SCRAPER POBIERAJĄCY PRZEPISY KULINARNE W JĘZYKU PYTHON

Projekt z przedmiotu Automaty, Języki i Obliczenia

Maciej Kozak

Spis treści

Wstęp

Temat mojej pracy inżynierskiej to ,,Aplikacja wspomagająca przygotowanie posiłków”. Aby aplikacja spełniała oczekiwania użytkownika konieczna jest obszerna baza danych, w której znajdują się opisy jak przygotować daną potrawę oraz potrzebne składniki. Wpadłem na pomysł, aby jako projekt z przedmiotu Automaty, Języki i Obliczenia napisać scraper, który umożliwi mi pobieranie interesujących mnie informacji ze strony kulinarnej. Dzięki niemu zaoszczędzę wiele czasu, który straciłbym, gdybym musiał wprowadzać dane do mojej bazy ręcznie. Zdecydowałem się na pobieranie przepisów ze strony kuchnialidla.pl. Znajduje się tam ogromna ilość przepisów oraz mam pewność, że przepisy są sprawdzone. Dzięki takiemu scraperowi wzbogaciłem swoją aplikację o wiele nowych przepisów.

Cel projektu

Celem projektu jest napisanie scrapera w języku python, który na początku wyodrębni ze strony kuchnialidla.pl tytuły, opisy przygotowania oraz listy składników przepisów, a następnie pozyskane informacje zapisze do mojej bazy danych postgreSQL. Dzięki temu przepisy ze strony będą wyświetlały się w mojej aplikacji.

Technologie

- Scraper napisałem w języku Python w wersji 3.10.0, a w nim posłużyłem się głównie biblioteką BeautifulSoup oraz Requests

-PostgreSQL

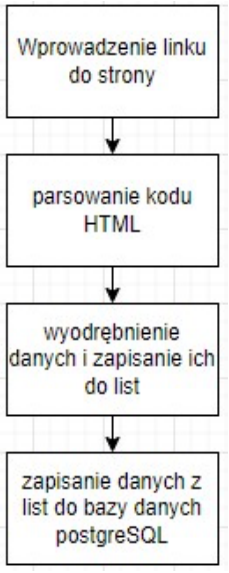
Jak działa web scraping?

Web scraping oznacza wyodrębnienie danych ze stron internetowych. Aby dostać się do pożądanych przez nas informacji należy napisać program, który wysyła żądanie do serwera, na którym znajduje się podana przez nas strona internetowa. Scraper pobiera kod źródłowy strony, ale nie wyświetla jej tylko ją filtruje aby wyodrębnić z niej zawartość, którą mu poleciliśmy. Przed przystąpieniem do takiego działania powinno się sprawdzić czy strona zezwala na pobieranie z niej informacji. Można to zrobić poprzez sprawdzenie pliku robots.txt lub przejrzenie warunków korzystania z usługi czy portalu. Należy również pamiętać o RODO, jeśli chcielibyśmy pobierać dane osobowe.

Realizacja projektu

Organizacja działania

- wprowadzenie w programie linku do strony, z której pobierane będą przepisy (kuchnialidla.pl)
- parsowanie kodu HTML podanej strony
- wyodrębnienie takich informacji jak tytuł, opis przygotowania oraz lista składników, a następnie zapisanie ich do list
- zapisanie danych z list do bazy danych postgreSQL



Wejście programu

Jako wejście programu podaję link do strony kuchnialidla.pl oraz liczbę stron, którą chcę aby mój program przeanalizował i wydobył z nich dane.

```
baseURL = f"https://kuchnialidla.pl/przepisy/dania-glowne/{x+1}#lista"
```

