O'REILLY®

TURING 图灵程序设计丛书

Python 网络数据采集

Web Scraping with Python

[美] Ryan Mitchell 著
陶俊杰 陈小莉 译

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

TURING 图灵程序设计丛书

Python网络数据采集

Web Scraping with Python

Collecting Data from the Modern Web

[美] Ryan Mitchell 著
陶俊杰 陈小莉 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc.授权人民邮电出版社出版

人民邮电出版社
北 京

图书在版编目（CIP）数据

Python网络数据采集 / （美）米切尔（Mitchell, R.）著 ; 陶俊杰, 陈小莉译. -- 北京 : 人民邮电出版社, 2016.3
（图灵程序设计丛书）
ISBN 978-7-115-41629-2

Ⅰ. ①P… Ⅱ. ①米… ②陶… ③陈… 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023324号

内 容 提 要

本书采用简洁强大的Python语言，介绍了网络数据采集，并为采集新式网络中的各种数据类型提供了全面的指导。第一部分重点介绍网络数据采集的基本原理：如何用Python从网络服务器请求信息，如何对服务器的响应进行基本处理，以及如何以自动化手段与网站进行交互。第二部分介绍如何用网络爬虫测试网站，自动化处理，以及如何通过更多的方式接入网络。

本书适合需要采集Web数据的相关软件开发人员和研究人员阅读。

◆ 著 [美] Ryan Mitchell
译 陶俊杰 陈小莉
责任编辑 岳新欣
执行编辑 李 敏
责任印制 杨林杰

◆ 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
三河市中晟雅豪印务有限公司印刷

◆ 开本：800×1000 1/16
印张：13.5
字数：280千字 2016年3月第1版
印数：1－4 000册 2016年3月河北第1次印刷
著作权合同登记号 图字：01-2015-8108号

定价：59.00元

读者服务热线：(010)51095186转600 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广字第8052号

译者序

每时每刻，搜索引擎和网站都在采集大量信息，非原创即采集。采集信息用的程序一般被称为网络爬虫（Web crawler）、网络铲（Web scraper，可类比考古用的洛阳铲）、网络蜘蛛（Web spider），其行为一般是先“爬”到对应的网页上，再把需要的信息“铲”下来。O'Reilly 这本书的封面图案是一只穿山甲，图灵公司把这本书的中文版定名为“Python 网络数据采集”。当我们看完这本书的时候，觉得网络数据采集程序也像是一只辛勤采蜜的小蜜蜂，它飞到花（目标网页）上，采集花粉（需要的信息），经过处理（数据清洗、存储）变成蜂蜜（可用的数据）。网络数据采集可以为生活加点儿蜜，亦如本书作者所说，“网络数据采集是为普通大众所喜闻乐见的计算机巫术”。

网络数据采集大有所为。在大数据深入人心的时代，网络数据采集作为网络、数据库与机器学习等领域的交汇点，已经成为满足个性化网络数据需求的最佳实践。搜索引擎可以满足人们对数据的共性需求，即“我来了，我看见”，而网络数据采集技术可以进一步精炼数据，把网络中杂乱无章的数据聚合成合理规范的形式，方便分析与挖掘，真正实现“我征服”。工作中，你可能经常为找数据而烦恼，或者眼睁睁看着眼前的几百页数据却只能长恨咫尺天涯，又或者数据杂乱无章的网站中满是带有陷阱的表单和坑爹的验证码，甚至需要的数据都在网页版的 PDF 和网络图片中。而作为一名网站管理员，你也需要了解常用的网络数据采集手段，以及常用的网络表单安全措施，以提高网站访问的安全性，所谓道高一尺，魔高一丈……一念清净，烈焰成池，一念觉醒，方登彼岸，本书试图成为解决这些问题的一念，让你茅塞顿开，船登彼岸。

网络数据采集并不是一门语言的独门秘籍，Python、Java、PHP、C#、Go 等语言都可以讲出精彩的故事。有人说编程语言就是宗教，不同语言的设计哲学不同，行为方式各异，“非我族类，其心必异”，但本着美好生活、快乐修行的初衷，我们对所有语言都时刻保持敬畏之心，尊重信仰自由，努力做好自己的功课。对爱好 Python 的人来说，人生苦短，Python 当歌！简洁轻松的语法，开箱即用的模块，强大快乐的社区，总可以快速构建出简单高效的解决方案。使用 Python 的日子总是充满快乐的，本书关于 Python 网络数

