PACKT PUBLISHING

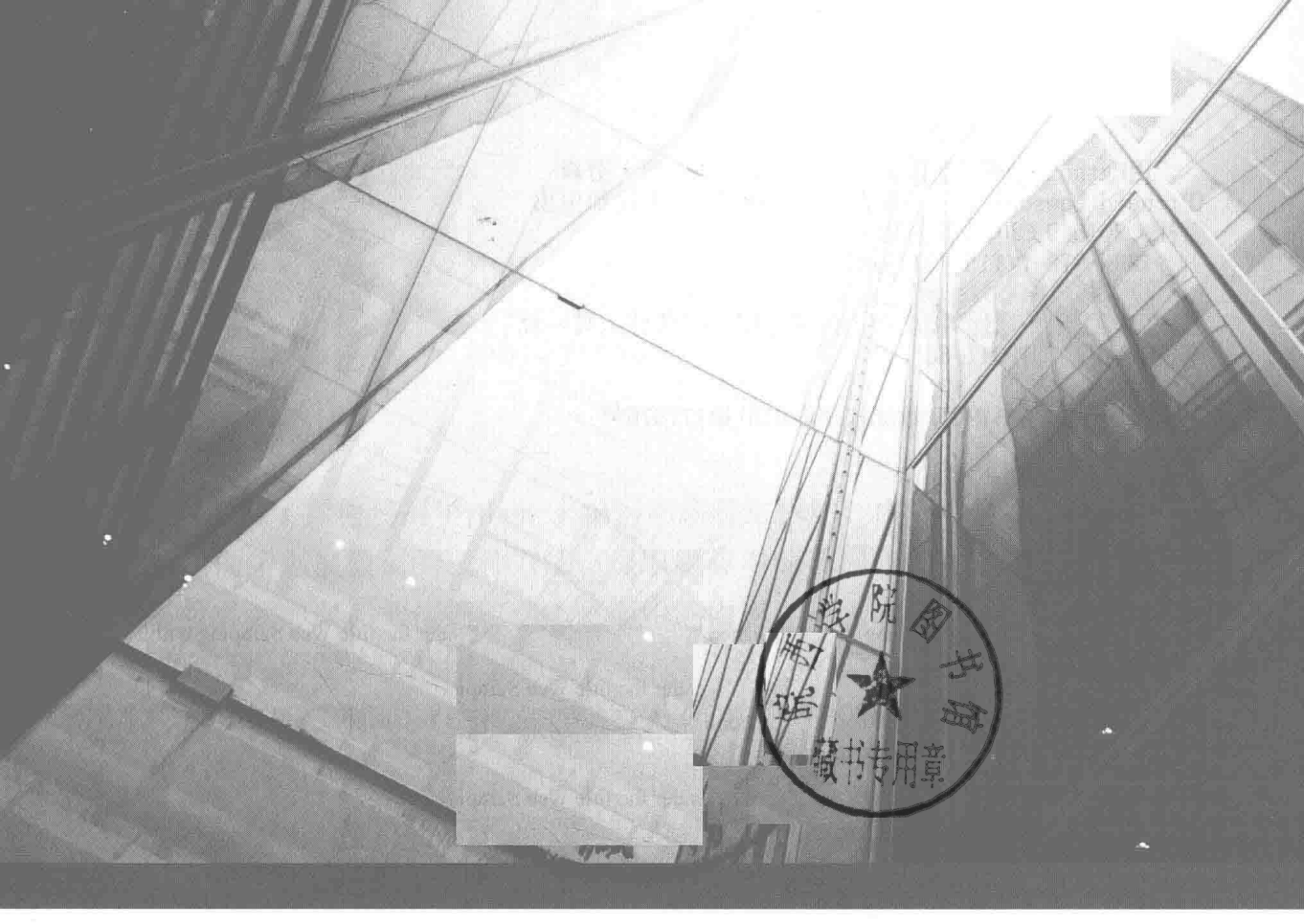
用Python写网络爬虫

Web Scraping with Python

Scrape data from any website with the power of Python

[澳] Richard Lawson 著
李斌 译

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS



用Python写网络爬虫

[澳] Richard Lawson 著
李斌 译

人民邮电出版社
北京

图书在版编目（CIP）数据

用Python写网络爬虫 /（澳大利亚）理查德·劳森（Richard Lawson）著；李斌译. -- 北京：人民邮电出版社，2016.9（2016.11重印）
ISBN 978-7-115-43179-0

