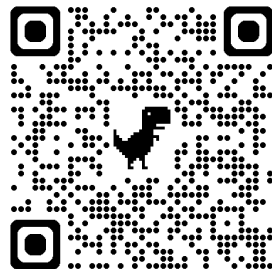


LangChain Overview

Slides & Code available in:
<https://tinyurl.com/kwlangchain>



Agenda

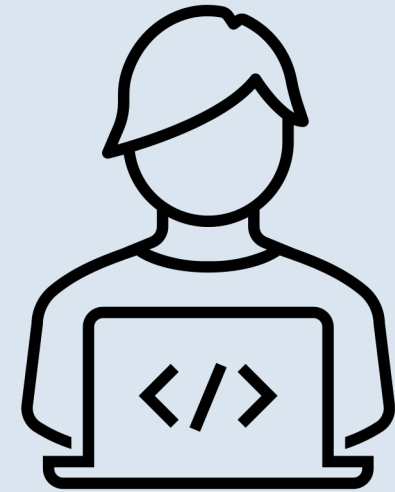
- Introduction
- LangChain Overview
- Simple Use Cases
- Questions

Introduction



About myself:

- LangChain Ambassador
- Courseware Developer in AI Singapore

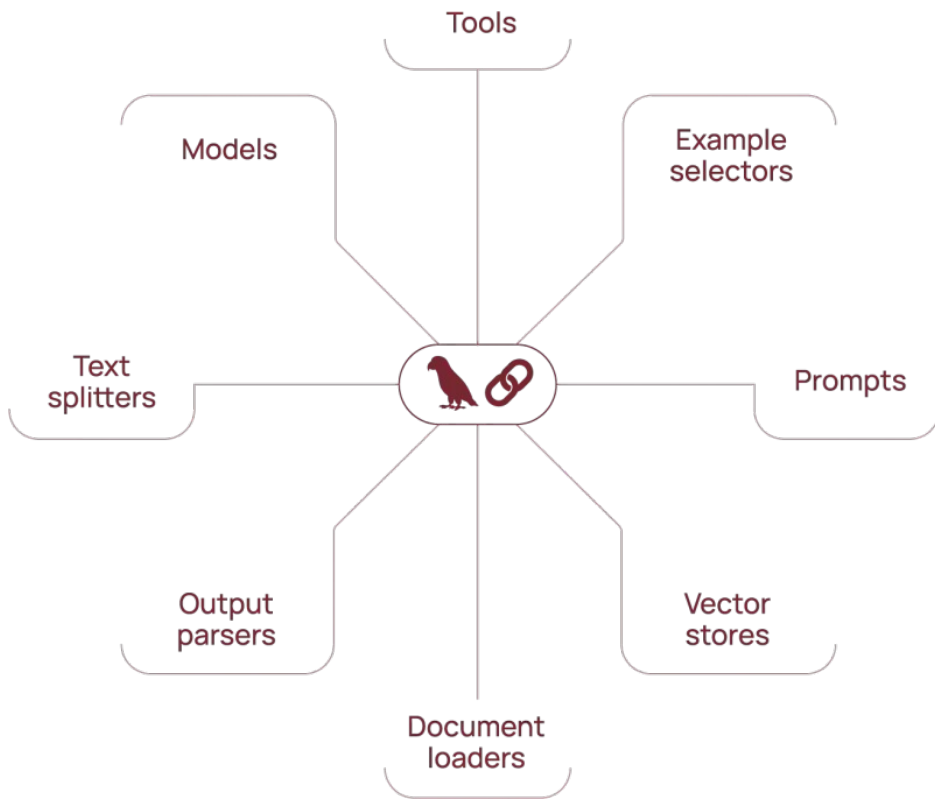


LangChain Overview



What is LangChain?

- Framework designed to streamline the development of applications powered by LLMs



- MIT Licensed (Free and Open Source)
- 100+ Community Integrations
- Used to build RAG apps
- Used to build Agentic LLM applications

