Python – Εισαγωγή στο web scraping

Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών

Γιάννης Γαροφαλάκης - Σπύρος Σιούτας

Γ. Γαροφαλάκης, Σ. Σιούτας, Π. Χατζηδούκας

Web scaping - api

- Το web scraping είναι μια τεχνική για τη συλλογή δεδομένων ή πληροφοριών σε ιστοσελίδες.
- Είναι μια μέθοδος εξαγωγής δεδομένων από έναν ιστότοπο που δεν διαθέτει API ή όταν θέλουμε να εξάγουμε ΠΟΛΛΑ δεδομένα, κάτι που δεν μπορούμε να κάνουμε μέσω ενός API λόγω περιορισμού ταχύτητας.
- Μπορούμε να κάνουμε εξαγωγή οποιωνδήποτε δεδομένων θέλουμε κατά την περιήγηση στο διαδίκτυο

ΠΑΡΑΔΕΙΓΜΑΤΑ ΧΡΗΣΗΣ WEB SCRAPING

- Εξαγωγή πληροφοριών προϊόντος
- Εξαγωγή αγγελιών εργασίας και πρακτικής άσκησης
- Εξαγωγή προσφορών και εκπτώσεων από ιστότοπους με προσφορές της ημέρας
- Εξαγωγή δεδομένων για τη δημιουργία μηχανής αναζήτησης
- Συλλογή δεδομένων καιρού

WEB SCRAPING ή χρήση API

- Η συλλογή δεδομένων από ιστοσελίδες δεν υπόκειται σε περιορισμούς ταχύτητας
- Ανώνυμη πρόσβαση στον ιστότοπο και συλλογή δεδομένων
- Ορισμένοι ιστότοποι δεν διαθέτουν API
- Ορισμένα δεδομένα δεν είναι προσβάσιμα μέσω API

Ροη εργασιών WEB SCRAPING

- Λήψη στοιχείωντου ιστότοπου - χρησιμοποιώντας τη βιβλιοθήκη HTTP
- Ανάλυση του εγγράφου html - χρησιμοποιώντας οποιαδήποτε βιβλιοθήκη ανάλυσης
- Αποθήκευση των αποτελεσμάτων - είτε σε βάση δεδομένων, csv, αρχείο κειμένου κ.α.

LIBRARIES

- **BeautifulSoup**
- lxml
- selenium
- re
- scrapy

HTTP LIBRARIES

- Requests

r = requests.get('https://www.google.com').html

requests.get(url) στέλνει ένα **GET** αίτημα στη διεύθυνση URL.

- urllib

html = urllib.request.urlopen('http://python.org/').read()()

Η urlopen() ανοίγει τον σύνδεσμο και read() διαβάζει τα bytes της απάντησης.

- httplib/httplib2

h = httplib2.Http(".cache")

(resp_headers, content) = h.request("http://pydelhi.org/", "GET")

Η httplib2 είναι εξωτερική βιβλιοθήκη που υποστηρίζει caching, authentication, redirects, κ.λπ.

- Http(".cache"): Δημιουργεί HTTP client που αποθηκεύει απαντήσεις σε τοπικό cache directory.
- h.request(...) επιστρέφει:

--resp_headers: Λεξικό με HTTP headers απόκρισης.

--content: Το σώμα της απάντησης (συνήθως HTML ή JSON).

PARSING LIBRARIES

- **BeautifulSoup (bs4)**

```
# Import the BeautifulSoup class from the bs4 module
from bs4 import BeautifulSoup
# Parse the HTML document using BeautifulSoup (defaults to 'html.parser')
tree = BeautifulSoup(html_doc)
# Access and return the <title> element of the HTML document
tree.title
```

- **lxml**

```
# Import lxml's HTML parsing module
import lxml.html

# Parse the HTML document into lxml tree structure
tree = lxml.html.fromstring(html_doc)
# Use XPath to extract the text inside the <title> tag
title = tree.xpath('/title/text()')
```

- **re**

```
# Import the regular expressions module
import re
# Use a regular expression to search for content inside the <title> tag
title = re.findall('<title>(.*?)</title>', html_doc)
```

BEAUTIFULSOUP

- μπορούμε να το μάθουμε γρήγορα
- πολύ εύκολο στη χρήση
- αποκλειστικά σε Python
- πολύ αργό

LXML

- πολύ γρήγορο
- όχι αποκλειστικά σε Python
- το lxml λειτουργεί με όλες τις εκδόσεις python από 2.x έως 3.x

RE

- Αναγκαστική γνώση των συμβόλων του
- μπορεί να γίνει περίπλοκο
- αποκλειστικά σε Python
- μέρος της τυπικής βιβλιοθήκης
- πολύ γρήγορο
- για κάθε έκδοση Python

Αποτέλεσμα

- Desktop$ python test.py
- bs_test took 1851.457 ms
- lxml_test took 232.942 ms
- regex_test took 7.186 ms

- lxml took 32x more time than re,
BeautifulSoup took 245x! more time than re

Είναι νόμιμο το web scraping;

- Ασφαλώς, η ενέργεια του web scraping δεν είναι παράνομη.

- Ωστόσο, πρέπει να ακολουθούνται ορισμένοι κανόνες.

- Το web scraping μπορεί να γίνει παράνομο όταν εξάγονται δεδομένα που δεν είναι δημόσια διαθέσιμα.