Lưu vào MySQL

Nguồn 1 (tiki)

import pymysql

import json

from config import MYSQL\_CONFIG # database: crawl\_data\_tiki

INPUT\_FILE = "tiki\_book\_data\_clean.jsonl"

def create\_books\_tiki\_table():

conn = pymysql.connect(

host=MYSQL\_CONFIG["host"],

user=MYSQL\_CONFIG["user"],

password=MYSQL\_CONFIG["password"],

database=MYSQL\_CONFIG["database"],

charset='utf8mb4'

)

cursor = conn.cursor()

cursor.execute("""

CREATE TABLE IF NOT EXISTS books\_tiki (

id INT AUTO\_INCREMENT PRIMARY KEY,

title TEXT,

price INT,

rating FLOAT,

url TEXT

)

""")

conn.commit()

cursor.close()

conn.close()

print("Đã tạo bảng books\_tiki")

def load\_jsonl(filepath):

items = []

with open(filepath, "r", encoding="utf-8") as f:

for line in f:

try:

items.append(json.loads(line))

except:

pass

return items

def save\_to\_books\_tiki(data):

conn = pymysql.connect(

host=MYSQL\_CONFIG["host"],

user=MYSQL\_CONFIG["user"],

password=MYSQL\_CONFIG["password"],

database=MYSQL\_CONFIG["database"],

charset='utf8mb4'

)

cursor = conn.cursor()

insert\_query = """

INSERT INTO books\_tiki (title, price, rating, url)

VALUES (%s, %s, %s, %s)

"""

for item in data:

cursor.execute(insert\_query, (

item.get("title"),

item.get("price"),

item.get("rating"),

item.get("url")

))

conn.commit()

cursor.close()

conn.close()

print(f"Đã lưu {len(data)} dòng vào bảng books\_tiki")

if \_\_name\_\_ == "\_\_main\_\_":

create\_books\_tiki\_table()

data = load\_jsonl(INPUT\_FILE)

save\_to\_books\_tiki(data)

Nguồn 2 (toscrape)

import requests

from bs4 import BeautifulSoup

import json

BASE\_URL = "http://books.toscrape.com"

OUTPUT\_FILE = "books\_raw.jsonl"

BOOKS\_TO\_CRAWL = 1000

BOOKS\_PER\_PAGE = 20

RATING\_MAP = {

"One": 1.0,

"Two": 2.0,

"Three": 3.0,

"Four": 4.0,

"Five": 5.0

}

def get\_page(url):

response = requests.get(url)

response.raise\_for\_status()

return BeautifulSoup(response.text, "html.parser")

def extract\_book\_info(book\_soup):

rating\_str = book\_soup.p['class'][1]

return {

"title": book\_soup.h3.a['title'],

"price": book\_soup.find('p', class\_='price\_color').text,

"rating": RATING\_MAP.get(rating\_str, 0.0), # Mặc định là 0.0 nếu không có rating hợp lệ

"url": BASE\_URL + "/catalogue/" + book\_soup.h3.a['href']

}

def crawl\_books():

books = []

total\_pages = (BOOKS\_TO\_CRAWL + BOOKS\_PER\_PAGE - 1) // BOOKS\_PER\_PAGE

for page in range(1, total\_pages + 1):

print(f"Crawling page {page}...")

url = f"{BASE\_URL}/catalogue/page-{page}.html"

soup = get\_page(url)

articles = soup.find\_all('article', class\_='product\_pod')

for a in articles:

if len(books) < BOOKS\_TO\_CRAWL:

books.append(extract\_book\_info(a))

else:

break

return books

def save\_to\_jsonl(data, filename):

with open(filename, 'w', encoding='utf-8') as f:

for item in data:

f.write(json.dumps(item, ensure\_ascii=False) + '\n')

if \_\_name\_\_ == "\_\_main\_\_":

books = crawl\_books()

save\_to\_jsonl(books, OUTPUT\_FILE)

print(f"Crawl xong {len(books)} sách. Đã lưu vào {OUTPUT\_FILE}")

)