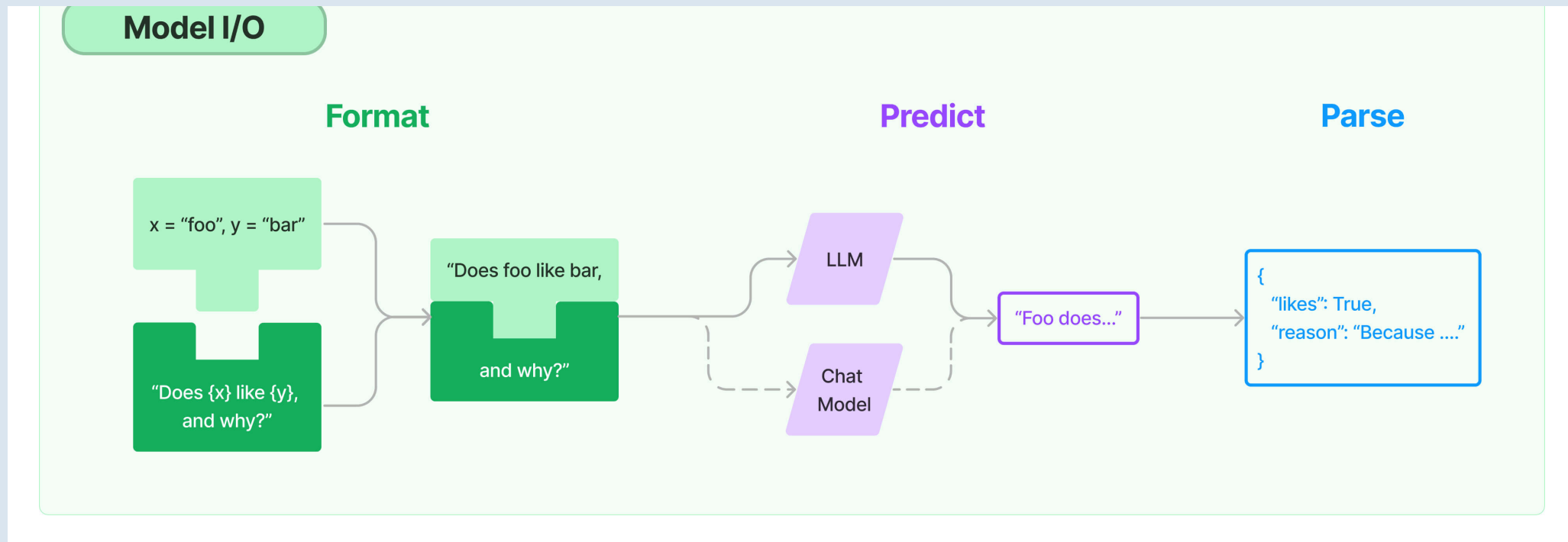


Key Features



Prompt Engineering

- Reuse prompts in run time (no need to repeat)
- Modify Prompts in run time (e.g. add conversation history, change the task etc.)



LLM Wrappers

- Unifies the method of sending messages to different LLM providers
- Learn once, use any models: GPT (OpenAI)

```
from langchain_openai import ChatOpenAI

llm = ChatOpenAI(
    model="gpt-4o",
    temperature=0,
    max_tokens=None,
    timeout=None,
    max_retries=2,
    # api_key="...", # if you prefer to pass api key in directly instead of using env vars
    # base_url="...",
    # organization="...",
    # other params...
)
```

LLM Wrappers

- Unifies the method of sending messages to different LLM providers
- Learn once, use any models: Claude (Anthropic)

```
from langchain_anthropic import ChatAnthropic

llm = ChatAnthropic(
    model="claude-3-5-sonnet-20240620",
    temperature=0,
    max_tokens=1024,
    timeout=None,
    max_retries=2,
    # other params...
)
```

Standard Interface - Runnableables

- Each component follows the “Runnable” protocol which have the same methods for running the component
- (e.g. Invoke Method for Prompt Templates)

```
from langchain_core.prompts import ChatPromptTemplate

prompt_template = ChatPromptTemplate([
    ("system", "You are a helpful assistant"),
    ("user", "Tell me a joke about {topic}")
])

prompt_template.invoke({"topic": "cats"})
```

Standard Interface - Runnables

- Each component follows the “Runnable” protocol which have the same methods for running the component
- (e.g. Invoke Method for LLMs)

```
messages = [  
    (  
        "system",  
        "You are a helpful assistant that translates English to French. Translate the user ser  
    ),  
    ("human", "I love programming."),  
]  
ai_msg = llm.invoke(messages)  
ai_msg
```

