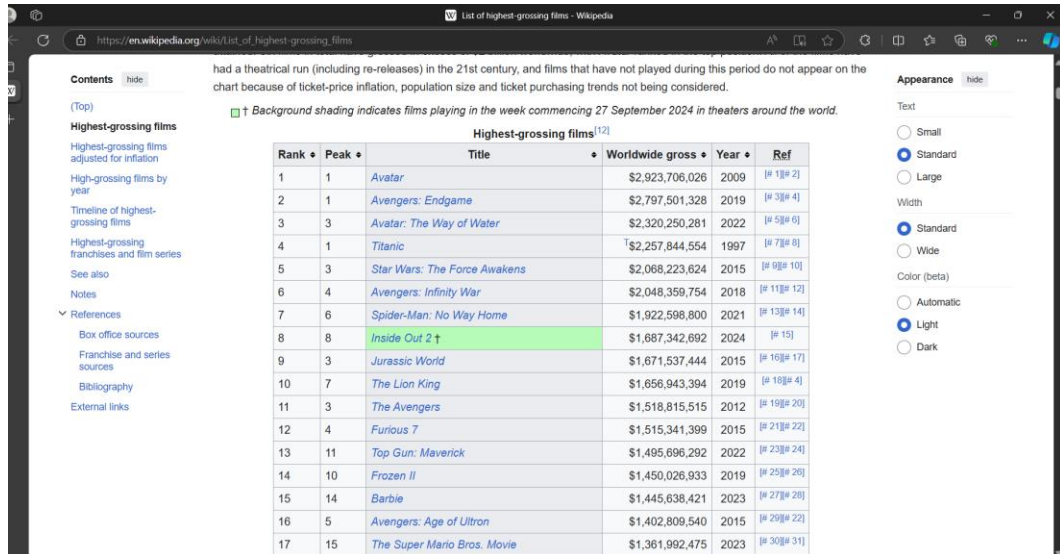


Dokumentasi Simple DE WEB Scrapping to Postgresql

1. Extract data : Scrapping data tabel dari web



The screenshot shows the Wikipedia page for 'List of highest-grossing films'. The table lists movies with their rank, peak rank, title, worldwide gross, year, and a reference link. The movie 'Avatar' is at the top of the list.

Rank	Peak	Title	Worldwide gross	Year	Ref
1	1	<i>Avatar</i>	\$2,923,706,026	2009	[# 1][# 2]
2	1	<i>Avengers: Endgame</i>	\$2,797,501,328	2019	[# 3][# 4]
3	3	<i>Avatar: The Way of Water</i>	\$2,320,250,281	2022	[# 5][# 6]
4	1	<i>Titanic</i>	\$2,257,844,554	1997	[# 7][# 8]
5	3	<i>Star Wars: The Force Awakens</i>	\$2,068,223,624	2015	[# 9][# 10]
6	4	<i>Avengers: Infinity War</i>	\$2,048,359,754	2018	[# 11][# 12]
7	6	<i>Spider-Man: No Way Home</i>	\$1,922,598,800	2021	[# 13][# 14]
8	8	<i>Inside Out 2</i> ↑	\$1,687,342,692	2024	[# 15]
9	3	<i>Jurassic World</i>	\$1,671,537,444	2015	[# 16][# 17]
10	7	<i>The Lion King</i>	\$1,656,943,394	2019	[# 18][# 4]
11	3	<i>The Avengers</i>	\$1,518,815,515	2012	[# 19][# 20]
12	4	<i>Furious 7</i>	\$1,515,341,399	2015	[# 21][# 22]
13	11	<i>Top Gun: Maverick</i>	\$1,495,696,292	2022	[# 23][# 24]
14	10	<i>Frozen II</i>	\$1,450,026,933	2019	[# 25][# 26]
15	14	<i>Barbie</i>	\$1,445,638,421	2023	[# 27][# 28]
16	5	<i>Avengers: Age of Ultron</i>	\$1,402,809,540	2015	[# 29][# 22]
17	15	<i>The Super Mario Bros. Movie</i>	\$1,361,992,475	2023	[# 30][# 31]

