

TÁC GIẢ: 8 SYNC DEV

1. IMPORT CÁC THƯ VIỆN CẦN THIẾT



{ 8 Sync Dev }

```
from langchain_core.prompts import ChatPromptTemplate
from langchain_ollama.llms import OllamaLLM
import streamlit as st
```

- **langchain_core.prompts.ChatPromptTemplate**: Đây là lớp được sử dụng để tạo mẫu (template) cho các câu hỏi và câu trả lời trong cuộc trò chuyện.
- **langchain_ollama.llms.OllamaLLM**: Đây là lớp đại diện cho mô hình ngôn ngữ (LLM) **llama3**, cho phép tạo ra phản hồi từ mô hình ngôn ngữ dựa trên đầu vào từ người dùng.
- **streamlit**: Đây là thư viện giúp xây dựng giao diện người dùng (UI) đơn giản trên web.

2. TẠO TIÊU ĐỀ CHO ỨNG DỤNG



{ 8 Sync Dev }

```
st.title("8 Sync Devs - Build ChatGPT")
```

- **st.title()**: Hàm này được sử dụng để đặt tiêu đề cho ứng dụng web, trong trường hợp này là "8 Sync Devs - Build ChatGPT".

3. TẠO MẪU CHO CÂU HỎI VÀ CÂU TRẢ LỜI



{ 8 Sync Dev }

```
template = """Question: {question}

Answer: Let's think step by step."""
```

- Đây là mẫu văn bản (template) cho cuộc hội thoại, trong đó {question} là placeholder cho câu hỏi của người dùng.
- Câu trả lời sẽ luôn bắt đầu bằng “Let’s think step by step.”, gợi ý rằng mô hình sẽ cố gắng suy nghĩ từng bước để trả lời câu hỏi.

4. TẠO ĐỐI TƯỢNG CHATPROMPTTEMPLATE TỪ MẪU



{ 8 Sync Dev }

```
prompt = ChatPromptTemplate.from_template(template)
```

- **ChatPromptTemplate.from_template()**: Phương thức này được sử dụng để tạo ra một đối tượng **ChatPromptTemplate** từ mẫu đã tạo ở trên. Đối tượng này sẽ kết hợp câu hỏi của người dùng với mẫu để tạo ra đầu vào cho mô hình ngôn ngữ.

5. KHỞI TẠO MÔ HÌNH NGÔN NGỮ OLLAMA



{ 8 Sync Dev }

```
model = OllamaLLM(model="llama3")
```

- `OllamaLLM(model="llama3")`: Đây là cách để khởi tạo một mô hình ngôn ngữ `llama3`, sử dụng lớp `OllamaLLM`. Mô hình này sẽ được sử dụng để tạo ra phản hồi dựa trên câu hỏi của người dùng.

6. KẾT HỢP MẪU VÀ MÔ HÌNH NGÔN NGỮ THÀNH MỘT CHUỖI XỬ LÝ



{ 8 Sync Dev }

```
chain = prompt | model
```

- `|` (pipe operator): Đây là toán tử được sử dụng để kết hợp hai đối tượng `prompt` và `model` thành một chuỗi xử lý (`chain`). Chuỗi này sẽ nhận đầu vào là câu hỏi, áp dụng mẫu câu hỏi, và sau đó gửi đến mô hình ngôn ngữ để tạo ra câu trả lời.

7. TẠO HỘP NHẬP LIỆU VÀ XỬ LÝ CÂU HỎI



{ 8 Sync Dev }

```
question = st.chat_input("Enter your question here")
if question:
    st.write(chain.invoke({"question": question}))
```

- `st.chat_input("Enter your question here")`: Tạo một hộp nhập liệu trên giao diện web để người dùng có thể nhập câu hỏi.
- `chain.invoke({"question": question})`: Gọi hàm `invoke` để xử lý câu hỏi từ người dùng thông qua chuỗi xử lý (`chain`). Phương thức này nhận đầu vào là một từ điển với cặp key-value, trong đó key là `question` và value là câu hỏi người dùng nhập vào.
- `st.write()`: Hiển thị câu trả lời từ mô hình lên giao diện web.

CỘNG ĐỒNG



[Kevin Nguyễn](#)



[Nhóm Chia Sẻ Công Nghệ](#)



[Nhóm BlockChain](#)



[Tiktok: 8 Sync](#)



[Youtube: 8 Sync Dev](#)



[Zalo](#)

KHÓA HỌC:



[Fullstack Python](#)



[Fullstack Nextjs](#)



[Fullstack Android-IOS](#)

Tài liệu sẽ được cập nhật định kì và thông báo trong group nên các bạn chú ý nhen .

Các khóa học khác xem tại : [8 Sync Dev](#)

Hoặc liên hệ trực tiếp qua fb: [Kevin Nguyễn](#) để được hỗ trợ