Standard Interface - Runnables

```
from langchain_core.prompts import ChatPromptTemplate

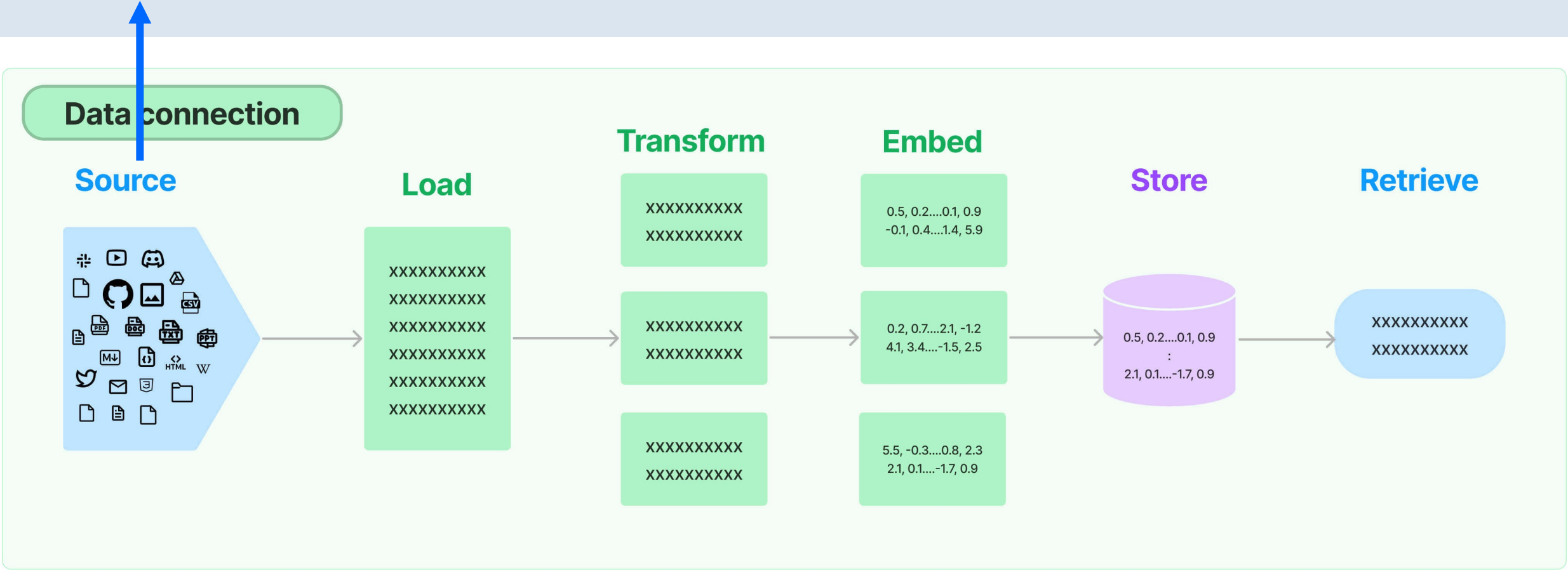
prompt = ChatPromptTemplate.from_messages(
    [
        (
            "system",
            "You are a helpful assistant that translates {input_language} to {output_language}",
        ),
        ("human", "{input}"),
    ]
)

chain = prompt | llm
chain.invoke(
    {
        "input_language": "English",
        "output_language": "German",
        "input": "I love programming.",
    }
)
```

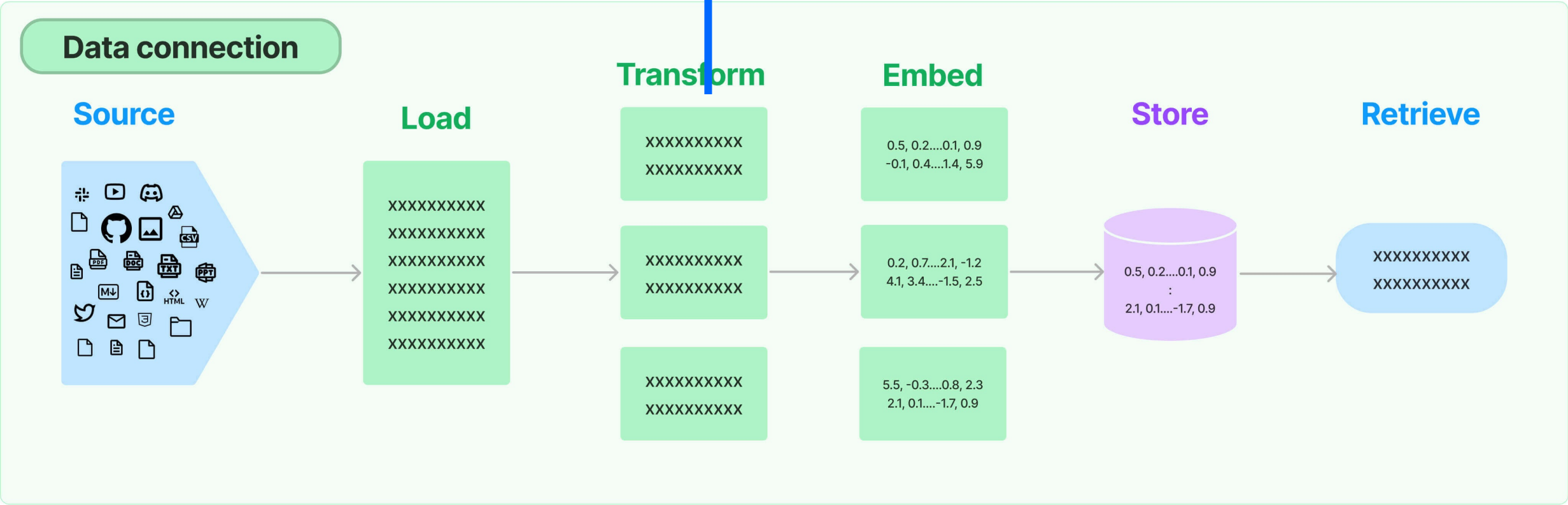
Fill Up Prompt Template
Pass in the Prompt
Get the AI Answer

