All Articles

Power your Apps: Leverage Natural Language, OpenAI, LangChain and Custom APIs

Natural Language Processing and OpenAI

As the world becomes increasingly digital, the use of natural language processing (NLP) in apps is becoming more and more common. NLP allows users to interact with apps using natural language, making the experience more intuitive and user-friendly. OpenAI is one of the leading companies in the field of NLP, offering a range of powerful tools and APIs to developers. With OpenAI's APIs, you can easily add NLP functionality to your app.

By incorporating OpenAI's APIs into your app, you can create a more engaging and interactive experience for your users. Whether you're building a chatbot, an e-commerce site, or a search engine, OpenAI's NLP features can help you take your app to the next level.

LangChain

LangChain is primarily a framework to use with Language Model (LLM) technology. It allows for the development of applications for chatbots that support Generative Question-Answering(GCA). Essentially, LangChain is designed to “chain” various components(LLMs, APIs, document retrieval, peristence, etc.) together, allowing for the creation of advanced LLM applications.

By leveraging the power of LangChain and OpenAI APIs, we can create applications that are not only innovative and powerful but also provide maximum value to our users.

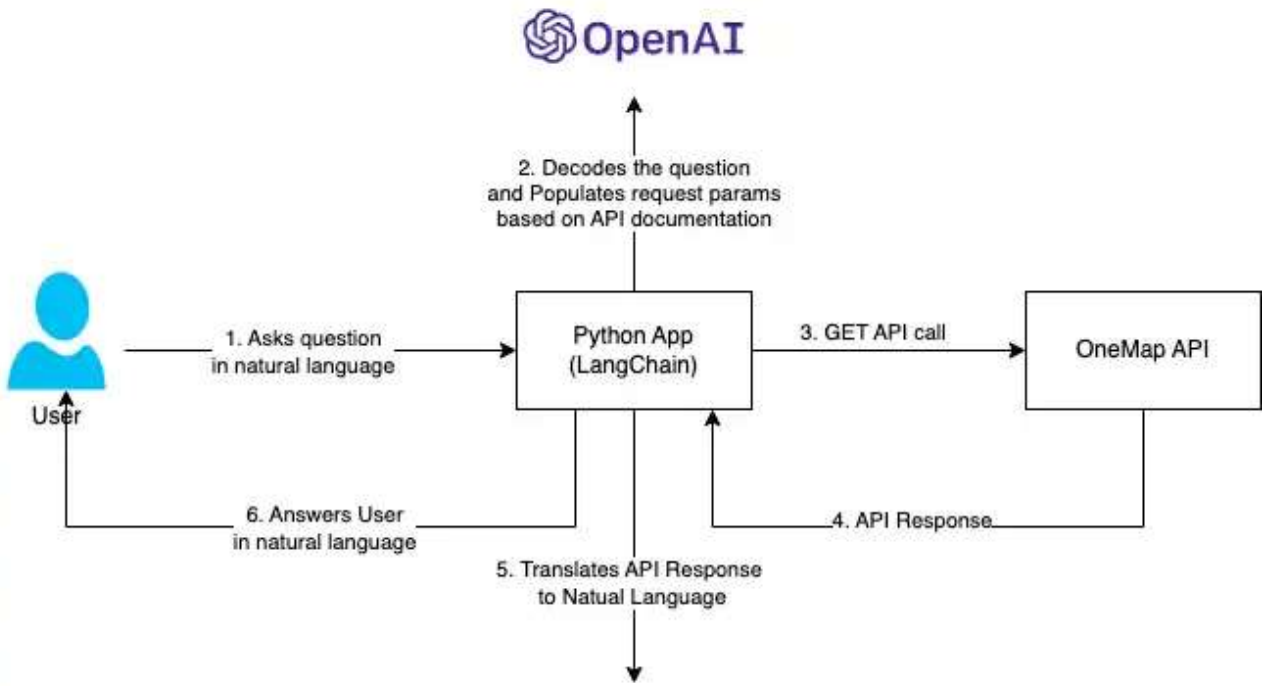
APIChain

APIChain is a feature of LangChain that allows your application to understand natural language and translate it into API calls. This means that your application can leverage the power of APIs to provide your users access to real-time data and provide them better user experience - Imagine an application which requires users to configure so many filters just to be able to search data. For example in a classified ads website, you need to specify filters to narrow down your search and find the specific item you need.