Wyjście programu

Wyjściem programu jest lista zawierająca tytuły, opisy przygotowania oraz lista zawierająca składniki, które następnie zapisywane są do bazy danych postgreSQL.

```
[1, 'Pasta z sosem tuńczykowym', '\n\nPrzygotuj:\n\nrobota kuchennego Monsieur Cuisine Smart\n\nPrzygotuj wcześniej:\nŚmietanę włóż do lodówki, aby w momenc
ie użycia była zimna.\nSposób przygotowania:\nKROK 1: ROZDRABNIANIE PARMEZANU I PIETRUSZKI\nParmezan w kawałkach włożyć do pojemnika miksującego i rozdrabni
ać (z zatkniętą miarką) 10 sekund/prędkość 10. Następnie przełożyć go do miseczki.\nNatkę pietruszki opłukać, otrząsnąć z wody, oberwać listki z gałązek i r
ozdrabniać (z zatkniętą miarką) 8 sekund/prędkość 6. Następnie przełożyć do innego naczynia. Umyć pojemnik miksujący.\nKROK 2: PRZYGOTOWANIE BAZY POD SOS\nC
ebulę obrać i przekroić na pół, ząbek czosnku obrać i oba składniki rozdrabniać w pojemniku miksującym (z zatkniętą miarką), przytrzymując przycisk turbo/1
sekundę. Kawałeczki cebuli i czosnku zgarnąć łopatką ze ścianek na dno pojemnika.\nOdsączone i opłukane anchois i kawałki tuńczyka włożyć do pojemnika miksu
jącego i rozdrabniać (z zatkniętą miarką), przytrzymując przycisk turbo/1 sekundę. Kawałeczki zgarnąć łopatką ze ścianek na dno pojemnika. Dodać odsączone k
apary i oliwę z oliwek, po czym przypiekać (bez miarki), ustawiwszy program przypiekanie/2 minuty/100°C.\nKROK 3: GOTOWANIE SOSU I MAKARONU\nPenne gotować w
edług wskazówek na opakowaniu.\nZimną śmietanę, krojone pomidory, oregano, sok z cytryny, ½ łyżeczki soli oraz ½ łyżeczki pieprzu i (z zatkniętą miarką) got
ować wszystko, ustawiwszy program mieszanie/10 minut/prędkość 1/100°C. Doprawić całość solą i pieprzem, po czym mieszać (z zatkniętą miarką), ustawiwszy pro
gram mieszanie/20 sekund/prędkość 1.\nKROK 4: PODANIE\nOdcedzić penne i podać z sosem z tuńczyka. Posypać pietruszką i parmezanem.\n\n\n\n', 'lidl-logo.png'
]
```

```
[1, 'cebula – 1 szt. (80 g)', 'czosnek – 1 ząbek', 'anchois z puszki – 2 szt.', 'tuńczyk w oliwie – 1 puszka (185 g)', 'kapary – 1 łyżka', 'śmietany (30% tł
.) – 100 ml', 'krojone pomidory z puszki – 400 g', 'suszone oregano – 1 łyżeczka', 'sok z cytryny – 1 łyżka', 'sól – ½ łyżeczki']
```

Biblioteki

- **Requests** to biblioteka do wykonywania żądań dostępu do strony internetowej, bardzo ułatwia komunikację HTTP
- **BeautifulSoup** to biblioteka służąca do parsowania kodu HTML i XML
- **Pandas** to biblioteka odpowiadająca za analizowanie, modyfikowanie, wczytywanie i czyszczenie danych

Analiza kodu

```
def extractLinks(link):
    page = requests.get(link)
    soup = BeautifulSoup(page.content,"html.parser")
    names = []
    for a in soup.find_all('a', class_='description', href=True):
        names.append(a['href'])
    return names
```

Funkcja extractLinks() odpowiedzialna jest za wydzielenie linków do poszczególnych przepisów.

```
def extractData(tableName):
    dao.truncateTable(tableName)
    pageNumber = 2
    lista = []
    componentsList = []
    listaSkladnikow = []
    for x in range(pageNumber):
        baseURL = f"https://kuchnialidla.pl/przepisy/dania-glowne/{x+1}#lista"
        lista.extend(extractLinks(baseURL))
    id = 0
    for x in lista:
        URL = f"https://kuchnialidla.pl{x}"
        page = requests.get(URL)
        soup = BeautifulSoup(page.content,"lxml")
        title = soup.find('h1')
        components = soup.find('div', class_='skladniki')
        componentsList.append(components)
        description = soup.find(id="opis")
        id += 1
        image = 'lidl-logo.png'
        recipe = []
        recipe.append(id)
        recipe.append(title.text)
        recipe.append(description.text)
        recipe.append(image)
        print("--------------------------------")
        #print(recipe)
        dao.insertData(recipe, tableName)
    idd=0
    for ul in componentsList:
        idd+=1
        listaSkladnikow.append(idd)
        for li in ul.findAll('li')[:10]:
            listaSkladnikow.append(li.text)
        for i in range(1,11):
            if len(listaSkladnikow)<11:
                listaSkladnikow.append(' ')
        print("--------------------------------")
        #print(listaSkladnikow)
        dao.insertData2(listaSkladnikow, tableName)
        listaSkladnikow = []

    print("Job is finished...")
```

W funkcji extractData() wydobywam interesujące mnie informacje oraz zapisuje je do poszczególnych list, które następnie dodaje do mojej bazy danych.