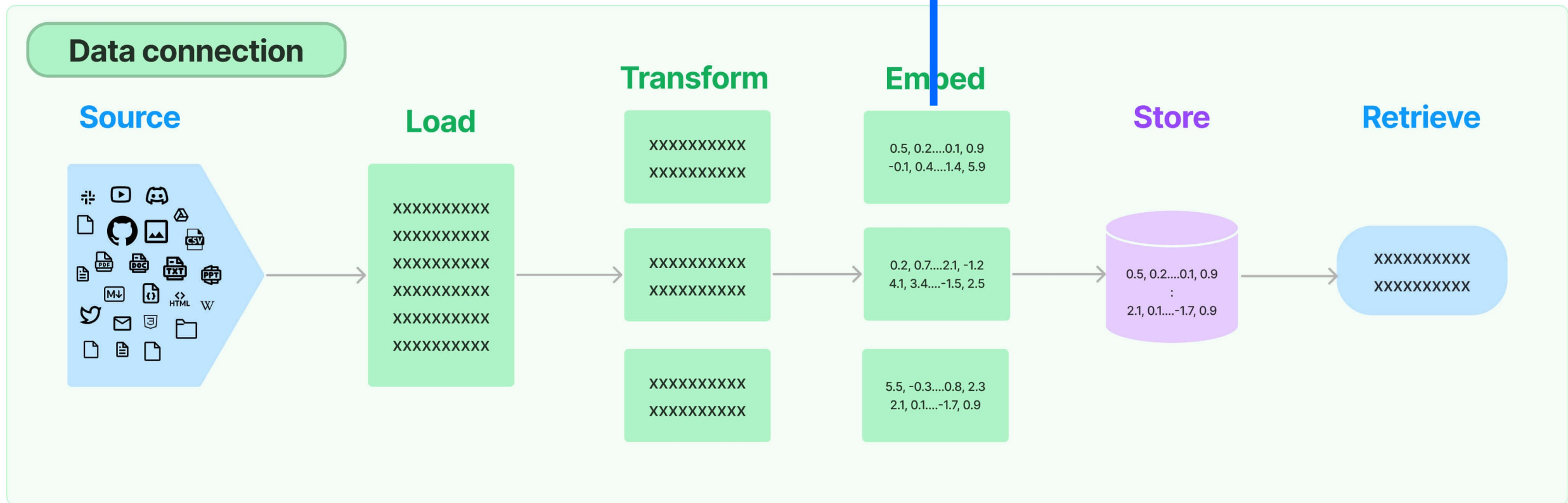
RAG Modules

Document Loaders



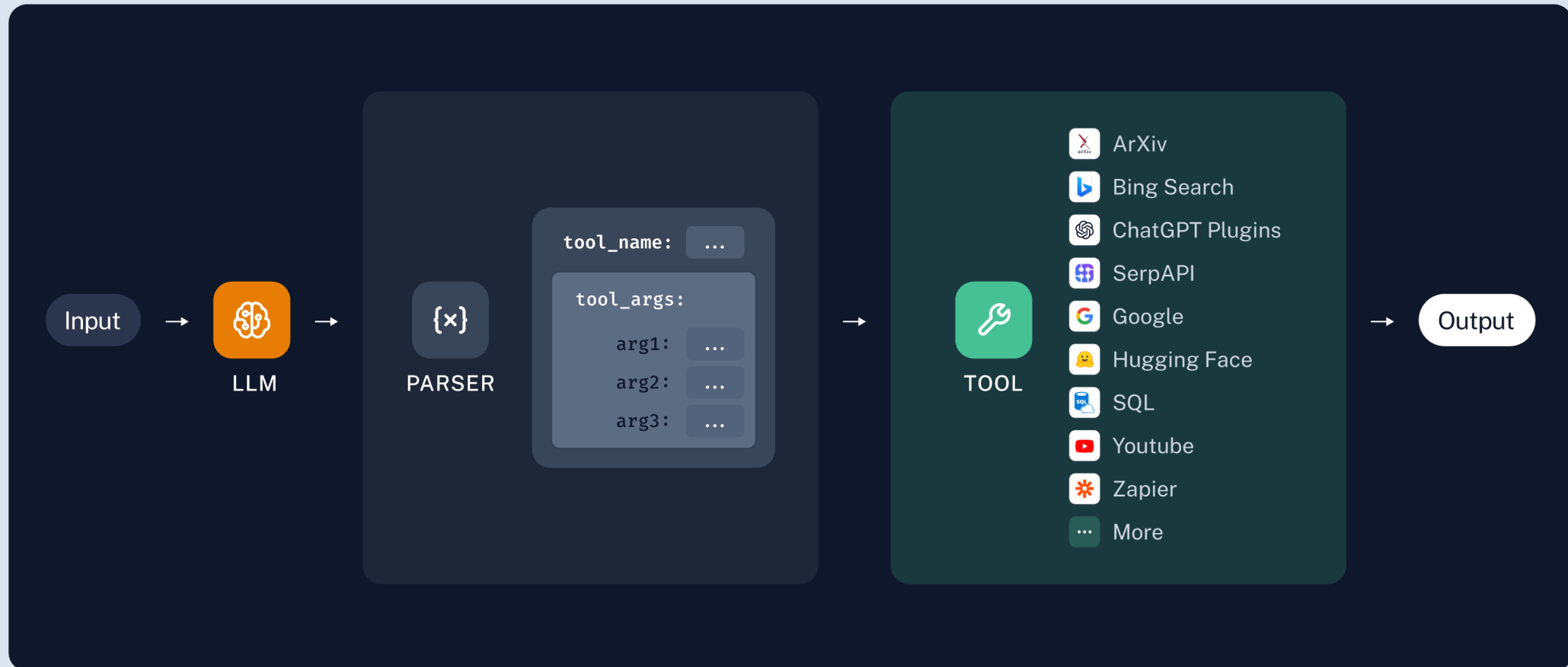
RAG Modules



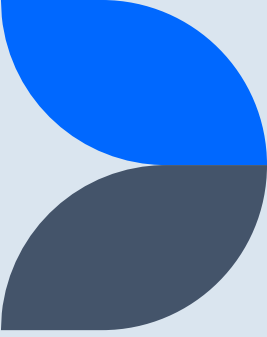


Tools

- Connect your LLM to Python Functions, Search APIs, Excel Modifier
- Build and use your own tool for with the standard interface



Agents (with LangGraph)



LangChain Inc > LangGraph

agent

Interrupts

687ae1dd-b802-4df8-b332-5203b3bb6dbe
13 seconds ago

Pretty Data + New Thread

write me a python script to track my MAUs
25 seconds ago __start__

Here's a Python script that can help you track your Month...
13 seconds ago agent

Message
