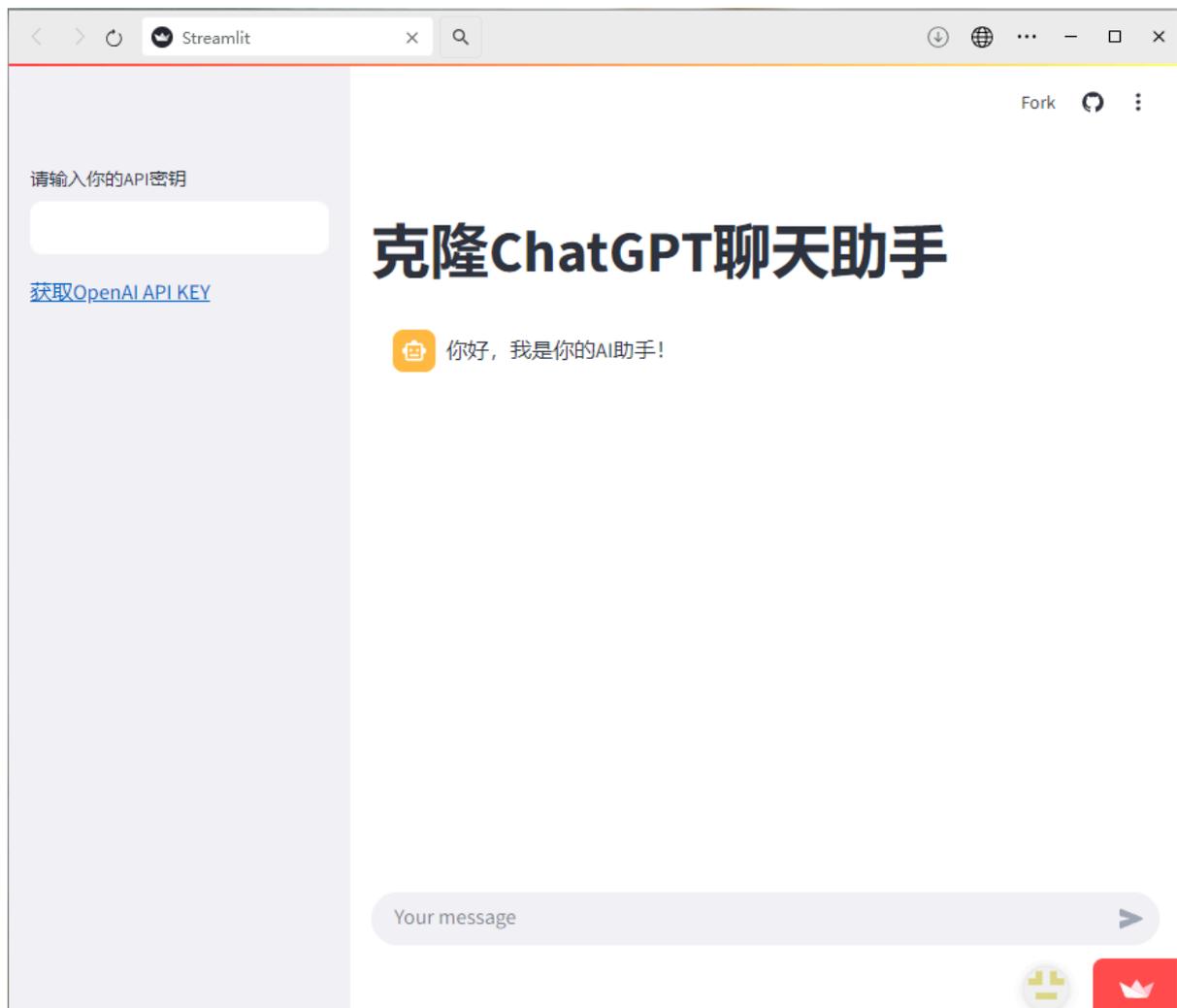
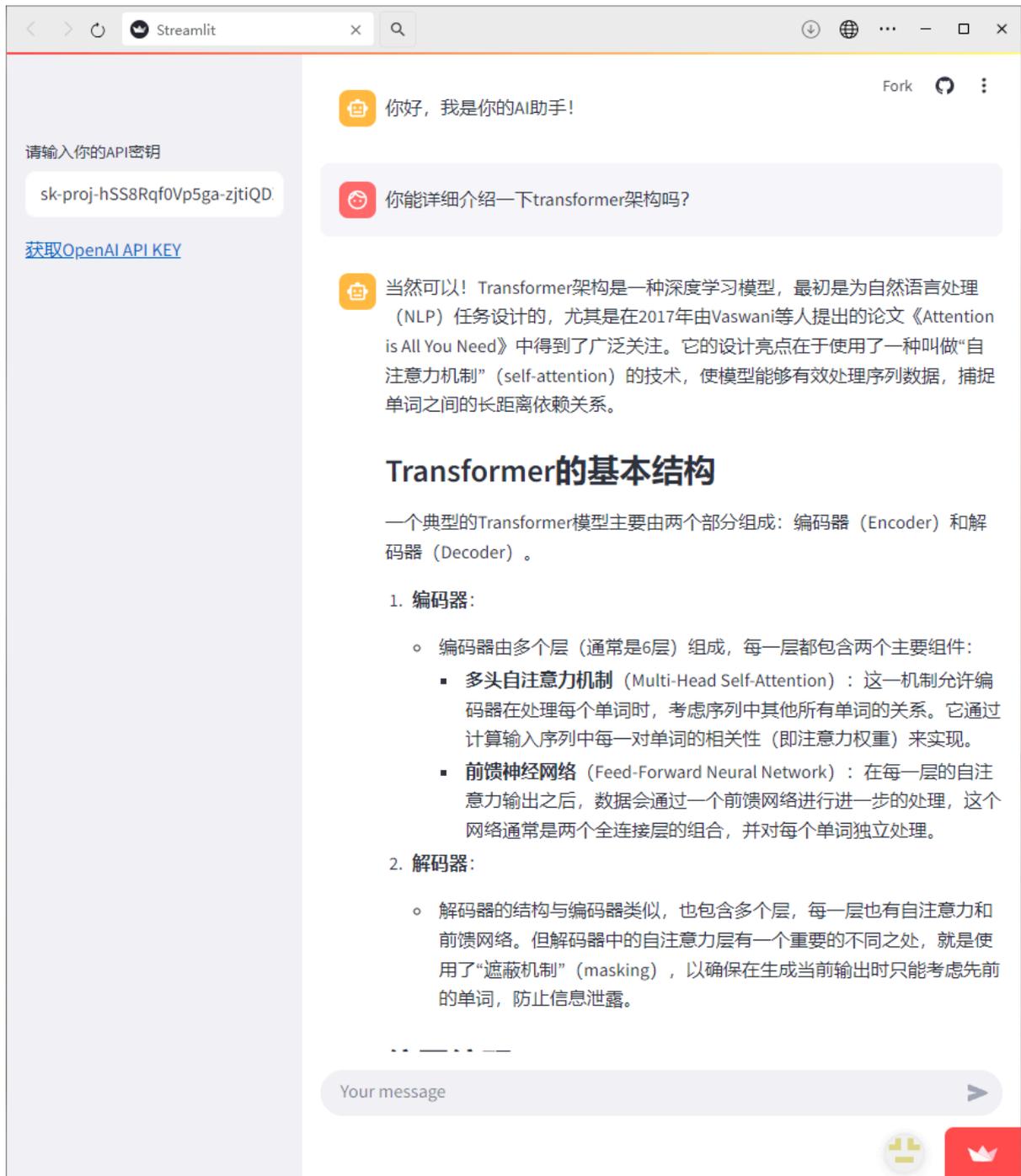