据采集的故事也不例外。网络数据采集涉及多个领域，内容包罗万象，因此本书覆盖的主题较多，涉及的知识面相对广阔，书中介绍的 Python 模块有 urllib、BeautifulSoup、lxml、Scrapy、PdfMiner、Requests、Selenium、NLTK、Pillow、unittest、PySocks 等，还有一些知名网站的 API、MySQL 数据库、OpenRefine 数据分析工具、PhanthomJS 无头浏览器以及 Tor 代理服务器等内容。每行到一处，皆是风景独好，而且作者也为每一个主题提供了深入研究的参考资料。不过，本书关于多进程（multiprocessing）、并发（concurrency）、集群（cluster）等高性能采集主题着墨不多，更加关注性能的读者，可以参考其他关于 Python 高性能和多核编程的书籍。总之，本书通俗易懂，简单易行，有编程基础的同学都可以阅读。不会 Python？抽一节课时间学一下吧。

网络数据采集也应该有所不为。国内外关于网络数据保护的法律法规都在不断地制定与完善中，本书作者在书中介绍了美国与网络数据采集相关的法律与典型案例，呼吁网络爬虫严格控制网络数据采集的速度，降低被采集网站服务器的负担。恶意消耗别人网站的服务器资源，甚至拖垮别人网站是一件不道德的事情。众所周知，这已经不仅仅是一句“吸烟有害健康”之类的空洞口号，它可能导致更严重的法律后果，且行且珍惜！

语言是思想的解释器，书籍是语言的载体。本书英文原著是作者用英文解释器为自己思想写的载体，而译本是译者根据英文原著以及与作者的交流，用简体中文解释器为作者思想写的载体。读者拿到的中译本，是作者思想经过两层解释器转换的结果，其目的是希望帮助中文读者消除语言障碍，理解作者的思想，与作者产生共鸣，一起面对作者曾经遇到的问题，共同探索解决问题的方法，从而帮助读者提高解决问题的能力，增强直面 bug 的信心。bug 是产品生命中的挑战，好产品是不断面对 bug 并战胜 bug 的结果。译者水平有限，译文 bug 也在所难免，翻译有不到之处，还请各位读者批评指正！

最后要感谢图灵公司朱巍老师的大力支持，让译作得以顺利出版。也要感谢神烦小宝的温馨陪伴，每天 6 点叫我们起床，让业余时间格外宽裕。

译者联系方式——
邮箱：muxuezi@gmail.com，微信号：muxuezi
邮箱：carrieforchen@gmail.com，微信号：陈小莉

陶俊杰
2015 年 10 月

前言

对那些没有学过编程的人来说，计算机编程看着就像变魔术。如果编程是魔术（magic），那么网络数据采集（Web scraping）就是巫术（wizardry）；也就是运用“魔术”来实现精彩实用却又不费吹灰之力的“壮举”。

说句实话，在我的软件工程师职业生涯中，我几乎没有发现像网络数据采集这样的编程实践，可以同时吸引程序员和门外汉的注意。虽然写一个简单的网络爬虫并不难，就是先收集数据，再显示到命令行或者存储到数据库里，但是无论你之前已经做过多少次了，这件事永远会让你感到兴奋，同时又有新的可能。

不过遗憾的是，当和别的程序员提起网络数据采集时，我听到了很多关于这件事的误解与困惑。有些人不确定它是不是合法的（其实合法），有人不明白怎么处理那些到处都是JavaScript、多媒体和 cookie 的新式网站，还有人对 API 和网络爬虫的区别感到困惑。

这本书的初衷是要解决人们对网络数据采集的诸多问题与误解，并对常见的网络数据采集任务提供全面的指导。

从第 1 章开始，我将不断地提供代码示例来演示书中内容。这些代码示例是开源的，无论注明出处与否都可以免费使用（但若注明会让作者感激不尽）。所有的代码示例都在 GitHub 网站上（https://github.com/REMitchell/python-scraping），可以查看和下载。