Here's a Python script that can help you track your Monthly Active Users (MAUs):

```
python
import datetime

# Dictionary to store user activity
user_activity = {}

def log_user_activity(user_id):
    """
    Logs the current month's activity for a given user.
    """
    current_month = datetime.datetime.now().strftime("%Y-%m")
    if user_id in user_activity:
        user_activity[user_id].add(current_month)
    else:
        user_activity[user_id] = set([current_month])
```

__start__

agent

action

__end__

continue

end

Input

Messages

+ Message

Configure

Submit

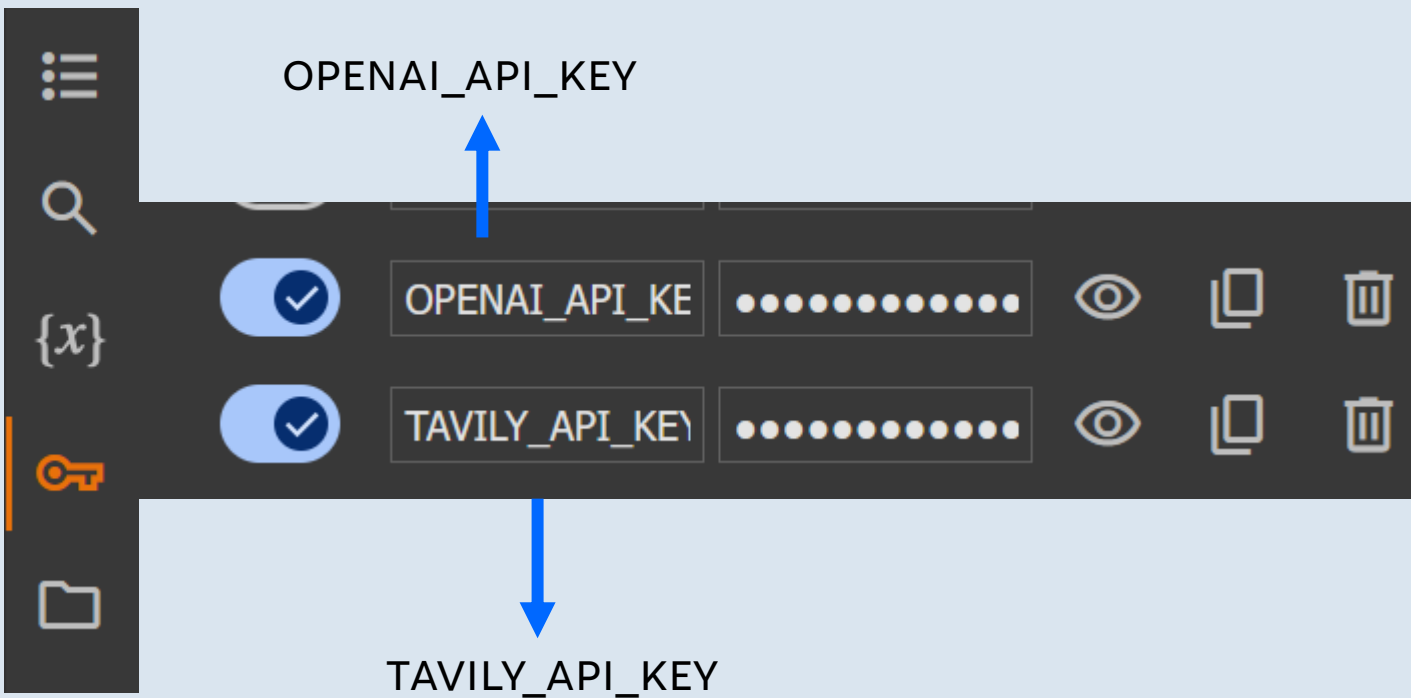
- Pass in all the tools you created to automate your life 😊

