实现的，所以在编写本书时，我们主要使用手工编码实现的 JavaScript 函数。
- 不必对 JavaScript 编程非常精通。虽然，Google Maps API 是使用 JavaScript 语言实现的，但是在阅读本书时不需要对 JavaScript 技术和原理非常精通。
- 不必为自己的 Rails 应用程序搭建在线的 Web 空间。当然，你也许希望拥有一个产品服务器，这样的话就可以向全世界展示自己的成果，但是在阅读本书时，使用本地的开发服务器就可以完全掌握本书讲述的内容。

你刚学习 Ruby 和 Rails 吗？

如果你刚接触 Ruby 和 Rails，并且认定编写地图应用程序是学习和掌握两者的不错方式，也许你需要更多资源，这样能够提升学习速度。Ruby on Rails 的学习难度比 PHP 或者 ColdFusion 的学习难度大，相信如下的资源对你会有所帮助：

- Peter Cooper 编著的《Beginning Ruby: From Novice to Professional》^①(<http://www.apress.com/book/bookDisplay.html?bID=10244>)。我们发现，很多开发者最初是通过 Rails Web 框架才接触 Ruby 的。如果你也是如此的话，我强烈建议你在深入研究 Rails 之前，最好熟练掌握 Ruby 语言。
- Cloves Carneiro Jr. 和 Jeffrey Allan Hardy 编著的《Beginning Rails: From Novice to Professional》^②(<http://www.apress.com/book/bookDisplay.html?bID=10124>)。

1.4 接下来的内容

我们希望你充满渴望地、循序渐进地使用 Rails 创建自己的基于地图的应用程序。在本书第一部分结束之后，使用 JavaScript 和 XHTML，你就有能力完成刚才 Wayfaring 演示的操作(除了折线和距离第 10 章才会讲解到)。在学习完本书后，你就会掌握创建自己的 Wayfaring 所需的大部分概念。

那么接下来从什么地方开始学习呢？本书分为三大部分和两个附录。第一部分覆盖第 1 章到第 4 章，讨论与本书主题相关的基础知识。你将完成一个地图，添加一些定制的标识，并使用免费服务的地理代码(geocode)数据集。第二部分(从第 5 章到第 8 章)更深入讨论有关地图开发的主题，比如创建有价值的界面，处理超大规模的数据点集合，以及查找有利于专业地图应用程序实现的原始信息源等。第三部分(第 9 章到第 11 章)讨论一些高级话题：创建定制的地图覆盖层(overlay)，比如自己的信息窗口和工具提示；创建自己的地图块和投影；使用必要的球面方程式计算地球的表面面积；从头创建自己的地理解码器(geocoder)。最后，第一个附录提供有关 Google Maps 2 API 的参考指南；第二个附录给出一些参考资源，有助于扩展本书的范例和给自己的项目增色。

① 该书简体中文版将由机械工业出版社在 2008 年出版，书名为《Ruby 从入门到精通》。——编辑注

② 该书简体中文版将由机械工业出版社在 2008 年出版，书名为《Rails 开发者指南》。——编辑注

第2章 开始行动

在本章中，你将学习如何创建自己的第一个 Google 地图项目、绘制一些标记(marker)，并添加一些交互功能。由于 JavaScript 对控制地图操作非常关键，所以最好在开始之前温习 JavaScript 的基础知识。

在本章中，我们将掌握如下内容：

- 建立自己的 Rails 应用程序。
- 创建一个基本的地图和获取 Google Maps API 密钥。
- 把地图应用程序的 JavaScript 函数和 XHTML 相分离。
- 卸载未完成的地图，以帮助浏览器释放内存。
- 创建地图标记，使用信息弹出式窗口响应对标记的单击操作。

2.1 关于 JavaScript、帮助器和插件

对于很多常见的 JavaScript 和 Ajax 用例而言，通过帮助器库(helper libraries)实现 Rails 与 Prototype 和 Scriptaculous 的集成是天赐之物。因为帮助器覆盖很多常见的情况，很多 Rails 开发者无需深入研究 JavaScript 本身。如果这恰恰适合于你的话，那就可以开始学习 JavaScript 了。Google Maps API 是 100% 的纯 JavaScript，你需要熟悉 JavaScript，这样才能充分利用 API 提供的所有价值。

类似的，RubyForge 列出一些 Google Maps 相关的插件，这些插件提供 Google Maps 专用的帮助器。RubyForge 上列出的 Google Maps 相关插件包括 Cartographer(<http://cartographer.rubyforge.org>)和 YM4R(<http://rubyforge.org/projects/ym4r>)。Cartographer 已经存在很长时间了(相对于 RubyForge 而言)，但是 YM4R 更成熟，功能更加丰富。现在看起来，YM4R 更新得也非常快，并保持旺盛的活力。

借助于极少的前端 JavaScript 代码，YM4R 就可以提供在 Rails 上使用 Google Maps 的绝佳方式。如果选择使用 YM4R，可以从 http://thepochisuperstarmegashow.com/ProjectsDoc/ym4r_gm-doc/ 找到相关参考文档。

本书所采用的方法是使用 JavaScript 和 Google Maps，而 JavaScript 是使用该 API 的指定语言。这样做有几方面好处：

- 可以充分使用这种 API 的全部功能。自己的代码和 Google Maps 之间不需要任何中介。
- 当自己的代码出现问题时，调试难度会比较小。
- 可以让自己的地图应用程序更及时地利用 Google Maps API 的最新更新。Google 已经证实，Maps API 将不断发展演化。如果使用原始的 API，将不能利用第三方来升级一个插件以利用最新提供的能力。

最后，我们相信，使用 JavaScript 来学习 Google Maps API，能够获得所需的服务和实现自己的目的。如果后来决定使用类似于 YM4R 的插件，你需要深刻理解该插件所产生的 JavaScript 代码。在必要的时候，你也能够扩展这个插件，或者与原始的 JavaScript 代码混合输出。