什么是网络数据采集

在互联网上进行自动数据采集这件事和互联网存在的时间差不多一样长。虽然网络数据采集并不是新术语，但是多年以来，这件事更常见的称谓是网页抓屏（screen scraping）、数据挖掘（data mining）、网络收割（Web harvesting）或其他类似的版本。今天大众好像更倾向于用“网络数据采集”，因此我在本书中使用这个术语，不过有时会把网络数据采集程序称为网络机器人（bots）。

理论上，网络数据采集是一种通过多种手段收集网络数据的方式，不光是通过与 API 交互（或者直接与浏览器交互）的方式。最常用的方法是写一个自动化程序向网络服务器请求数据（通常是用 HTML 表单或其他网页文件），然后对数据进行解析，提取需要的信息。

实践中，网络数据采集涉及非常广泛的编程技术和手段，比如数据分析、信息安全等。本书将在第一部分介绍关于网络数据采集和网络爬行（crawling）的基础知识，一些高级主题放在第二部分介绍。

为什么要做网络数据采集

如果你上网的唯一方式就是用浏览器，那么你其实失去了很多种可能。虽然浏览器可以更方便地执行 JavaScript，显示图片，并且可以把数据展示成更适合人类阅读的形式，但是网络爬虫收集和处理大量数据的能力更为卓越。不像狭窄的显示器窗口一次只能让你看一个网页，网络爬虫可以让你一次查看几千甚至几百万个网页。

另外，网络爬虫可以完成传统搜索引擎不能做的事情。用 Google 搜索“飞往波士顿最便宜的航班”，看到的是大量的广告和主流的航班搜索网站。Google 只知道这些网站的网页会显示什么内容，却不知道在航班搜索应用中输入的各种查询的准确结果。但是，设计较好的网络爬虫可以通过采集大量的网站数据，做出飞往波士顿航班价格随时间变化的图表，告诉你买机票的最佳时间。

你可能会问：“数据不是可以通过 API 获取吗？”（如果你不熟悉 API，请阅读第 4 章。）确实，如果你能找到一个可以解决你的问题的 API，那会非常给力。它们可以非常方便地向用户提供服务器里格式完好的数据。当你使用像 Twitter 或维基百科的 API 时，会发现一个 API 同时提供了不同的数据类型。通常，如果有 API 可用，API 确实会比写一个网络爬虫程序来获取数据更加方便。但是，很多时候你需要的 API 并不存在，这是因为：

- 你要收集的数据来自不同的网站，没有一个综合多个网站数据的 API；
- 你想要的数据非常小众，网站不会为你单独做一个 API；
- 一些网站没有基础设施或技术能力去建立 API。

即使 API 已经存在，可能还会有请求内容和次数限制，API 能够提供的数据类型或者数据格式可能也无法满足你的需求。

这时网络数据采集就派上用场了。你在浏览器上看到的内容，大部分都可以通过编写 Python 程序来获取。如果你可以通过程序获取数据，那么就可以把数据存储到数据库里。如果你可以把数据存储到数据库里，自然也就可以将这些数据可视化。

显然，大量的应用场景都会需要这种几乎可以毫无阻碍地获取数据的手段：市场预测、机器语言翻译，甚至医疗诊断领域，通过对新闻网站、文章以及健康论坛中的数据进行采集

和分析，也可以获得很多好处。

甚至在艺术领域，网络数据采集也为艺术创作开辟了新方向。由 Jonathan Harris 和 Sep Kamvar 在 2006 年发起的“我们感觉挺好”（We Feel Fine，http://wefeelfine.org/）项目，从大量英文博客中抓取许多以“I feel”和“I am feeling”开头的短句，最终做成了一个很受大众欢迎的数据可视图，描述了这个世界每天、每分钟的感觉。

无论你现在处于哪个领域，网络数据采集都可以让你的工作更高效，帮你提升生产力，甚至开创一个全新的领域。

关于本书

本书不仅介绍了网络数据采集，也为采集新式网络中的各种数据类型提供了全面的指导。虽然本书用的是 Python 编程语言，里面涉及 Python 的许多基础知识，但这并不是一本 Python 入门图书。