Aby dodawanie do bazy danych było możliwe utworzyłem plik dao.py, w którym na początku w funkcji getConnection() tworze połączenie z moją bazą danych znajdującą się na Heroku:

```
import psycopg2
def getConnection():
    conn = psycopg2.connect(
            host = "[redacted]",
            database = "[redacted]",
            user = "[redacted]",
            password = "[redacted]"
        )
    return conn
```

A następnie w funkcjach insertData() oraz insertData2() przekazuje wartości z list do mojej bazy danych.

```
def insertData(recipe, tableName):

    conn = getConnection()
    cur = conn.cursor()
    insertSql = f"""
                INSERT INTO recipes
                (id_recipe, title, description, image)
                values ('{recipe[0]}', '{recipe[1]}', '{recipe[2]}', '{recipe[3]}')
                """

    cur.execute(insertSql)
    conn.commit()
    cur.close()
    conn.close()

def insertData2(listaSkladnikow, tableName):

    conn = getConnection()
    cur = conn.cursor()
    insertSql = f"""
                UPDATE recipes
                SET skladnik1 = '{listaSkladnikow[1]}',
                    skladnik2 = '{listaSkladnikow[2]}',
                    skladnik3 = '{listaSkladnikow[3]}',
                    skladnik4 = '{listaSkladnikow[4]}',
                    skladnik5 = '{listaSkladnikow[5]}',
                    skladnik6 = '{listaSkladnikow[6]}',
                    skladnik7 = '{listaSkladnikow[7]}',
                    skladnik8 = '{listaSkladnikow[8]}',
                    skladnik9 = '{listaSkladnikow[9]}',
                    skladnik10 = '{listaSkladnikow[10]}'
                WHERE id_recipe = {listaSkladnikow[0]};
                """

    cur.execute(insertSql)
    conn.commit()
    cur.close()
    conn.close()
```

Po wykonaniu programu tak prezentuje się część mojej bazy danych:

title	description
Pierogi z mięsem	⏎⏎Przygotuj:⏎⏎foremkę do wycinania krążków o ś…
Tacos w stylu meksykańskim z wołowiną i salsą	⏎⏎SPOSÓB PRZYGOTOWANIA:⏎SMAŻONE WARZYWA⏎KROK 1…
Zapiekanka z indykiem i ziemniaczanym purée	⏎⏎Przygotuj wcześniej:⏎Ziemniaki obierz i ugot…
Kluski francuskie z sosem szpinakowym	⏎⏎Przygotuj:⏎⏎robot kuchenny⏎⏎Przygotuj wcześn…
Curry rybne z mango i kolendrą	⏎⏎PRZYGOTUJ:⏎⏎robot Monsieur Cuisine⏎⏎SPOSÓB P…
Pasta z sosem tuńczykowym	⏎⏎Przygotuj:⏎⏎robota kuchennego Monsieur Cuisi…

skladnik1	skladnik2	skladnik3
olej roślinny - 1 łyżka	mąka (typ 405) - 420 g	sól - ½ łyżeczki
czerwona papryka - 1NBSPszt.	czerwona cebula - 1/2 szt.	czosnek - 1 ząbek
mączyste ziemniaki - 500 g	słodka śmietanka - 200 g	masło - 100 g
masło - 40 g	jajka - 2 szt.	mąka pszenna - 4 łyżki
cebula - 1 szt. (200 g)	czosnek - 2 ząbki	świeży imbir - kawałek ok 2 cm
cebula - 1 szt. (80 g)	czosnek - 1 ząbek	anchois z puszki - 2 szt.

Propozycja rozbudowy projektu

W zależności od tego czego potrzebuje użytkownik projekt można rozszerzyć o pobieranie dodatkowych informacji. W takim przypadku za pomocą biblioteki BeautifulSoup można wydobyć inne fragmenty strony oraz zapisać je do bazy za pomocą odpowiedniego polecenia PostgreSQL.

Bibliografia

https://miroslawmamczur.pl/web-scraping-co-to-i-jakie-sa-dobre-praktyki/

https://kamil.kwapisz.pl/web-scraping-python/

https://miroslawmamczur.pl/beautifulsoup/