Web Scraping
for Data Science
with Python

Seppe vanden Broucke and Bart Baesens

– Free Extract –

Seppe vanden Broucke, Naamsestraat 69 - box 3555, 3000 Leuven, Belgium
seppe.vandenbroucke@kuleuven.be

Chapter 1

Introduction

1.1 About this Book

1.1.1 Welcome

Congratulations! By picking up this book, you've set the first steps into the exciting world of web scraping. First of all, we want to thank you, the reader, for choosing this guide to accompany you on this journey.

For those who are not familiar with programming or the deeper workings of the web, web scraping often looks like a black art: the ability to write a program that sets off on its own to explore the Internet and collect data is seen as a magical, exciting, perhaps even scary power to possess. Indeed, there are not many programming tasks that are able to fascinate both experienced and novice programmers in quite such a way as web scraping. Seeing a program working for the first time as it reaches out on the web and starts gathering data never fails to provide a certain rush, feeling like you've circumvented the "normal way" of working and just cracked some sort of enigma. It is perhaps because of this reason that web scraping is also making a lot of headlines these days.

In this book, we set out to provide a concise and modern guide to web scraping, using Python as our programming language. We know that there are a lot of other books and online tutorials out there, but we felt that there was room for another entry. In particular, we wanted to provide a guide that is "short and sweet", without falling into the typical "learn this in X hours"-trap where important details or best practices are glossed over just for the sake of speed. In addition, you'll note that we have titled this book as "Web Scraping for Data Science". We're data scientists ourselves, and have very often found web scraping to be a powerful tool to have in your arsenal for the purpose of data gathering. Many data science projects start with the first step of obtaining an appropriate data set. In some cases (the “ideal situation”, if you will), a data set is readily provided by a business partner, your company’s data warehouse, or your academic supervisor, or can be bought or obtained in a structured format by external data providers, but many truly interesting projects start from collecting a treasure trove of information from the same place as humans do: the web. As such, we set out to offer something that:

- Is concise and to the point, whilst still being thorough.
- Is geared towards data scientists: we’ll show you how web scraping “plugs into” several parts of the data science workflow.
- Takes a code first approach to get you up to speed quickly without too much boilerplate text.
- Is modern by using well-established best practices and publicly available, open-source Python libraries only.
- Goes further than simple basics by showing how to handle the web of today, including JavaScript, cookies, and common web scraping mitigation techniques.
- Includes a thorough managerial and legal discussion regarding web scraping.
- Provides lots of pointers for further reading and learning.
- Includes many larger, fully worked out examples.

We hope you enjoy reading this book as much as we had writing it. Feel free to contact us in case you have questions, find mistakes, or just want to get in touch! We love hearing from our readers and are open to receive any thoughts and questions.

— Seppe vanden Broucke, *seppe.vandenbroucke@kuleuven.be*

— Bart Baesens, *bart.baesens@kuleuven.be*

1.1.2 Audience

We have written this book with a data science oriented audience in mind. As such, you’ll probably already be familiar with Python or some other programming language or analytical toolkit (be it R, SAS, SPSS, or something else). If you’re using Python already, you’ll feel right at home. If not, we include a quick Python primer later on in this chapter to catch up with the basics and provide pointers to other guides as well. Even if you’re not using Python yet for your daily data science tasks (many will argue that you should), we want to show you that Python is a particularly powerful language to use for scraping data from the web. We also assume that you have some basic knowledge regarding how the web works. That is, you know your way around a web browser and know what URLs are; we’ll explain the details in depth as we go along.

To summarize, we have written this book to be useful to the following target groups:

- Data science practitioners already using Python and wanting to learn how to scrape the web using this language.
- Data science practitioners using another programming language or toolkit, but want to adopt Python to perform the web scraping part of their pipeline.
- Lecturers and instructors of web scraping courses.
- Students working on a web scraping project or aiming to increase their Python skill set.
- "Citizen data scientists" with interesting ideas requiring data from the web.
- Data science or business intelligence managers wanting to get an overview of what web scraping is all about and how it can bring a benefit to their teams, and what the managerial and legal aspects are that need to be considered.

1.1.3 Structure

The chapters in this book can be divided into three parts:

- **Part 1: Web Scraping Basics (Chapters 1 to 3):** In these chapters, we'll introduce you to web scraping, why it is useful to data scientists, and discuss the key components of the web—HTTP, HTML and CSS. We'll show you how to write basic scrapers using Python, using the "requests" and "Beautiful Soup" libraries.
- **Part 2: Advanced Web Scraping (Chapters 4-6):** here, we delve deeper into HTTP and show you how to work with forms, login screens and cookies. We'll also explain how to deal with JavaScript-heavy websites and show you how to go from simple web scrapers to advanced web crawlers.
- **Part 3: Managerial Concerns and Best Practices (Chapters 7-9):** In this concluding part, we discuss managerial and legal concerns regarding web scraping in the context of data science, and also "open the door" to explore other tools and interesting libraries. We also list a general overview regarding web scraping best practices and tips. The final chapter includes some larger web scraping examples to show how all concepts covered before can be combined and highlights some interesting data science oriented use cases using web scraped data.

