Web Scraping – Ochrana

Pokročilí 28

Programovanie v Pythone • Stredná škola

Dáta na webe a ich hodnota

Dáta sú v dnešnom svete jednou z najcennejších komodít. Firmy zbierajú informácie o používateľoch, zákazníkoch, produktoch či trendoch. Tieto dáta im umožňujú lepšie cieliť reklamu, zlepšovať produkty alebo predávať prístup k nim.

Preto mnohé stránky zavádzajú rôzne formy ochrany, aby zabránili automatizovanému sťahovaniu dát.

Typické ochranné mechanizmy:

- Paywall – obsah je dostupný až po zaplatení
- Login wall – vyžaduje prihlásenie
- Rate limiting – obmedzenie počtu požiadaviek
- robots.txt – pravidlá pre automatizované roboty

Spôsoby ochrany obsahu na webe

Paywall
Pristup k obsahu je možný až po zaplatení (napr. New York Times, HNonline, vedecké časopisy).

Login-required
Obsah je dostupný len pre prihlásených používateľov.

Rate limiting
Server obmedzuje počet požiadaviek z jednej IP adresy za určitý čas.

robots.txt
Súbor, ktorý hovorí slušným robotom, kam môžu a kam nesmú chodiť.

Čo je Web Scraping?

Web scraping je automatizované sťahovanie a spracovanie dát z webových stránok pomocou programu (bota).

Rozlišujeme dva súvisiace pojmy:

Web crawling
Automatické prechádzanie webu a objavovanie nových URL adries (napr. vyhľadávacie roboty Google).

Web scraping
Parsovanie stiahnutých stránok a extrakcia konkrétnych informácií (názvy produktov, ceny, články...).

Ako servery bránia scrapingu?

Servery sa snažia rozpoznať a obmedziť automatizovaných botov, pretože môžu výrazne zaťažiť server.

Rate limiting
Obmedzenie počtu požiadaviek z jednej IP adresy za minútu/hodinu.

Browser fingerprinting
Server zisťuje, či ide o bežný prehliadač (JavaScript, cookies, hlavičky...).

Honeypoty
Skryté odkazy, ktoré bežný používateľ nevidí, ale bot ich môže navštíviť.

robots.txt
Slušné boty by mali tento súbor rešpektovať.

Súbor robots.txt

Súbor `robots.txt` je umiestnený v koreňovom adresári webu a obsahuje pravidlá pre automatizovaných robotov.

Príklad obsahu robots.txt:

```
User-agent: *
Disallow: /admin/
Disallow: /private/
Allow: /public/

Sitemap: https://example.com/sitemap.xml
```

Browser Fingerprinting

Server môže zistiť veľa informácií o návštevníkovi aj bez cookies:

- HTTP hlavičky (User-Agent, Accept-Language...)
- Podpora JavaScriptu a cookies
- Rozlíšenie obrazovky a nainštalované fonty
- Časová zóna a jazyk prehliadača
- WebGL a Canvas fingerprint

Ak sa tieto hodnoty veľmi líšia od bežného používateľa, server môže podozrievať bota.

Etika a zákonnosť scrapingu

Web scraping nie je automaticky nelegálny, ale môže byť problematický:

- Môže preťažiť server (podobne ako DoS útok)
- Porušuje podmienky používania stránky (Terms of Service)
- Môže porušovať autorské práva alebo GDPR
- Niektoré stránky poskytujú oficiálne API – to je najslušnejší spôsob získavania dát

Vždy rešpektuj robots.txt a nezaťažuj server zbytočne.

Nástroje na scraping v Pythone

Na tvorbu scraperov v Pythone sa najčastejšie používajú tieto knižnice:

requests
Jednoduché odosielanie HTTP požiadaviek (GET, POST).

BeautifulSoup (bs4)
Parsovanie a vyhľadávanie v HTML kóde.

Crawlee
Moderný framework na crawling a scraping (odporúčaný na cvičeniach).

Zhrnutie

- Dáta na webe sú cenné, preto ich stránky chránia rôznymi spôsobmi.
- Web scraping je mocný nástroj, ale môže zaťažovať servery.
- Servery používajú rate limiting, fingerprinting a robots.txt na obranu.
- Slušný bot rešpektuje pravidlá (robots.txt, rate limits).
- Vždy zváž etické a právne dôsledky scrapingu.
- Na scraping v Pythone sa používa requests + BeautifulSoup alebo Crawlee.