FREE eBook

LEARNING web-scraping

Free unaffiliated eBook created from **Stack Overflow contributors.**

#web-scraping

Table of Contents

About 1

Chapter 1: Getting started with web-scraping 2

Remarks 2

Examples 2

Web Scraping in Python (using BeautifulSoup) 2

Credits 4

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: web-scraping

It is an unofficial and free web-scraping ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official web-scraping.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with web-scraping

Remarks

This section provides an overview of what web-scraping is, and why a developer might want to use it.

It should also mention any large subjects within web-scraping, and link out to the related topics. Since the Documentation for web-scraping is new, you may need to create initial versions of those related topics.

Examples

Web Scraping in Python (using BeautifulSoup)

When performing data science tasks, it’s common to want to use data found on the internet. You’ll usually be able to access this data via an Application Programming Interface(API) or in other formats. However, there are times when the data you want can only be accessed as part of a web page. In cases like this, a technique called web scraping comes into picture.
To apply this technique to get data from web-pages, we need to have basic knowledge about web-page structure and tags used in web-page development(i.e, `<html>` ,`<li>`,`<div>` etc.,). If you are new to web development you can learn it here.

So to start with web scrapping, we'll use a simple website. We'll use `requests` module to get the web-page content OR source code.

```
import requests
page = requests.get("http://dataquestio.github.io/web-scraping-pages/simple.html")
print (page.content) ## shows the source code
```

Now we'll use bs4 module to scrap the content to get the useful data.

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(page.content, 'html.parser')
print(soup.prettify()) ##shows source in html format
```

You can find the required tags using `inspect element` tool in your browser.Now let's say you want to get all the data that is stored with `<li>` tag.Then you can find it with the script

```
soup.find_all('li')
# you can also find all the list items with class='ABC'
# soup.find_all('p', class_='ABC')
# OR all elements with class='ABC'
# soup.find_all(class_="ABC")
# OR all the elements with class='ABC'
# soup.find_all(id="XYZ")
```

Then you can get the text in the tag using

```
for i in range(len(soup.find_all('li'))):
    print (soup.find_all('li')[i].get_text())
```

The whole script is small and pretty simple.

```
import requests
from bs4 import BeautifulSoup

page = requests.get("http://dataquestio.github.io/web-scraping-pages/simple.html") #get the
page
soup = BeautifulSoup(page.content, 'html.parser') # parse according to html
soup.find_all('li') #find required tags

for i in range(len(soup.find_all('li'))):
    print (soup.find_all('li')[i].get_text())
```

Read Getting started with web-scraping online: https://riptutorial.com/web-scraping/topic/7746/getting-started-with-web-scraping

Credits

S. No	Chapters	Contributors
1	Getting started with web-scraping	Community, thepurpleowl