Building a Chatbot with Chainlit and Langchain

This project utilizes two powerful tools, **Chainlit** and **Langchain**, to create a user-friendly chatbot application. Users can interact with this chatbot and have conversations powered by a large language model (LLM).
Core Functionalities:

- **Chatbot Interface**: Chainlit provides the building blocks for the chatbot's interface. This interface allows users to interact with the system by sending text messages, creating a natural conversation flow.
- **LLM Integration**: Langchain bridges the gap between the application and a pre-trained LLM hosted on Hugging Face Hub. In this case, the "gpt2-medium" model is used for conversation generation.
- **Conversational Flow Management:**
    - A prompt template guides the conversation structure, ensuring a smooth flow for user queries.
    - The LLM and prompt details are stored for efficient handling of user messages, optimizing performance.
    - By leveraging user inputs, the LLM generates responses, creating an interactive conversation experience.

Steps Involved:

1. **Gathering Necessary Tools:** The first step involves importing essential libraries like Chainlit, Langchain, libraries for secure API token handling, and other functionalities required for the application to run.

2. **API Token and Model Configuration:**
    - A secure API token is retrieved to access the chosen LLM model on Hugging Face Hub.
    - The "gpt2-medium" model is selected as the LLM for conversation generation.
    - Specific model parameters are configured to control the variation and length of the LLM's responses.

3. **Chatbot Setup:**
    - A prompt template is created, incorporating placeholders for user queries to guide the conversation.
    - An LLMChain object is constructed, essentially linking the chosen LLM model with the defined prompt template.
    - This LLMChain is stored for later use, ensuring efficient retrieval during user interactions.

4. **Handling User Messages:**
    - Whenever a user sends a message, the stored LLMChain is retrieved for processing.
    - The user's message is passed to the LLM through the LLMChain, triggering the generation of a response.

    - The LLM's response is then sent back to the user as a new message, continuing the conversation.

Overall Purpose:

This approach provides a foundation for building a chatbot application. It leverages Chainlit's user-friendly interface capabilities and Langchain's LLM integration functionalities to create an interactive conversation experience powered by a pre-trained LLM.