Simple Use Cases



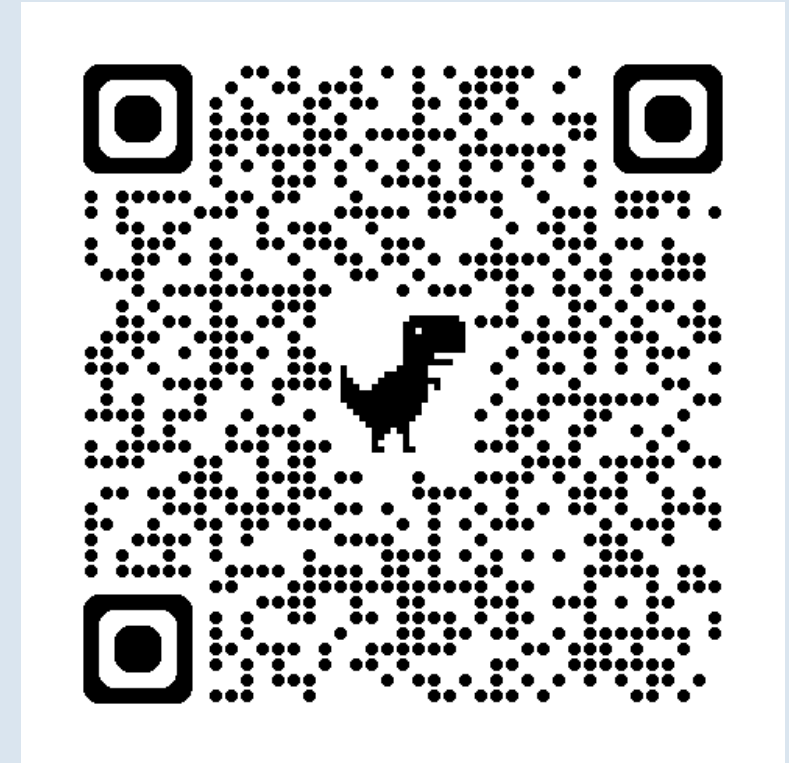
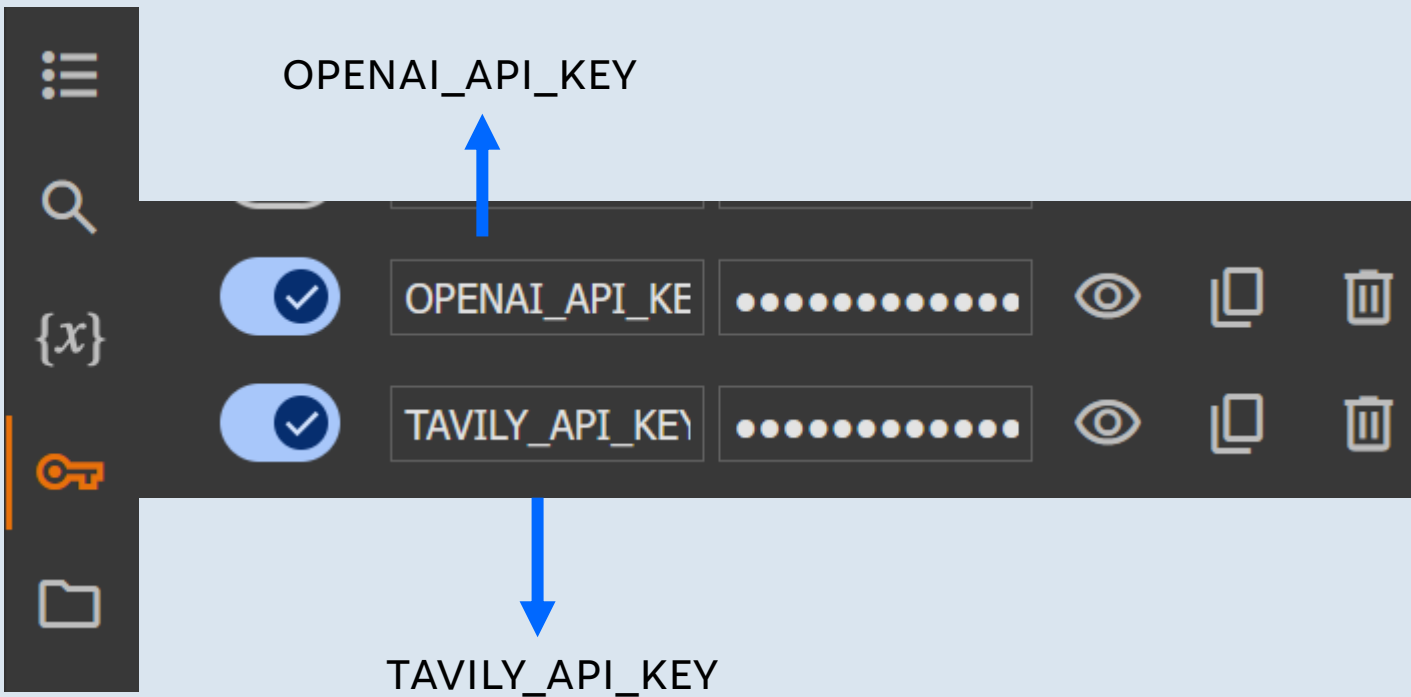
Colab Notebooks