Ⅰ. ①用… Ⅱ. ①理… ②李… Ⅲ. ①软件工具－程序设计 Ⅳ. ①TP311.56

中国版本图书馆CIP数据核字(2016)第177976号

版权声明

◆ 著 [澳] Richard Lawson
译 李 斌
责任编辑 傅道坤
责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
三河市海波印务有限公司印刷

◆ 开本：800×1000 1/16
印张：10.75
字数：148 千字 2016 年 9 月第 1 版
印数：11 001－16 000册 2016 年 11 月河北第 5 次印刷

著作权合同登记号 图字：01-2016-3962 号

定价：45.00 元

读者服务热线：(010)81055410 印装质量热线：(010)81055316
反盗版热线：(010)81055315

内容提要

本书讲解了如何使用 Python 来编写网络爬虫程序，内容包括网络爬虫简介，从页面中抓取数据的三种方法，提取缓存中的数据，使用多个线程和进程来进行并发抓取，如何抓取动态页面中的内容，与表单进行交互，处理页面中的验证码问题，以及使用 Scarpy 和 Portia 来进行数据抓取，并在最后使用本书介绍的数据抓取技术对几个真实的网站进行了抓取，旨在帮助读者活学活用书中介绍的技术。

本书适合有一定 Python 编程经验，而且对爬虫技术感兴趣的读者阅读。

关于作者

Richard Lawson 来自澳大利亚，毕业于墨尔本大学计算机科学专业。毕业后，他创办了一家专注于网络爬虫的公司，为超过 50 个国家的业务提供远程工作。他精通于世界语，可以使用汉语和韩语对话，并且积极投身于开源软件。他目前在牛津大学攻读研究生学位，并利用业余时间研发自主无人机。

我要感谢 Timothy Baldwin 教授将我引入这个令人兴奋的领域，以及本书编写时在巴黎招待我的 Tharavy Douc。

关于审稿人

Martin Burch 是一名常驻纽约的数据记者，其工作是为华尔街日报绘制交互式图表。他在新墨西哥州立大学获得了新闻学和信息系统专业的学士学位，然后在纽约城市大学新闻学研究院获得了新闻学专业硕士学位。

我要感谢我的妻子 Lisa 鼓励我协助本书的创作，我的叔叔 Michael 耐心解答我的编程问题，以及我的父亲 Richard 激发了我对新闻学和写作的热爱。

William Sankey 是一位数据专业人士，也是一位业余开发人员，生活在马里兰州科利奇帕克市。他于 2012 年毕业于约翰·霍普金斯大学，获得了公共政策硕士学位，专业方向为定量分析。他目前在 L&M 政策研究有限责任公司担任健康服务研究员，从事与美国医疗保险和医疗补助服务中心（CMS）相关的项目。这些项目包括责任医疗机构评估以及精神病院住院患者预付费系统监测。

我要感谢我深爱的妻子 Julia 和顽皮的小猫 Ruby，给予我全部的爱和支持。

Ayush Tiwari 是一名 Python 开发者，本科就读于印度理工学院罗克分校。他自 2013 年起工作于印度理工学院罗克分校信息管理小组，并活跃于网络开发领域。对他而言，审阅本书是一个非常棒的经历。他不仅是一名审稿人，也是一名狂热的网络爬虫学习者。他向所有 Python 爱好者推荐本书，以便享受爬虫的益处。

他热衷于 Python 网络爬虫，曾参与体育直播订阅、通用 Python 电子商务网络爬虫（在 Miranj）等相关项目。

他还使用 Django 应用开发了就业门户，帮助改善印度理工学院罗克分校的就业流程。

除了后端开发之外，他还喜欢使用诸如 NumPy、SciPy 等 Python 库进行科学计算和数据分析，目前他从事计算流体力学领域的研究。你可以在 GitHub 上访问到他的项目，他的用户名是 `tiwariayush`。

