

Biddit webscraper:

Stap 1: Importeer de vereiste bibliotheken

De Scraper importeert verschillende bibliotheken die nodig zijn voor het uitvoeren van de scraping- en onderhoudsactiviteiten. Bij problemen kijk eerst na of deze correct zijn geïnstalleerd in uw programmeer omgeving!

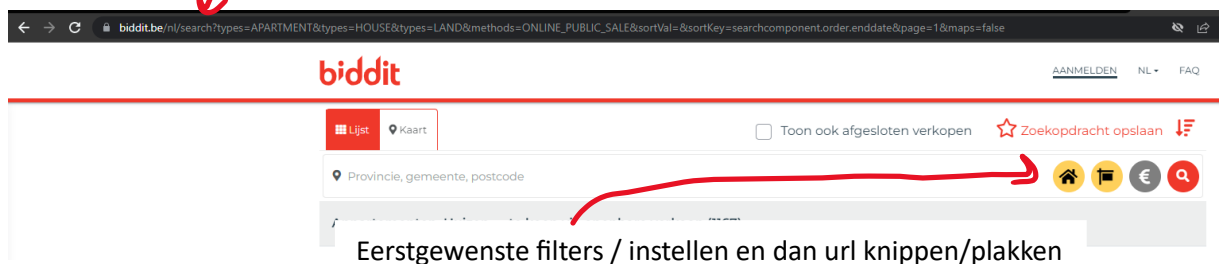
Stap 2: TOMTOM API instellingen

De TOMTOM API instellingen in de code zijn nodig voor geocodering. Controleer of de TOMTOM API-sleutel (TOMTOM_API_KEY) correct is en overeenkomt met de gewenste functionaliteit > kijk ook na of het dagelijkse limiet niet is overschreden.

Stap 3: Biddit URL's

De Biddit URL's in de code zijn de basis- en start-URL's die worden gebruikt om de scraping uit te voeren. Er is een basis-URL (BASE_URL) en een startpagina (START_PAGE) die kunnen worden aangepast aan je behoeften. Controleer of de URL's overeenkomen met de gewenste scraping-doelen.

```
# Biddit URL's
BIDDIT_URL = "https://www.biddit.be"
BASE_URL = "https://www.biddit.be/nl/search?types=APARTMENT&types=HOUSE&types=LAND&methods=ONLINE_PUBLIC_SALE&nonActive=true"
START_PAGE = 1
```



Eerstgewenste filters / instellen en dan url knippen/plakken

Stap 4: Vastgoed Mapping

De vastgoed_omgezet dictionary in de code wordt gebruikt om het originele vastgoedtype om te zetten naar een gestandaardiseerd type. Als je wijzigingen wilt aanbrengen in de mapping, pas dan de waarden aan in de dictionary volgens je vereisten.

Stap 5: Configuratie van de browser

De functie `create_headless_browser()` creëert een nieuw exemplaar van de Chrome-browser in de headless-modus. Als je de browser onzichtbaar wilt maken tijdens het uitvoeren van de code, kun je de opmerkingstekens (#) verwijderen bij de regels `#chrome_options.add_argument("--headless")`

Stap 6: Sluiten van het cookiebanner

De functie `close_cookie_banner(browser)` wordt gebruikt om het cookiebanner op de website te sluiten.

Stap 7: Scraping van de detailpagina's

De functie `scrape_page(browser, url, data)` is verantwoordelijk voor het scrapen van de gegevens van een enkele detailpagina. (dus elke listing)

Stap 8: Scrapen van de Biddit-gegevens

De functie `scrape_biddit_data()` voert de hoofdscraping uit van de Biddit-website.

Stap 9: Verwerken van de detail-URL's

De functie `process_detail_urls(data, detail_urls, filename)` wordt gebruikt om de lijst met detail-URL's te verwerken en de scraping op elke pagina uit te voeren.

Stap 10: Initialiseren van de DataFrame

De functie `create_initial_dataframe(filename)` initialiseert een lege DataFrame of laadt een bestaande DataFrame uit een CSV-bestand. De bestandsnaam kan worden gewijzigd naar de gewenste naam voor de gegevensopslag. Als er iets aan de data bestanden veranderd moet je dit hier aanpassen.

Stap 11: Wijzigen van het vastgoedtype

De functie `wijzig_vastgoed_type(type_vastgoed)` wordt gebruikt om het originele type vastgoed om te zetten naar het nieuwe type volgens de `vastgoed_omgezet` mapping. Als je de mapping wilt aanpassen, zie stap 4

Stap 12: Hoofdfunctie en uitvoering

De functie `main()` is de hoofdfunctie die wordt uitgevoerd wanneer het script wordt gestart.

Onderhoudsinstructies:

Om de HTML-elementen te vinden en aan te passen in het geval de website van stijl verandert, moet je mogelijk de HTML-structuur en de CSS-klassen van de elementen inspecteren. Dit kan gedaan worden met behulp van ontwikkelingstools in de webbrowser, zoals de inspectietool in Google Chrome (F12).

vb.:

```
# Zoek het type vastgoed
try:
    type_vastgoed = soup.find("span", {"class": "property-type me-2"}).text.strip()
```

The screenshot shows a web browser displaying a real estate listing on the website biddit.nl. The listing is for a house with 1 room, located at 3120 Tremelo, Looisestraat 46. The price is 240,000. The property type is 'Huis'. The developer tools are open, showing the HTML structure of the page. Red arrows indicate the mapping between the BeautifulSoup code and the HTML elements. The code searches for a 'span' with the class 'property-type me-2'. In the browser, this class is used for the property type 'Huis' and the price '240000'. The developer tools show the corresponding HTML structure.

Om HTML-elementen te vinden via de code, kun je gebruikmaken van methoden zoals `find()`, `find_all()`, `select_one()` en `select()` van BeautifulSoup. Je moet de juiste HTML-selectoren en attributen gebruiken om de gewenste elementen te identificeren. Zie screenshot hierboven.

Als de website volledig van stijl verandert en de HTML-structuur wordt gewijzigd, moet je de relevante delen van de code bijwerken om de juiste elementen te vinden en de scraping-functionaliteit te behouden. Dit kan betekenen dat je de HTML-selectoren moet wijzigen, nieuwe selectoren moet toevoegen of de logica voor het vinden van elementen moet aanpassen.