This book is set up to be very easy to read and work through. Newcomers are hence simply advised to read through this book from start to finish. That said, the book is structured in such a way that it should be easy to refer back to any part later on in case you want to brush up your knowledge or look up a particular concept.

1.1.4 About the Authors

Seppe vanden Broucke is an assistant professor of data and process science at the Faculty of Economics and Business, KU Leuven, Belgium. His research interests include business data mining and analytics, machine learning, process management, and process mining. His work has been published in well-known international journals and presented at top conferences. Seppe's teaching includes Advanced Analytics, Big Data and Information Management courses. He also frequently teaches for industry and business audiences. Besides work, Seppe enjoys travelling, reading (Murakami to Bukowski to Asimov), listening to music (Booka Shade to Miles Davis to Claude Debussy), watching movies and series (less so these days due to a lack of time), gaming, and keeping up with the news.

Bart Baesens is a professor of big data and analytics at KU Leuven, Belgium, and a lecturer at the University of Southampton, United Kingdom. He has done extensive research on big data and analytics, credit risk modeling, fraud detection and marketing analytics. Bart has written more than 200 scientific papers and several books. Besides enjoying time with his family, he is also a diehard Club Brugge soccer fan. Bart is a foodie and amateur cook. He loves drinking a good glass of wine (his favorites are white Viognier or red Cabernet Sauvignon) either in his wine cellar or when overlooking the authentic red English phone booth in his garden. Bart loves traveling and is fascinated by World War I and reads many books on the topic.

More information about the authors and their research can be found online at *www.dataminingapps.com*. The companion website for this book can be found at *www.webscrapingfordatascience.com*, where you'll find more information, an errata list, and where we host the examples used throughout this book.

1.2 What is Web Scraping?

Web “scraping” (also called “web harvesting”, “web data extraction” or even “web data mining”), can be defined as “the construction of an agent to download, parse, and organize data from the web in an automated manner”. Or, in other words: instead of a human end-user clicking away in a web browser and copy-pasting interesting parts into, say, a spreadsheet, web scraping offloads this task to a computer program which can execute it much faster, and more correctly, than a human can.

The automated gathering of data from the Internet is probably as old as the Internet itself, and the term “scraping” has been around for much longer than the web. Before “web scraping” became popularized as a term, a practice known as “screen scraping” was already well-established as a way to extract data from a visual representation—which in the early days of computing (think 1960s-80s) often boiled down to simple, text based “terminals”. Just as today, people in those days were also interested in “scraping” large amounts of text from such terminals and store this data for later use.

1.2.1 Why Web Scraping for Data Science?

When surfing the web using a normal web browser, you’ve probably encountered multiple sites where you considered the possibility of gathering, storing, and analyzing the data presented on the site’s pages. Especially for data scientists, whose “raw material” is data, the web exposes a lot of interesting opportunities:

- There might be an interesting table on a Wikipedia page (or pages) you want to retrieve to perform some statistical analysis.
- Perhaps you want to get a list of reviews from a movie site to perform text mining, create a recommendation engine or build a predictive model to spot fake reviews.
- You might wish to get a listing of properties on a real-estate site to build an appealing geo-visualization.
- You’d like to gather additional features to enrich your data set based on information found on the web, say, weather information to forecast e.g. soft drink sales.
- You might be wondering about doing social network analytics using profile data found on a web forum.
- It might be interesting to monitor a news site for trending new stories on a particular topic of interest.

The web contains lots of interesting data sources that provide a treasure trove for all sorts of interesting things. Sadly, the current unstructured nature of the web does not always make it easy to gather or export this data in an easy manner. Web browsers are very good at showing images, displaying animations, and laying out websites in a way that is visually appealing to humans, but they do not expose a simple way to export their data, at least not in most cases. Instead of viewing the web page by page through your web browser's window, wouldn't it be nice to be able to automatically gather a rich data set? This is exactly where web scraping enters the picture.