By allowing for seamless integration with various APIs, APIChain allows you to create advanced applications that can perform complex tasks based on user input.

Searching the Street Address or Postal Code using LangChain and Singapore's OneMap API

In this example, let's use LangChain's APIChain feature to retrieve the street address and postal code of "OurTampines Hub" from the OneMap API. We can achieve this by creating a LangChain instance and calling its APIChain function with the appropriate API parameters.



The OneMap API would then return the desired information, which LangChain would process and return to the user in a natural language format. For instance, LangChain might respond with “The street address of Our Tampines Hub is “1 TAMPINES WALK OUR TAMPINES HUB SINGAPORE 528523”.

The result of the API is a JSON object. We can then use the JSON object to retrieve the desired information and return it to the user in a natural language format. This will be done automatically by LangChain.

OneMap API Endpoint

```
arch?searchVal=our%tampines%hub&returnGeom=N&getAddrDetails=Y&pageNum=1
```

Response


```
{
    "found": 1,
    "totalNumPages": 1,
    "pageNum": 1,
    "totalNumEntries": 1,
```

```
            "BUILDING": "OUR TAMPINES HUB",
            "ADDRESS": "1 TAMPINES WALK OUR TAMPINES HUB SINGAPORE 5285
            "POSTAL": "528523"
        }
]
}
```