如果你不太懂编程，也完全不了解 Python，那么这本书看起来可能有点儿费劲。但是，如果你懂编程，那么书中的内容可以很快上手。附录 A 介绍了 Python 3.x 版本的安装和使用方法，全书将使用这个版本的 Python。如果你的电脑里只装了 Python 2.x 版本，可能需要先看看附录 A。

如果你想更全面地学习 Python，Bill Lubanovic 写的《Python 语言及其应用》[1] 是本非常好的教材，只是书有点儿厚。如果不想看书，Jessica McKellar 的教学视频 Introduction to Python（http://shop.oreilly.com/product/110000448.do）也非常不错。

附录 C 介绍并分析了几个商业案例以及犯罪事件，可以帮助你了解如何在美国合法地运行网络爬虫并使用数据。

技术书通常都是介绍一种语言或技术，而网络数据采集是一个比较综合的主题，涉及数据库、网络服务器、HTTP 协议、HTML 语言、网络安全、图像处理、数据科学等内容。本书尝试涵盖网络数据采集的所有内容。

第一部分深入讲解网络数据采集和网络爬行相关内容，并重点介绍全书都要用到的几个 Python 库。这部分内容可以看成这些库和技术的综合参考（对于一些特殊情形，后面会提供其他参考资料）。

第二部分介绍读者在动手编写网络爬虫的过程中可能会涉及的一些主题。不过，这些主题的范围特别广泛，这部分内容也不足以道尽玄机。因此，文中提供了许多常用的参考资料来补充更多的信息。

注 1：中文版已经由人民邮电出版社出版。——编者注

本书结构组织灵活，便于你直接跳到感兴趣的章节中阅读相应的网络数据采集技术。如果一个概念或一段代码在之前的章节中出现过，那么我会明确标注出具体的位置。

排版约定

本书使用了下列排版约定。

- 楷体
  表示新术语。
- 等宽字体（`constant width`）
  表示程序片段，以及正文中出现的变量、函数名、数据库、数据类型、环境变量、语句和关键字等。
- 加粗等宽字体（**`constant width bold`**）
  表示应该由用户输入的命令或其他文本。
- 斜体等宽字体（***`constant width italic`***）
  表示应该替换成用户输入的值，或根据上下文替换的值。

该图标表示提示或建议。

该图标表示一般性说明。

该图标表示警告或警示。

使用代码示例

补充材料（代码示例、练习等）可以从 https://github.com/REMitchell/python-scraping 下载。

本书是要帮你完成工作的。一般来说，如果本书提供了示例代码，你可以把它用在你的程序或文档中。除非你使用了很大一部分代码，否则无需联系我们获得许可。比如，用本书

的几个代码片段写一个程序就无需获得许可，销售或分发 O'Reilly 图书的示例光盘则需要获得许可；引用本书中的示例代码回答问题无需获得许可，将书中大量的代码放到你的产品文档中则需要获得许可。

我们很希望但并不强制要求你在引用本书内容时加上引用说明。引用说明一般包括书名、作者、出版社和 ISBN。比如："*Web Scraping with Python* by Ryan Mitchell (O'Reilly). Copyright 2015 Ryan Mitchell, 978-1-491-91029-0."

如果你觉得自己对示例代码的用法超出了上述许可的范围，欢迎你通过 permissions@oreilly.com 与我们联系。

Safari® Books Online

Safari Books Online（http://www.safaribooksonline.com）是应运而生的数字图书馆。它同时以图书和视频的形式出版世界顶级技术和商务作家的专业作品。技术专家、软件开发人员、Web 设计师、商务人士和创意专家等，在开展调研、解决问题、学习和认证培训时，都将 Safari Books Online 视作获取资料的首选渠道。

对于组织团体、政府机构和个人，Safari Books Online 提供各种产品组合和灵活的定价策略。用户可通过一个功能完备的数据库检索系统访问 O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 以及其他几十家出版社的上千种图书、培训视频和正式出版之前的书稿。要了解 Safari Books Online 的更多信息，我们网上见。