If you know your way around the web a bit, you'll probably be wondering: "isn't this exactly what Application Programming Interface (APIs) are for?" Indeed, many websites nowadays provide such an API which provides a means for the outside world to access their data repository in a structured way—meant to be consumed and accessed by computer programs, not humans (although the programs are written by humans, of course). Twitter, Facebook, LinkedIn, and Google, for instance, all provide such APIs in order to search and post tweets, get a list of your friends and their likes, see who you're connected with, and so on. So why, then, would we still need web scraping? The point is that APIs are great means to access data sources, provided the website at hand provides one and to begin with and that the API exposes the functionality you want. The general rule of thumb is to look for an API first and use that if you can, before setting off to build a web scraper to gather the data. For instance, you can easily use Twitter's API to get a list of recent tweets, instead of re-inventing the wheel yourself. Nevertheless, there are still various reasons why web scraping might be preferable over the use of an API:

- The website you want to extract data from does not provide an API.
- The API provided is not free (whereas the website is).
- The API provided is rate limited: meaning you can only access it a certain times per second, per day, ...
- The API does not expose all the data you wish to obtain (whereas the website does).

In all of these cases, the usage of web scraping might come in handy. The fact remains that if you can view some data in your web browser, you will be able to access and retrieve it through a program. If you can access it through a program, the data can be stored, cleaned, and used in any way.

1.2.2 Who is Using Web Scraping?

There are many practical applications of having access to and gathering data on the web, many of which fall in the realm of data science. The following list outlines some interesting real-life use cases:

- Many of Google's products have benefited from Google's core business of crawling the web. Google Translate, for instance, utilizes text stored on the web to train and improve itself.

- Scraping is being applied a lot in HR and employee analytics. The San Francisco based hiQ startup specializes in selling employee analyses by collecting and examining public profile information, for instance from LinkedIn (who was not happy about this but was so far unable to prevent this practice following a court case, see *https://www.bloomberg.com/news/features/2017-11-15/the-brutal-fight-to-mine-your-data-and-sell-it-to-your-boss*).
- Digital marketeers and digital artists often use data from the web for all sorts of interesting and creative projects. "We Feel Fine" by Jonathan Harris and Sep Kamvar, for instance, scraped various blog sites for phrases starting with "I feel", the results of which could then visualize how the world was feeling throughout the day.
- In another study, messages scraped from Twitter, blogs and other social media were scraped to construct a data set which was used to build a predictive model towards identifying patterns of depression and suicidal thoughts. This might be an invaluable tool for aid providers, though of course warrants a thorough consideration of privacy related issues as well (see *https://www.sas.com/en_ca/insights/articles/analytics/using-big-data-to-predict-suicide-risk-canada.html*).
- In a paper titled "The Billion Prices Project: Using Online Prices for Measurement and Research" (see *http://www.nber.org/papers/w22111*), web scraping was used to collect a data set of online price information which was used to construct a robust daily price index for multiple countries.
- Banks and other financial institutions are using web scraping for competitor analysis. For example, banks frequently scrape competitor's sites to get an idea of where branches are being opened or closed, or to track loan rates offered—all of which is interesting information which can be incorporated in their internal models and forecasting. Investment firms also often use web scraping, for instance to keep track of news articles regarding assets in their portfolio.
- Sociopolitical scientists are scraping social websites to track population sentiment and political orientation. A famous article called "Dissecting Trump's Most Rabid Online Following" (see *https://fivethirtyeight.com/features/dissecting-trumps-most-rabid-online-following/*) analyzes user discussions on reddit using semantic analysis to characterize the online followers and fans of Donald Trump.
- One researcher was able to train a deep learning model based on scraped images from Tinder and Instagram together with their "likes" to predict whether an image would be deemed "attractive" (see *http://karpathy.github.io/2015/10/25/selfie/*). Smartphone makers are already incorporating such models in their photo apps to help you brush up your pictures.
- In "The Girl with the Brick Earring", Lucas Woltmann sets out to scrape Lego brick

information from *https://www.bricklink.com* to determine the best selection of Lego pieces to represent an image (see *http://lucaswoltmann.de/art'n'images/2017/04/08/the-girl-with-the-brick-earring.html*).

- Lyst, a London based online fashion marketplace, scraped the web for semi-structured information about fashion products and then applied machine learning to present this information cleanly and elegantly for consumers from one central website. Other data scientists have done similar projects to cluster similar fashion products (see *http://talks.lystit.com/dsl-scraping-presentation/*).
- We've supervised a study where web scraping was used to extract information from job sites, to get an idea regarding the popularity of different data science and analytics related tools in the workplace (spoiler: Python and R were both rising steadily).
- Another study from our research group involved using web scraping to monitor news outlets and web forums to track public sentiment regarding Bitcoin.

No matter your field of interest, there's almost always a use case to improve or enrich your practice based on data. "Data is the new oil", so the common saying goes, and the web has a lot of it.

Web Scraping for Data Science with Python

Seppe vanden Broucke and Bart Baesens

– End of Extract –