<https://tinyurl.com/samplelangchain>



Colab Notebooks

<https://tinyurl.com/samplelangchain>

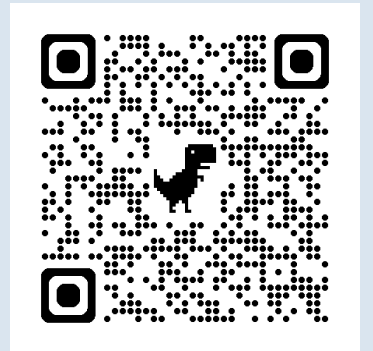


Bonus Demo!



LangGraph & Custom Tool Demo

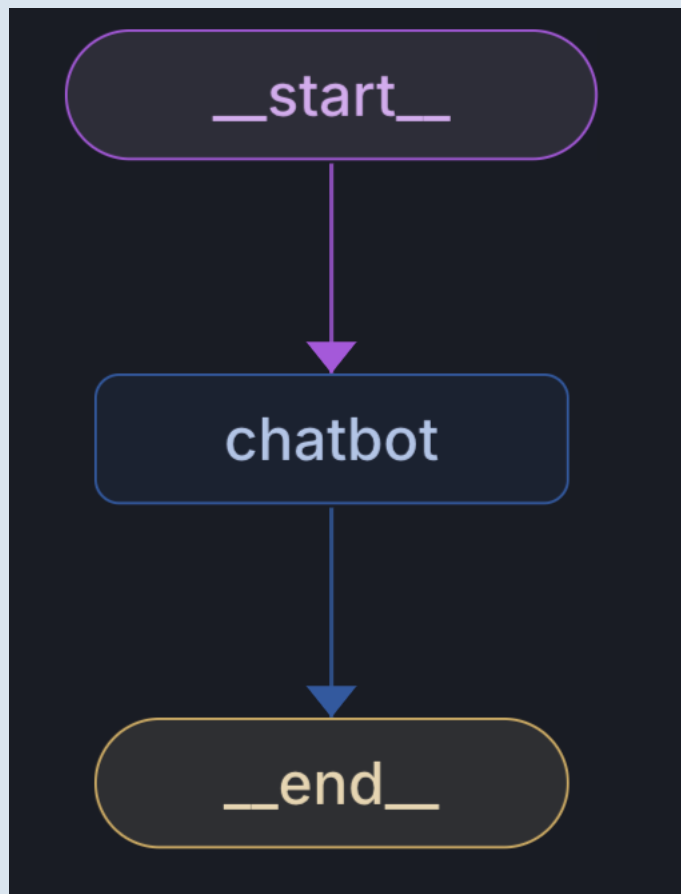
- Released on 19th Nov 😊
- GUI for visualising graph invocation with control



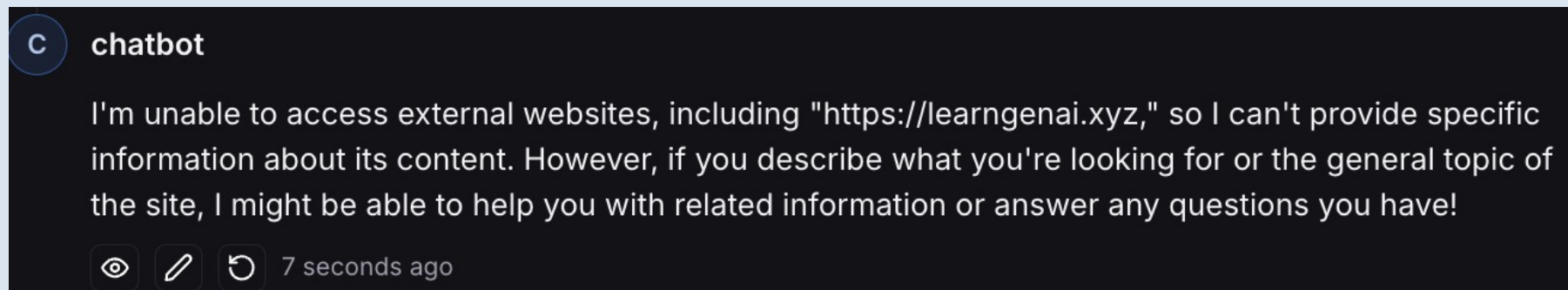
<https://github.com/DangerousPotential/LangGraph-Desktop-Demo>