他喜欢徒步穿越喜马拉雅山谷，每年会参加多次徒步行走活动。此外，他还喜欢弹吉他。他的成就还包括参加国际知名的 Super 30 小组，并在其中成为排名保持者。他在高中时，还参加了国际奥林匹克数学竞赛。

我的家庭成员（我的姐姐 Aditi、我的父母以及 Anand 先生）、我在 VI 和 IMG 的朋友以及我的教授都为我提供了很大的帮助。我要感谢他们所有人对我的支持。

最后，感谢尊敬的作者和 Packt 出版社团队出版了这些非常好的技术书籍。我要对他们在编写这些书籍时的所有辛苦工作表示赞赏。

前言

互联网包含了迄今为止最有用的数据集，并且大部分可以免费公开访问。但是，这些数据难以复用。它们被嵌入在网站的结构和样式当中，需要抽取出来才能使用。从网页中抽取数据的过程又被称为网络爬虫。随着越来越多的信息被发布到网络上，网络爬虫也变得越来越有用。

本书内容

第 1 章，网络爬虫简介，介绍了网络爬虫，并讲解了爬取网站的方法。

第 2 章，数据抓取，展示了如何从网页中抽取数据。

第 3 章，下载缓存，学习了如何通过缓存结果避免重复下载的问题。

第 4 章，并发下载，通过并行下载加速数据抓取。

第 5 章，动态内容，展示了如何从动态网站中抽取数据。

第 6 章，表单交互，展示了如何与表单进行交互，从而访问你需要的数据。

第 7 章，验证码处理，阐述了如何访问被验证码图像保护的数据。

第 8 章，Scrapy，学习了如何使用流行的高级框架 Scrapy。

第 9 章，总结，对我们介绍的这些网络爬虫技术进行总结。

阅读本书的前提

本书中所有的代码都已经在 Python 2.7 环境中进行过测试，并且可以从 http://bitbucket.org/wswp/code 下载到这些源代码。理想情况下，本书未来的版本会将示例代码移植到 Python 3 当中。不过，现在依赖的很多库（比如 Scrapy/Twisted、Mechanize 和 Ghost）还只支持 Python 2。为了帮助阐明爬取示例，我们创建了一个示例网站，其网址为 http://example.webscraping.com。由于该网站限制了下载内容的速度，因此如果你希望自行搭建示例网站，可以从 http://bitbucket.org/wswp/places 获取网站源代码和安装说明。

我们决定为本书中使用的大部分示例搭建一个定制网站，而不是抓取活跃网站，这样我们就对环境拥有了完全控制。这种方式为我们提供了稳定性，因为活跃网站要比书中的定制网站更新更加频繁，并且当你尝试运行爬虫示例时，代码可能已经无法工作。另外，定制网站允许我们自定义示例，用于阐释特定技巧并避免其他干扰。最后，活跃网站可能并不欢迎我们使用它作为学习网络爬虫的对象，并且可能会尝试封禁我们的爬虫。使用我们自己定制的网站可以规避这些风险，不过在这些例子中学到的技巧确实也可以应用到这些活跃网站当中。

本书读者

阅读本书需要有一定的编程经验，并且不适用于绝对的初学者。在实践中，我们将会首先实现我们自己的网络爬虫技术版本，然后才会介绍现有的流行模块，这样可以让你更好地理解这些技术是如何工作的。本书中的这些示例将假设你已经拥有 Python 语言以及使用 pip 安装模块的能力。如果你想复习一下这些知识，有一本非常好的免费在线书籍可以使用，其作者为 Mark

Pilgrim，书籍网址是 http://www.diveintopython.net。这本书也是我初学 Python 时所使用的资源。

此外，这些例子还假设你已经了解网页是如何使用 HTML 进行构建并通过 JavaScript 更新的知识。关于 HTTP、CSS、AJAX、WebKit 以及 MongoDB 的既有知识也很有用，不过它们不是必需的，这些技术会在需要使用时进行介绍。上述很多主题的详细参考资料可以从 http://www.w3schools.com 获取到。

