

Wiktoria Wnuk 277993

Nataniel
Piekarski 278692

WEBSCRAPING

DO

TRÓJMIASTA . P

L

WEBSCRAPING

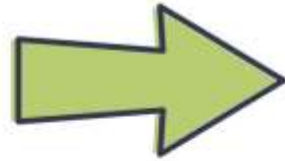
- Technika ekstrakcji danych z witryn internetowych
- Wykorzystywany w gromadzeniu dużej ilości danych z strony internetowych
- Może być użyty do monitorowania cen, śledzenia wiadomości i trendów
- Przykładowe sposoby webscrapingu: manualny, zautomatyzowany



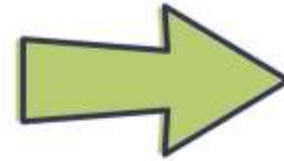
WEB SCRAPING



HTML WEBSITES



WEB SCRAPING



DATA

PROCES WEBSCRAPINGU



KONTROWERSJE WEBSCRAPINGU

- **Obciążenie serwerów, kradzież informacji, naruszenie praw autorskich i danych personalnych oraz baz danych**
- **Strony zabezpieczają się w przykładowe sposoby:**
- Zmienne nazwy klas i identyfikatorów
- Pliki typu robots.txt (zasady korzystania z scrapperów)
- Ograniczanie żądań z jednego adresu IP
- CAPTCHA
- Honeypot (serwer otwarty dla botów)

Co chcemy zrobić

Aplikacje która zbiera informacje z Trójmiasta.pl na temat wydarzeń które dzieją się w Trójmieście

Użytkownik będzie mógł wyszukiwać wydarzenia którą są w pobliżu jego lokalizacji oraz zmieniać kategorię i preferencje eventów



```

1  """Moduł do scrapingu wydarzeń z trojmiasto.pl"""
2
3  import requests
4  from bs4 import BeautifulSoup
5  from typing import List
6  import re
7  from config import BASE_URL, TARGET_URL, HEADERS, REQUEST_TIMEOUT, CITY_KEYWORDS
8  from models import Event
9
10
11 def extract_city_from_location(location: str) -> str:
12     """Wyciąga nazwę miasta z lokalizacji"""
13     location_lower = location.lower()
14
15     for city, keywords in CITY_KEYWORDS.items():
16         for keyword in keywords:
17             if keyword in location_lower:
18                 return city.capitalize()
19
20     return "Trojmiasto"
21
22
23 def extract_price_from_text(text: str) -> str:
24     """Próbuje wyciągnąć cenę z tekstu"""
25     # Szukamy wzorców typu: "20 zł", "20-50 zł", "bezpłatnie", "wstęp wolny"
26     price_patterns = [
27         r'\d+[-]\d+\s*zł', # 20-50 zł
28         r'\d+\s*zł', # 20 zł
29         r'bezpłatnie',
30         r'wstęp wolny',
31         r'free',
32         r'gratis'
33     ]
34
35     for pattern in price_patterns:
36         match = re.search(pattern, text.lower())
37         if match:
38             return match.group(0)
39
40     return "Brak informacji"
41

```

SCRAPER

```

1 """Aplikacja Flask do wyszukiwania wydarzeń kulturalnych w Trójmieście"""
2
3 from flask import Flask, jsonify, request, render_template_string
4 from flask_cors import CORS
5 from scraper import scrape_events, get_unique_categories, get_unique_cities
6 from cache import events_cache
7 from models import Event, EventsResponse, City
8 import re
9
10 app = Flask(__name__)
11 CORS(app) # Umożliwia dostęp z frontendu
12
13 # HTML template dla prostego frontendu
14 HTML_TEMPLATE = """
15 <!DOCTYPE html>
16 <html lang="pl">
17 <head>
18     <meta charset="UTF-8">
19     <meta name="viewport" content="width=device-width, initial-scale=1.0">
20     <title>Wydarzenia Kulturalne - Trójmiasto</title>
21     <style>
22         * {
23             margin: 0;
24             padding: 0;
25             box-sizing: border-box;
26         }
27         body {
28             font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
29             background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
30             min-height: 100vh;
31             padding: 20px;
32         }
33         .container {
34             max-width: 1200px;
35             margin: 0 auto;
36         }
37         header {
38             text-align: center;
39             color: white;
40             margin-bottom: 30px;
41         }
42         h1 {
43             font-size: 2.5em;
44             margin-bottom: 10px;
45             text-shadow: 2px 2px 4px rgba(0,0,0,0.3);

```

WIZUALIZACJA



Wydarzenia Kulturalne

Znajdź najciekawsze wydarzenia w Trójmieście



Słowo kluczowe

np. koncert, wystawa...



Miasto

Wszystkie



Kategoria

Wszystkie



Maks. cena (zł)

np. 50

Szukaj wydarzeń

Wyczyść filtry



Brak wyników

Spróbuj zmienić kryteria wyszukiwania


```

1 from pydantic import BaseModel, HttpUrl
2 from typing import Optional
3 from enum import Enum
4
5
6 class City(str, Enum):
7     """Miasta Trójmiasta"""
8     GDANSK = "gdansk"
9     GDYNIA = "gdynia"
10    SOPOT = "sopot"
11    ALL = "wszystkie"
12
13
14 class Event(BaseModel):
15     """Model pojedynczego wydarzenia"""
16     title: str
17     date: str
18     location: str
19     category: str
20     link: Optional[str] = None
21     price: Optional[str] = None
22     city: Optional[str] = None
23
24     class Config:
25         json_schema_extra = {
26             "example": {
27                 "title": "Koncert jazzowy",
28                 "date": "2024-12-15 19:00",
29                 "location": "Gdańska Scena Jazzowa",
30                 "category": "Muzyka",
31                 "link": "https://www.trojmiasto.pl/kultura/",
32                 "price": "50 zł",
33                 "city": "Gdańsk"
34             }
35         }
36
37
38 class EventsResponse(BaseModel):
39     """Model odpowiedzi API z listą wydarzeń"""
40     count: int
41     results: list[Event]
42     cached: bool = False

```

MODEL

DZIĘKUJEMY ZA
UWAGĘ