LangGraph & Custom Tool Demo

Normal Chatbot



Unable to give response

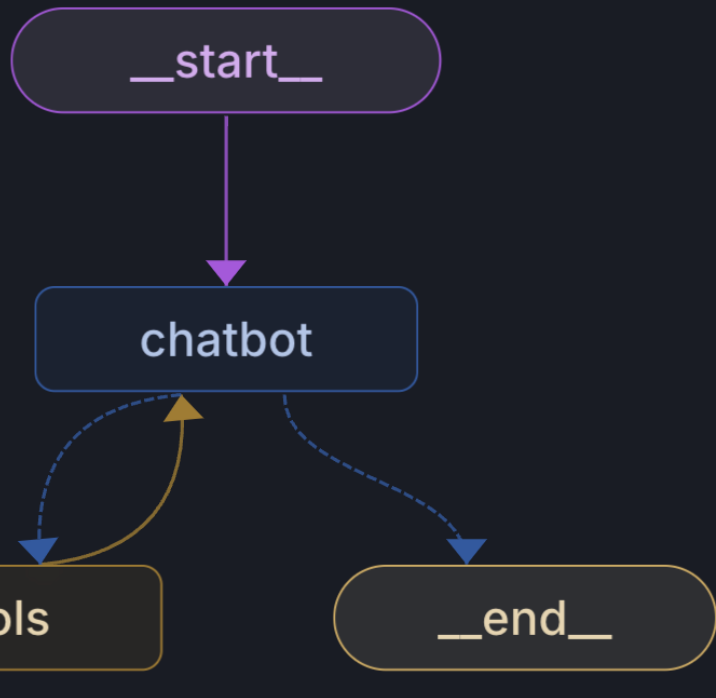


Design a Screenshot Tool with LangChain +
use multi-modal capability of LLMs

LangGraph & Custom Tool Demo

ReAct Chatbot

Gives Response!



c chatbot

The webpage at "<https://learngenai.xyz>" contains the following information:

****Learn GenAI****

Self-learn Python and Gen AI!

Sit, Learn, Write, Repeat

- ****About Me****

- ****Intro to Python****

Learn the basics of Python Programming in next to no time!

- ****LangChain (RAG Route)****

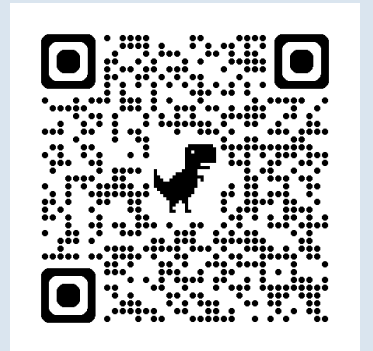
Learn how to empower your LLMs with external data sources.

There are also navigation links for "Home," "Intro to Python," and "LangChain."

If you need more specific details or have any questions about the content, feel free to ask!

Try it Yourself!

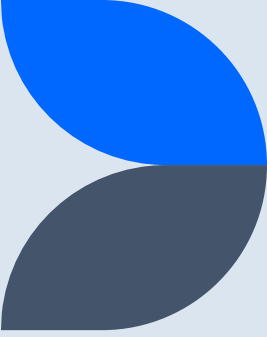
- Clone the Repo
- Set up Virtual Environment
- Create environment file with OPENAI API KEY
- Run ``langgraph serve``



<https://github.com/DangerousPotential/LangGraph-Desktop-Demo>

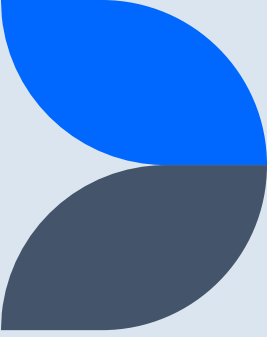
(FREE) Learning Resources





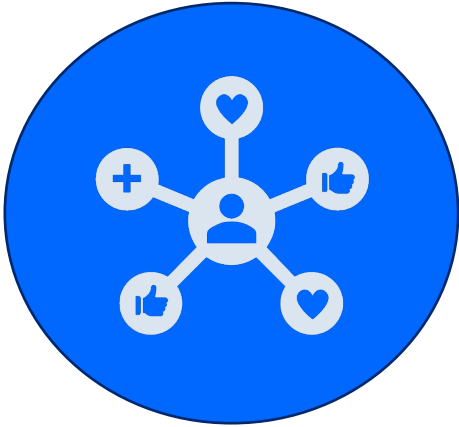
Resources (LangChain: RAG):

- <https://learngenai.xyz/>
- <https://www.deeplearning.ai/short-courses/langchain-chat-with-your-data/>



Resources (LangGraph: Tools & Agents):

- <https://academy.langchain.com/courses/intro-to-langgraph>
- <https://www.deeplearning.ai/short-courses/ai-agents-in-langgraph/>



- Github: DangerousPotential
- Linkedin: [Kim Wee Teh](#)

Contact Me for Feedback!
langchain@aicomputing.sg



Questions?

Thanks!





Thanks