目录

第 1 章
网络爬虫简介

本章中，我们将会介绍如下主题：

- 网络爬虫领域简介；
- 解释合法性质疑；
- 对目标网站进行背景调研；
- 逐步完善一个高级网络爬虫。

1.1 网络爬虫何时有用

假设我有一个鞋店，并且想要及时了解竞争对手的价格。我可以每天访问他们的网站，与我店铺中鞋子的价格进行对比。但是，如果我店铺中的鞋类品种繁多，或是希望能够更加频繁地查看价格变化的话，就需要花费大量的时间，甚至难以实现。再举一个例子，我看中了一双鞋，想等它促销时再购买。我可能需要每天访问这家鞋店的网站来查看这双鞋是否降价，也许需要等待几个月的时间，我才能如愿盼到这双鞋促销。上述这两个重复性的手工流程，都可以利用本书介绍的网络爬虫技术实现自动化处理。

理想状态下，网络爬虫并不是必须品，每个网站都应该提供 API，以结构化的格式共享它们的数据。然而现实情况中，虽然一些网站已经提供了这种 API，但是它们通常会限制可以抓取的数据，以及访问这些数据的频率。另外，对于网站的开发者而言，维护前端界面比维护后端 API 接口优先级更高。总之，我们不能仅仅依赖于 API 去访问我们所需的在线数据，而是应该学习一些网络爬虫技术的相关知识。

1.2　网络爬虫是否合法

网络爬虫目前还处于早期的蛮荒阶段，“允许哪些行为”这种基本秩序还处于建设之中。从目前的实践来看，如果抓取数据的行为用于个人使用，则不存在问题；而如果数据用于转载，那么抓取的数据类型就非常关键了。

世界各地法院的一些案件可以帮助我们确定哪些网络爬虫行为是允许的。在 *Feist Publications, Inc.*起诉 *Rural Telephone Service Co.*的案件中，美国联邦最高法院裁定抓取并转载真实数据（比如，电话清单）是允许的。而在澳大利亚，*Telstra Corporation Limited* 起诉 *Phone Directories Company Pty Ltd* 这一类似案件中，则裁定只有拥有明确作者的数据，才可以获得版权。此外，在欧盟的 *ofir.dk* 起诉 *home.dk* 一案中，最终裁定定期抓取和深度链接是允许的。

这些案件告诉我们，当抓取的数据是现实生活中的真实数据（比如，营业地址、电话清单）时，是允许转载的。但是，如果是原创数据（比如，意见和评论），通常就会受到版权限制，而不能转载。

无论如何，当你抓取某个网站的数据时，请记住自己是该网站的访客，应当约束自己的抓取行为，否则他们可能会封禁你的 IP，甚至采取更进一步的法律行动。这就要求下载请求的速度需要限定在一个合理值之内，并且还需要设定一个专属的用户代理来标识自己。在下面的小节中我们将会对这些实践进行具体介绍。

关于上述几个法律案件的更多信息可以参考下述地址：

- http://caselaw.lp.findlaw.com/scripts/getcase. pl?court=US&vol=499&invol=340
- http://www.austlii.edu.au/au/cases/cth/FCA/2010/44.html
- http://www.bvhd.dk/uploads/tx_mocarticles/S_og_Handelsrettens_afg_relse_i_Ofir-sagen.pdf

1.3 背景调研

在深入讨论爬取一个网站之前，我们首先需要对目标站点的规模和结构进行一定程度的了解。网站自身的 robots.txt 和 Sitemap 文件都可以为我们提供一定的帮助，此外还有一些能提供更详细信息的外部工具，比如 Google 搜索和 WHOIS。

1.3.1 检查 robots.txt

大多数网站都会定义 robots.txt 文件，这样可以让爬虫了解爬取该网站时存在哪些限制。这些限制虽然仅仅作为建议给出，但是良好的网络公民都应当遵守这些限制。在爬取之前，检查 robots.txt 文件这一宝贵资源可以最小化爬虫被封禁的可能，而且还能发现和网站结构相关的线索。关于 robots.txt 协议的更多信息可以参见 http://www.robotstxt.org。下面的代码是我们的示例文件 robots.txt 中的内容，可以访问 http://example.webscraping.com/robots.txt 获取。

```
# section 1
User-agent: BadCrawler
Disallow: /
```