1. Extract: Web Scrapping Data dari Wikipedia

```
def extract_data():
    print('memulai ekstrak data')
    url = 'https://en.wikipedia.org/wiki/List_of_highest-grossing_films'
    response = requests.get(url)

    if response.status_code == 200:
        soup = BeautifulSoup(response.text, 'html.parser')

        # Cari tabel yang memiliki class 'wikitable'
        table = soup.find_all('table', {'class': 'wikitable'})[0]

        # Ambil semua baris dari tabel
        rows = table.find_all('tr')

        # Ambil header dari tabel
        headers1 = [header.text.strip() for header in rows[0].find_all('th')]
        print(headers1)

        # Ambil data dari setiap baris tabel
        data = []
        for row in rows[1:]:
            cols = [col.text.strip() for col in row.find_all(['td', 'th'])]
            if cols:
                data.append(cols)
        print(data)
```

Simpan data

```
# Konversi data menjadi DataFrame pandas
df = pd.DataFrame(data, columns=headers1)
print("Data berhasil diekstrak dari web.")
# Simpan data dalam csv
df.to_csv('List Movie Scrap')
```

2. Transformasi data

```
# 2. Transform: Membersihkan dan mengolah data
def transform_data(df):
    # Ambil kolom judul film, pendapatan, dan tahun rilis
    df_transformed = df[['Title', 'Worldwide gross', 'Year']]

    # Bersihkan data, misalnya dengan menghapus karakter khusus dan mengubah tipe data
    df_transformed['Worldwide gross'] = df_transformed['Worldwide gross'].replace({'\': ' ', ',': ' '}, regex=True)
    df_transformed['Worldwide gross'] = pd.to_numeric(df_transformed['Worldwide gross'], errors='coerce')
    df_transformed['Year'] = pd.to_numeric(df_transformed['Year'], errors='coerce')

    # Hapus baris dengan nilai NaN
    df_transformed.dropna(inplace=True)
    print("Data berhasil ditransformasi.")
    return df_transformed
```

3. Load data ke postgresql

```
# 3. Load: Memuat data ke PostgreSQL
def load_data(df_transformed):
    conn = connect_to_db()
    if conn is not None:
        cur = conn.cursor()

        # Buat tabel jika belum ada
        cur.execute('''
        CREATE TABLE IF NOT EXISTS movies (
            title VARCHAR(255),
            worldwide_gross NUMERIC,
            year INTEGER
        )
        ''')

        # Insert data ke dalam tabel movies
        for index, row in df_transformed.iterrows():
            cur.execute("INSERT INTO movies (title, worldwide_gross, year) VALUES (%s, %s, %s)",
                (row['Title'], row['Worldwide gross'], row['Year']))

        conn.commit()
        cur.close()
        conn.close()
        print("Data berhasil dimuat ke PostgreSQL.")
    else:
        print("Gagal memuat data ke PostgreSQL.")
```

Data berhasil ditransformasi.

Data berhasil dimuat ke PostgreSQL.

PS C:\Users\diana\OneDrive\Documents\File Script Postgres\DE> █

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- > Languages
- > Publications
- > Schemas (1)
 - > public
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > Sequences
 - > Tables (4)
 - > employees
 - > employees_transformed
 - > movies
 - > users
 - > Trigger Functions
 - > Types
 - > Views
 - > Subscriptions
- > pegawai
- > postgres
 - > Login/Group Roles
 - > Tablespaces

Dashboard X Properties X SQL X Statistics X Dependencies X Dependents X Processes X guest/postgres@PostgreSQL 16* X

guest/postgres@PostgreSQL 16

Query Query History

```
1 select* from movies;
```

Data Output Messages Notifications

	title	worldwide_gross	year
	character varying (255)	numeric	integer
1	Avatar	2923706026.0	2009
2	Avengers: Endgame	2797501328.0	2019
3	Avatar: The Way of Water	2320250281.0	2022
4	Star Wars: The Force Awakens	2068223624.0	2015
5	Avengers: Infinity War	2048359754.0	2018
6	Spider-Man: No Way Home	1922598800.0	2021
7	Inside Out 2	1687342692.0	2024
8	Jurassic World	1671537444.0	2015
9	The Lion King	1656943394.0	2019
10	The Avengers	1518815515.0	2012
11	Furious 7	1515341399.0	2015
12	Top Gun: Maverick	1495696292.0	2022
13	Frozen II	1450026933.0	2019
14	Barbie	1445638421.0	2023
15	Avengers: Age of Ultron	1402809540.0	2015
16	The Super Mario Bros. Movie	1361992475.0	2023

Total rows: 47 of 47 Query complete 00:00:00.159 Ln 1. Col 20