# 聊天机器人网址： <https://chatboximitate.streamlit.app/>

🕒 创建时间	@2025年7月17日 21:44
🏷️ 标签	

- 🌟 网页需要科学上网才可见 许久不打开需要手动刷新
- 🌟 本视频脚本生成器调用OpenAI API
- 🌟 下方为前后效果图





```
from langchain.chains import ConversationChain
from langchain_openai import ChatOpenAI
#from langchain.memory import ConversationBufferMemory
#import os
#定义一个函数 调用API 注意这里记忆应该是来自外部对话
def get_chat_response(prompt,memory,openai_api_key):
#创建模型实例
    model=ChatOpenAI(model="gpt-4o-mini",openai_api_key=openai_api_key)
```

```

#创建对话链实例 (传入模型 和记忆)
    chain=ConversationChain(llm=model,memory=memory)
#固定格式chain.invoke{"input":prompt}
    response=chain.invoke({"input":prompt})
#response是一个列表 其中ai的回答在response中所以可以这样表示
    return response["response"]
#memory = ConversationBufferMemory(return_messages=True)
#print(get_chat_response("牛顿提出过哪些知名的定律?", memory, os.getenv("OPENAI_API_KEY")))

```

```

import streamlit as st
from utils import get_chat_response
from langchain.memory import ConversationBufferMemory

st.title("克隆ChatGPT聊天助手")

with st.sidebar:
    openai_api_key=st.text_input("请输入你的API密钥")
    st.markdown("[获取OpenAI API KEY](https://platform.openai.com/account/api-keys)")
#因为每次用户一交互就会重新运行整个编码 那么之前的聊天记录就会归零 我们需要借助会话状态st.session_state来初始化记忆 只有会话状态里面没有记忆才初始化记忆
#这里的代码意思就是如果会话状态里没有记忆 就创建一个类似记忆的键(仓库)等于新创建的记忆
#创建一个message键(专门来显示消息)储存AI和用户的消息 这里输入AI的初始化消息
if "memory" not in st.session_state:
    st.session_state["memory"]=ConversationBufferMemory(return_messages=True)
    st.session_state["messages"]=[{"role":"ai","content":"你好, 我是你的AI助手! "}]
#上面都是储存消息 页面上看不到 要展示消息(运行后就是展示过往的消息)
for message in st.session_state["messages"]:
    st.chat_message(message["role"]).write(message["content"])
#创建prompt 这里用st.chat_input 因为输入以后储存的是消息
prompt=st.chat_input()
if prompt:
    if not openai_api_key:
        st.info("请输入你的OpenAI API KEY")
        st.stop()
#如上面 先储存对话 之后再展示出来对话
    st.session_state["messages"].append({"role":"human","content":"prompt"})
    st.chat_message("human").write(prompt)

```

```
with st.spinner("AI正在思考中, 请稍等。。。。。。 "):
    response=get_chat_response(prompt,st.session_state["memory"],o
penai_api_key)
    msg={"role":"ai","content":response}
    st.session_state["messages"].append(msg)
    st.chat_message("ai").write(response)
```