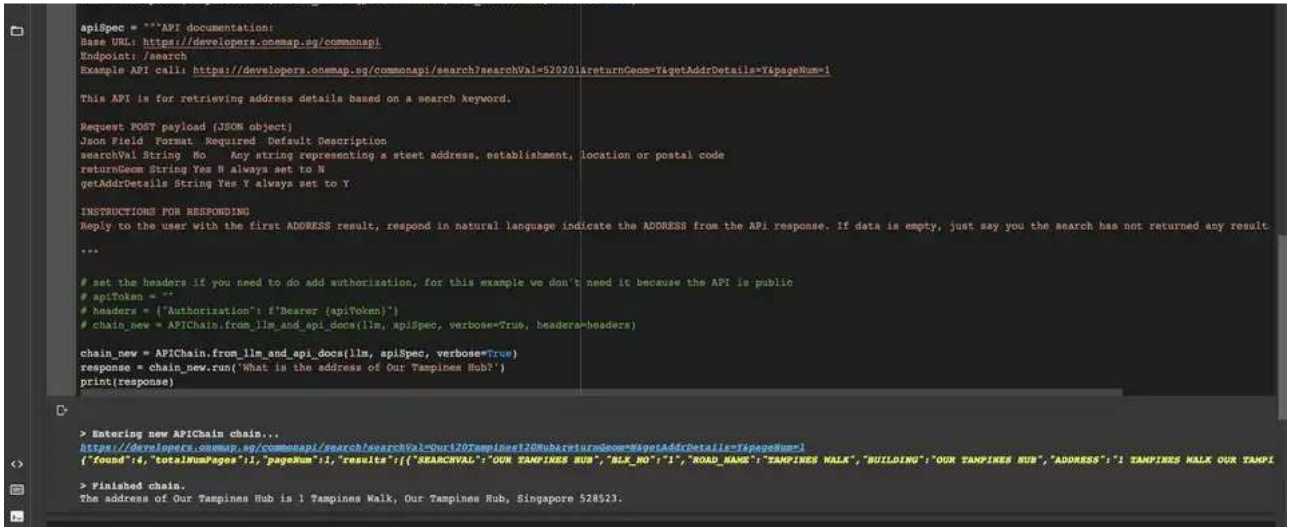
Source Code of Example

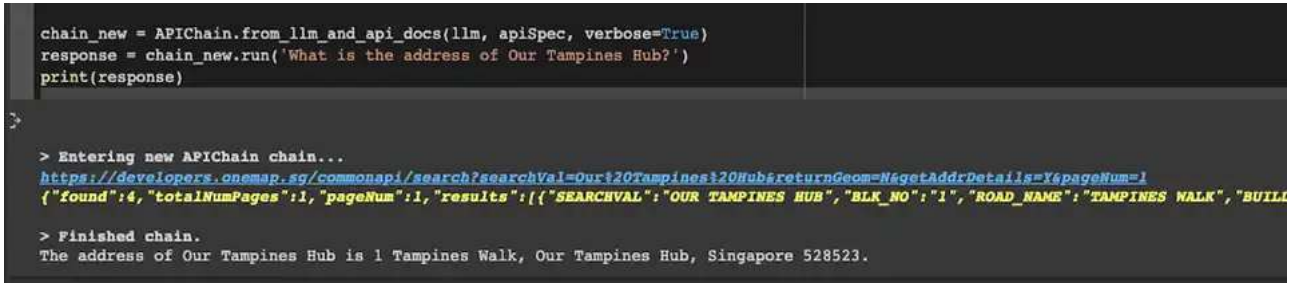
Below is the full code for the example above

You can also copy the code from my Github Gist

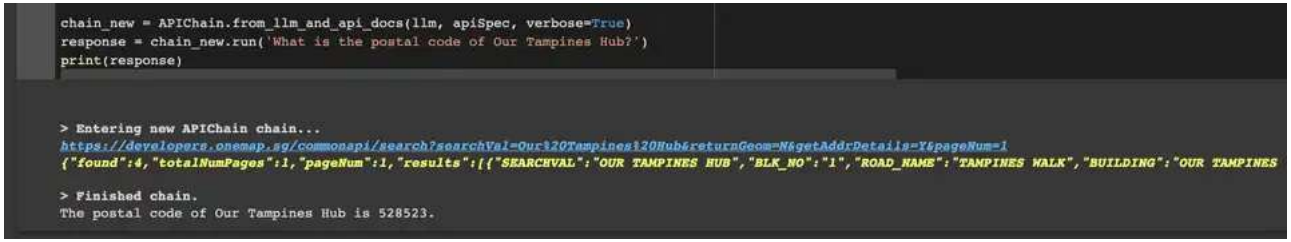
```
import os

from langchain.chains import APIChain
from langchain.chat_models import ChatOpenAI

os.environ["OPENAI_API_KEY"] = ""

llm = ChatOpenAI(temperature=0, model_name="gpt-3.5-turbo", max_tokens=
```

```
This API is for retrieving address details based on a search keyword.

Request POST payload (JSON object)
Json Field  Format  Required    Default Description
searchVal   String  No      Any string representing a steet address, es
returnGeom String Yes N always set to N
getAddrDetails String Yes Y always set to Y

INSTRUCTIONS FOR RESPONDING
Reply to the user with the first ADDRESS result, respond in natural lan

"""

# set the headers if you need to do add authorization, for this example
# apiToken = ""
# headers = {"Authorization": f"Bearer {apiToken}"}
# chain_new = APIChain.from_llm_and_api_docs(llm, apiSpec, verbose=True

chain_new = APIChain.from_llm_and_api_docs(llm, apiSpec, verbose=True)
response = chain_new.run('What is the postal code of Our Tampines Hub?'
print(response)
```


apiSpec = """API documentation:
Base URL: https://developers.onemap.sg/commonapi
Endpoint: /search
Example API call: https://developers.onemap.sg/commonapi/search?searchVal=520201&returnGeom=Y&getAddrDetails=Y&pageNum=1
This API is for retrieving address details based on a search keyword.
Request POST payload (JSON object)
Json Field Format Required Default Description
searchVal String No Any string representing a steet address, establishment, location or postal code
returnGeom String Yes N always set to N
getAddrDetails String Yes Y always set to Y
INSTRUCTIONS FOR RESPONDING
Reply to the user with the first ADDRESS result, respond in natural language indicate the ADDRESS from the API response. If data is empty, just say you the search has not returned any result
---
# set the headers if you need to do add authorization, for this example we don't need it because the API is public
# apiToken = ""
# headers = {"Authorization": f"Bearer {apiToken}"}
# chain_new = APIChain.from_llm_and_api_docs(llm, apiSpec, verbose=True, headers=headers)
chain_new = APIChain.from_llm_and_api_docs(llm, apiSpec, verbose=True)
response = chain_new.run('What is the address of Our Tampines Hub?')
print(response)
> Entering new APIChain chain...
https://developers.onemap.sg/commonapi/search?searchVal=Our%20Tampines%20Hub&returnGeom=N&getAddrDetails=Y&pageNum=1
{"found":4,"totalNumPages":1,"pageNum":1,"results":[{"SEARCHVAL":"OUR TAMPINES HUB","BLK_NO":"1","ROAD_NAME":"TAMPINES WALK","BUILDING":"OUR TAMPINES HUB","ADDRESS":"1 TAMPINES WALK OUR TAMPI
> Finished chain.
The address of Our Tampines Hub is 1 Tampines Walk, Our Tampines Hub, Singapore 528523.


