Cài đặt chương trình thu thập  
Làm sạch, chuẩn hóa dữ liệu

Lưu trữ dữ liệu thu thập vào các DBMS

Tiki  
  
- cleaner.py

MYSQL\_CONFIG = {

"host": "localhost",

"user": "crawler",

"password": "123456",

"database": "crawl\_data\_tiki"

}

CATEGORY\_ID = 8322

LIMIT = 20 # 20 sản phẩm / trang

MAX\_PAGE = 75 # 75 trang x 20 = 1500 sản phẩm

* Config.py

MYSQL\_CONFIG = {

"host": "localhost",

"user": "crawler",

"password": "123456",

"database": "crawl\_data\_tiki"

}

CATEGORY\_ID = 8322

LIMIT = 20 # 20 sản phẩm / trang

MAX\_PAGE = 75 # 75 trang x 20 = 1500 sản phẩm

* Crawler.py

import requests

import json

import time

import config

def crawl\_products(page=1):

url = f"https://tiki.vn/api/personalish/v1/blocks/listings?limit={config.LIMIT}&page={page}&category={config.CATEGORY\_ID}"

headers = {"User-Agent": "Mozilla/5.0"}

try:

response = requests.get(url, headers=headers, timeout=10)

response.raise\_for\_status()

return response.json().get('data', [])

except Exception as e:

print(f"Trang {page} lỗi: {e}")

return []

def extract\_fields(product):

rating = product.get("rating\_average")

try:

rating = float(rating) if rating is not None else 0.0

except (ValueError, TypeError):

rating = 0.0

return {

"title": product.get("name"),

"price": product.get("price"),

"rating": rating,

"url": f"https://tiki.vn/{product.get('url\_path')}"

}

def save\_crawled\_data():

total\_products = 0

with open("tiki\_book\_data.jsonl", "w", encoding="utf-8") as f:

for page in range(1, config.MAX\_PAGE + 1):

print(f"Đang crawl trang {page}")

products = crawl\_products(page)

if not products:

break

for product in products:

if total\_products >= 1500:

break

item = extract\_fields(product)

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

total\_products += 1

if total\_products >= 1500:

break

time.sleep(1)

print(f"\nĐã lưu {total\_products} sách vào 'tiki\_book\_data.jsonl'.")

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

save\_crawled\_data()

* save\_to\_mysql.py

import json

import pymysql

from config import MYSQL\_CONFIG

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

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\_mysql(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()

cursor.execute("""

CREATE TABLE IF NOT EXISTS books (

id INT AUTO\_INCREMENT PRIMARY KEY,

title TEXT,

price INT,

rating FLOAT,

url TEXT

)

""")

insert\_query = """

INSERT INTO books (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 MySQL")

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

data = load\_jsonl(INPUT\_FILE)

save\_to\_mysql(data)  
  
  
TOSCRAPE

* clean

import json

import re

INPUT\_FILE = "books\_raw.jsonl"

OUTPUT\_FILE = "books\_clean.jsonl"

EXCHANGE\_RATE = 30000 # 1£ = 30,000 VNĐ (giá trị giả định)

def clean\_price(price\_raw):

if price\_raw is None:

return None

price\_str = str(price\_raw).replace("Â£", "£").strip()

numeric\_part = re.sub(r"[^\d.]", "", price\_str)

try:

gbp\_value = float(numeric\_part) if numeric\_part else None

if gbp\_value is not None:

vnd\_value = int(gbp\_value \* EXCHANGE\_RATE) # Chuyển thành int, bỏ phần thập phân

return vnd\_value

else:

return None

except ValueError:

return None

def clean\_rating(rating\_raw):

rating\_map = {

"One": 1.0,

"Two": 2.0,

"Three": 3.0,

"Four": 4.0,

"Five": 5.0

}

if isinstance(rating\_raw, str):

return rating\_map.get(rating\_raw.strip(), 0.0)

try:

return float(rating\_raw)

except:

return 0.0

def clean\_book(book):

title = book.get("title")

price\_raw = book.get("price")

rating\_raw = book.get("rating")

url = book.get("url")

if not title or not price\_raw or not url:

return None

price\_cleaned = clean\_price(price\_raw)

rating\_cleaned = clean\_rating(rating\_raw)

if price\_cleaned is None:

return None

cleaned = dict(book)

cleaned["price"] = price\_cleaned

cleaned["rating"] = rating\_cleaned

cleaned["title"] = title.strip()

return cleaned

def load\_jsonl(filepath):

data = []

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

for line in f:

try:

data.append(json.loads(line))

except json.JSONDecodeError:

print("Bo qua dong loi: ", line)

return data

def save\_jsonl(data, filepath):

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

for item in data:

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

def clean\_and\_deduplicate():

raw\_data = load\_jsonl(INPUT\_FILE)

seen\_titles = set()

cleaned = []

for book in raw\_data:

cleaned\_book = clean\_book(book)

if cleaned\_book is None:

continue

title\_lower = cleaned\_book["title"].lower()

if title\_lower in seen\_titles:

print("Bo trung lap: ", cleaned\_book['title'])

continue

seen\_titles.add(title\_lower)

cleaned.append(cleaned\_book)

save\_jsonl(cleaned, OUTPUT\_FILE)

print("\nDa clean va luu", len(cleaned), "sach vao", OUTPUT\_FILE)

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

clean\_and\_deduplicate()

* craw

import requests

from bs4 import BeautifulSoup

import json

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

OUTPUT\_FILE = "books\_raw.jsonl"

BOOKS\_TO\_CRAWL = 10

BOOKS\_PER\_PAGE = 20

# Map từ chữ sang số thập phân

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}")

* config

MYSQL\_CONFIG = {

"host": "localhost",

"user": "crawler",

"password": "123456",

"database": "crawl\_data\_uk"

}

* save

import pymysql

import json

from config import MYSQL\_CONFIG # config có database crawl\_data\_uk

def save\_to\_mysql(data):

try:

conn = pymysql.connect(

host=MYSQL\_CONFIG["host"],

user=MYSQL\_CONFIG["user"],

password=MYSQL\_CONFIG["password"],

database=MYSQL\_CONFIG["database"],

charset='utf8mb4'

)

cursor = conn.cursor()

# Tạo bảng nếu chưa có, trường title là UNIQUE để tránh trùng lặp

cursor.execute("""

CREATE TABLE IF NOT EXISTS books (

id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(255) UNIQUE,

price INT,

rating FLOAT,

url TEXT

)

""")

sql = """

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

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

ON DUPLICATE KEY UPDATE

price = VALUES(price),

rating = VALUES(rating),

url = VALUES(url)

"""

for item in data:

cursor.execute(sql, (

item.get("title"),

item.get("price"),

item.get("rating"),

item.get("url")

))

conn.commit()

cursor.close()

conn.close()

print(f"Đã lưu {len(data)} bản ghi vào MySQL database '{MYSQL\_CONFIG['database']}'.")

except Exception as e:

print("Lỗi khi lưu vào MySQL:", e)

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

# Mở file clean jsonl đã tạo bởi file clean.py

with open("books\_clean.jsonl", "r", encoding="utf-8") as f:

data = [json.loads(line) for line in f]

save\_to\_mysql(data)