

Python Coding

LLM Model API

LMStudio

```
from langchain.llms import OpenAI

#set llm for langchain using model from lmstudio
llm = OpenAI(
    openai_api_base='http://localhost:1234/v1',
    openai_api_key='NULL'
)
```

```
import streamlit as st
from openai import OpenAI

# Set up the Streamlit App
st.title("ChatGPT Clone using Llama-3 🦙")
st.caption("Chat with locally hosted Llama-3 using the LM Studio 🦙")

# Point to the local server setup using LM Studio
client = OpenAI(base_url="http://localhost:1234/v1", api_key="lm-studio")

# Initialize the chat history
if "messages" not in st.session_state:
    st.session_state.messages = []

# Display the chat history
for message in st.session_state.messages:
    with st.chat_message(message["role"]):
        st.markdown(message["content"])

# Accept user input
if prompt := st.chat_input("What is up?"):
```

```

# Add user message to chat history
st.session_state.messages.append({"role": "user", "content": prompt})

# Display user message in chat message container
with st.chat_message("user"):
    st.markdown(prompt)

# Generate response
response = client.chat.completions.create(
    model="lmstudio-community/Meta-Llama-3-8B-Instruct-GGUF",
    messages=st.session_state.messages, temperature=0.7
)

# Add assistant response to chat history
st.session_state.messages.append({"role": "assistant", "content": response.choices[0].message.content})

# Display assistant response in chat message container
with st.chat_message("assistant"):
    st.markdown(response.choices[0].message.content)

```

GPT

```

from langchain_openai import ChatOpenAI

llm = ChatOpenAI(
    model="gpt-4o",
    temperature=0,
    max_tokens=None,
    timeout=None,
    max_retries=2,
    # api_key="...",
    # base_url="...",
    # organization="...",
    # other params...
)

```

Ollama

```

from langchain_community.llms import Ollama

llm = Ollama(model="llama2:13b")

llm.invoke("The first man on the moon was ... think step by step")

```

Chunking/Splitting

??????

```
# Unicode []
# \u3002 []
# \uff0c []
# Get Unicode for specific character
# >>> ' '.encode('unicode-escape') # for py3
# >>> list(u' ') # for py2

import re
text = "[]"
chunks = re.split('[\u3002\uff0c]', text)
#print("\n\n".join([chunk for chunk in chunks]))
for chunk in chunks:
    print("---" * 10)
    print(chunk)
```

??????

```
# \s+ []
chunks = re.split(r'(?<=[?!])\s+', text)
```

Revision #10

Created 3 June 2024 13:53:11 by Admin

Updated 20 August 2024 17:46:55 by Admin