chain_new = APIChain.from_llm_and_api_docs(llm, apiSpec, verbose=True)
response = chain_new.run('What is the address of Our Tampines Hub?')
print(response)
> Entering new APIChain chain...
https://developers.onemap.sg/commonapi/search?searchVal=Our%20Tampines%20Hub&returnGeom=N&getAddrDetails=Y&pageNum=1
{"found":4,"totalNumPages":1,"pageNum":1,"results":[{"SEARCHVAL":"OUR TAMPINES HUB","BLK_NO":"1","ROAD_NAME":"TAMPINES WALK","BUILD
> Finished chain.
The address of Our Tampines Hub is 1 Tampines Walk, Our Tampines Hub, Singapore 528523.

api_chain_example
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
import os
from langchain.chains import APIChain
from langchain.chat_models import ChatOpenAI
os.environ["OPENAI_API_KEY"] = ""
llm = ChatOpenAI(temperature=0, model_name="gpt-3.5-turbo", max_tokens=256, verbose=True)
apiSpec = """API documentation:
Base URL: https://developers.onemap.sg/commonapi
Endpoint: /search
Example API call: https://developers.onemap.sg/commonapi/search?searchVal=520201&returnGeom=Y&getAddrDetails=Y&pageNum=1
This API is for retrieving address details based on a search keyword.



The use of OneMap API in the previous example was just an illustration. APIChain can be used with any API that accepts natural language inputs and returns data in a structured format. The possibilities are limitless, and businesses can choose to integrate with any API that meets their specific needs.

APIChain makes it easy to integrate with multiple APIs simultaneously, allowing businesses to create custom workflows that leverage the strengths of different APIs. This not only enhances the user experience but also streamlines business operations, allowing organizations to focus on other critical aspects of their operations.

Buy me a coffee 1

.

- **E-commerce**: With APIChain, businesses could allow customers to use natural language to search for products and retrieve relevant results from their e-commerce API.
- **Finance**: Users could ask natural language questions like "What is the current stock price of Amazon?" and APIChain could use the appropriate financial API to retrieve the information and return it in a human-readable format.
- **Travel**: Users could use natural language to search for flights, hotels, and other travel-related information from various travel APIs, making the travel booking experience more personalized and intuitive.
- **Healthcare**: APIChain could be used to retrieve medical information from various healthcare APIs and provide personalized responses to user inquiries about symptoms, conditions, and treatments.
- **Education**: APIChain could be used to provide personalized learning experiences to students by integrating with various educational APIs and providing answers to questions in real-time.

Overall, APIChain has endless possibilities for businesses and developers looking to leverage the power of natural language processing and APIs to deliver personalized, intuitive experiences to their users.

Reference used

I hope this article helps you in your journey to develop your own applications with Open AI. I will find time to write more articles about Open AI and how to use it in your applications. You can reach me in my social media accounts below.

If you'd like to show your support and appreciation for my work, please consider buying me a coffee. :)

.

Your support means a lot to me and will motivate me to continue creating content that you love. Thank you for considering to support my work!

Cheers!

Melvin Dave

.

Or you can reach me in my social media accounts below.

- Twitch
- Twitter
- LinkedIn
- GitHub.

Published Apr 15, 2023

langchain openai chatgpt ai

Got feedback? Send me a message in Twitter, Github, LinkedIn or Youtube