

Chatbot Guitar Recommend

การซื้อกีตาร์และเครื่องดนตรีออนไลน์ต้องการข้อมูลที่ครบถ้วน และการเปรียบเทียบสินค้าซึ่งใช้เวลานานการพัฒนา Line Chat Bot ตัวนี้จึงช่วยให้ลูกค้าค้นหาข้อมูลได้ง่ายและรวดเร็วพร้อมคำแนะนำสินค้าอัตโนมัติ

โครงสร้าง CHATBOT

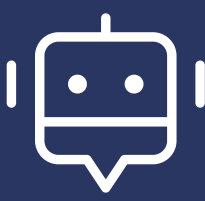
Selenium\ Beautifulsoup

Line API



Scraping Data

ข้อมูล



Chatbot

ข้อความ

การตอบกลับ



การตั้งค่าเริ่มต้นและการเริ่มต้นการเชื่อมต่อ

Python Library

- sys
- os
- json
- logging
- functools import lru_cache
- time
- flask import Flask, request, jsonify
- BeautifulSoup
- selenium import webdriver
- chromedriver_autoinstaller
- selenium.webdriver.chrome.options
- selenium.webdriver.chrome.service
- neo4j import GraphDatabase
- linebot import LineBotApi, WebhookHandler
- linebot.models import FlexSendMessage, TextSendMessage, QuickReply, QuickReplyButton, MessageAction, BubbleContainer, BoxComponent, TextComponent, ImageComponent, ButtonComponent, URIAction
- linebot.exceptions import InvalidSignatureError
- sentence_transformers import SentenceTransformer, util
- faiss
- numpy
- requests

NLP Model

SentenceTransformer('sentence-transformers/distiluse-base-multilingual-cased-v2')

Config Ngrok

```
C:\Users\00M>ngrok config add-authtoken 2k
Auth token saved to configuration file: C:\Users\00M\Ap
il
C:\Users\00M>ngrok http http://localhost:5000

Version Status
Account Jatnipat (Plan: Free)
Update update available (version 3.17.0, Ctrl-U to update)
Version 3.14.1
Region Asia Pacific (ap)
Latency 70ms
Web Interface http://127.0.0.1:4080
Forwarding https://2161-1-46-20-176.ngrok-free.app -> http://localhost:5000

Connections
t1l opn rt1 rt5 p50 p90
0 0 0.00 0.00 0.00 0.00
```

Config Selenium

```
def setup_chrome_driver():
    try:
        chromedriver_autoinstaller.install()
        chrome_options = Options()
        chrome_options.add_argument('--headless')
        chrome_options.add_argument('--no-sandbox')
        chrome_options.add_argument('--disable-dev-shm-usage')
        chrome_options.add_argument('--disable-gpu')
        chrome_options.add_argument('--window-size=1920,1080')
        chrome_options.add_argument('--user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4431.24 Safari/537.36')
    except Exception as e:
        print(f"Error: {e}")

def scrape_guitar_data(sort_url=None, keyword=None):
    if sort_url:
        url = sort_url
    elif keyword:
        url = f"https://www.music.co.th/search?q={keyword}"
    else:
        return TextSendMessage(text="ไม่พบสินค้าที่ค้นหา") # Return error message if both are missing

    # Setup Chrome driver using the defined setup function
    driver = setup_chrome_driver()
    if not driver:
        return TextSendMessage(text="ขออภัย เกิดข้อผิดพลาดในการเชื่อมต่อ")

    driver.get(url)
    driver.implicitly_wait(5)
    html = driver.page_source
    driver.quit()

    soup = BeautifulSoup(html, "html.parser")
    job_elements = soup.find_all("div", {"class": "product-list-item-wrapper"})
```

Config Line

```
# LINE API credentials
ACCESS_TOKEN = '1khrjmlx0l'
SECRET = '7'

line_bot_api = LineBotApi(ACCESS_TOKEN)
handler = WebhookHandler(SECRET)

@app.route("/", methods=['POST'])
def linebot():
    body = request.get_data(as_text=True)
    logging.info(f"Received webhook: {body}")
```

Config Ollama

```
def llama_change(bot_response):
    try:
        OLLAMA_API_URL = "http://localhost:11434/api/generate"
        headers = {"Content-Type": "application/json"}
        payload = {"model": "supachai/llama-3-typhoon-v1.5", "prompt": f"Generate a response: {bot_response}", "stream": False}
        response = requests.post(OLLAMA_API_URL, headers=headers, data=json.dumps(payload))

        if response.status_code == 200:
            response_data = response.json()
            return response_data.get("response", bot_response)
        else:
            return bot_response # ถ้า OLLAMA ใช้งานไม่ได้ ให้ออกข้อความเดิม
    except Exception as e:
        return bot_response # fallback ให้ออกข้อความเดิมถ้าเกิดข้อผิดพลาด
```

Config Neo4j

```
# Neo4j connection setup
NEO4J_URI = "neo4j://localhost:7687"
NEO4J_USER = "neo4j"
NEO4J_PASSWORD = "xxxxxxxxxxxx"

# Initialize Neo4j driver
driver = GraphDatabase.driver(NEO4J_URI, auth=(NEO4J_USER, NEO4J_PASSWORD))
```

Config Flask

```
app = Flask(__name__)

if __name__ == '__main__':
    app.run(debug=True)
```

ตัวอย่างการทำงาน



ทักทาย

เลือกสินค้า

การแนะนำ

โอกาสในการพัฒนา

- พัฒนาความเร็วในการตอบโดยหาโมเดลและวิธีการScrap ข้อมูลที่เหมาะสม
- เพิ่มฟีเจอร์เรื่องการตอบโต้และแจ้งเตือนหากมีโปรโมชั่นต่างๆ
- เพิ่มความแม่นยำในการตอบคำถามและดึงข้อมูลได้มากขึ้น

Github

<https://github.com/JatnipatSangkaneel/Ai-Chatbot-Scraping-Guitar>