联系我们

请把对本书的评价和问题发给出版社。

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室（100035）
奥莱利技术咨询（北京）有限公司

O'Reilly 的每一本书都有专属网页，你可以在那儿找到本书的相关信息，包括勘误表、示例代码以及其他信息。本书的网站地址是：

http://oreil.ly/1ePG2Uj

对于本书的评论和技术性问题，请发送电子邮件到：bookquestions@oreilly.com

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息，请访问以下网站：

http://www.oreilly.com

我们在 Facebook 的地址如下：http://facebook.com/oreilly

请关注我们的 Twitter 动态：http://twitter.com/oreillymedia

我们的 YouTube 视频地址如下：http://www.youtube.com/oreillymedia

致谢

和那些基于海量用户反馈诞生的优秀产品一样，如果没有许多协作者、支持者和编辑的帮助，本书可能永远都不会出版。首先要感谢 O'Reilly 团队对这个小众主题图书的大力支持，感谢我的朋友和家人阅读初稿并提出宝贵的建议，还要感谢和我一起在 LinkeDrive 奋战的同事们帮我分担了很多工作。

尤其要感谢 Allyson MacDonald、Brian Anderson、Miguel Grinberg 和 Eric VanWyk 的建议、指导和偶尔的爱之深责之切。有一些章节和代码示例是根据他们的建议写成的。

还要感谢 Yale Specht 过去九个月用无尽的耐心和鼓励促成了这个项目，并在我的写作过程中对文体提出了宝贵的建议。没有他，这本书可能只用一半时间就能写完，但是不会像现在这么实用。

最后，要感谢 Jim Waldo，是他许多年前给一个小孩邮寄了一个 Linux 机箱和 *The Art and Science of C* 那本书，帮她开启了计算机世界的大门。

目录

第二部分　高级数据采集

第一部分

创建爬虫

这部分内容重点介绍网络数据采集的基本原理：如何用 Python 从网络服务器请求信息，如何对服务器的响应进行基本处理，以及如何以自动化手段与网站进行交互。最终，你将轻松游弋于网络空间，创建出具有域名切换、信息收集以及信息存储功能的爬虫。

说实话，如果你想以较少的预先投入获取较高的回报，网络数据采集肯定是一个值得踏入的神奇领域。大体上，你遇到的 90% 的网络数据采集项目使用的都是接下来的六章里介绍的技术。这部分内容涵盖了一般人（也包括技术达人）在思考“网络爬虫”时通常的想法：

- 通过网站域名获取 HTML 数据
- 根据目标信息解析数据
- 存储目标信息
- 如果有必要，移动到另一个网页重复这个过程

这将为你学习本书第二部分中更复杂的项目奠定坚实的基础。不要天真地认为这部分内容没有第二部分里的一些比较高级的项目重要。其实，当你写自己的网络爬虫时，几乎每天都要用到第一部分的所有内容。

第1章

初见网络爬虫

一旦你开始采集网络数据，就会感受到浏览器为我们做的所有细节。网络上如果没有HTML文本格式层、CSS样式层、JavaScript执行层和图像渲染层，乍看起来会有点儿吓人，但是在这一章和下一章，我们将介绍如何不通过浏览器的帮助来格式化和理解数据。

本章将首先向网络服务器发送GET请求以获取具体网页，再从网页中读取HTML内容，最后做一些简单的信息提取，将我们要寻找的内容分离出来。

1.1 网络连接

如果你没在网络或网络安全上花过太多时间，那么互联网的原理可能看起来有点儿神秘。准确地说，每当打开浏览器连接http://google.com的时候，我们不会思考网络正在做什么，而且如今也不必思考。实际上，我认为很神奇的是，计算机接口已经如此先进，让大多数人上网的时候完全不思考网络是如何工作的。

但是，网络数据采集需要抛开一些接口的遮挡，不仅是在浏览器层（它如何解释所有的HTML、CSS和JavaScript），有时也包括网络连接层。

我们通过下面的例子让你对浏览器获取信息的过程有一个基本的认识。Alice有一台网络服务器。Bob有一个台式机正准备连接Alice的服务器。当一台机器想与另一台机器对话时，下面的某个行为将会发生。

1. Bob的电脑发送一串1和0比特值，表示电路上的高低电压。这些比特构成了一种信息，包括请求头和消息体。请求头包含当前Bob的本地路由器MAC地址和Alice的IP