OpenAPI chain

This notebook shows an example of using an OpenAPI chain to call an endpoint in natural language, and get back a response in natural language.

```
from langchain.tools import OpenAPISpec, APIOperation
from langchain.chains import OpenAPIEndpointChain
from langchain.requests import Requests
from langchain.llms import OpenAI
```

Load the spec

Load a wrapper of the spec (so we can work with it more easily). You can load from a url or from a local file.

```
spec = OpenAPISpec.from_url(
    "https://www.klarna.com/us/shopping/public/openai/v0/api-docs/"
)
```

```
    Attempting to load an OpenAPI 3.0.1 spec.  This may result in degraded performance. Convert your OpenAPI spec to 3.1.* spec for better support.
```

```
# Alternative loading from file
# spec = OpenAPISpec.from_file("openai_openapi.yaml")
```

Select the Operation

In order to provide a focused on modular chain, we create a chain specifically only for one of the endpoints. Here we get an API operation from a specified endpoint and method.

```
operation = APIOperation.from_openapi_spec(spec, "/public/openai/v0/products", "get")
```

Construct the chain

We can now construct a chain to interact with it. In order to construct such a chain, we will pass in:

1. The operation endpoint
2. A requests wrapper (can be used to handle authentication, etc)
3. The LLM to use to interact with it

```
llm = OpenAI()  # Load a Language Model
```

```
chain = OpenAPIEndpointChain.from_api_operation(
    operation,
    llm,
```

```
    requests=Requests(),
    verbose=True,
    return_intermediate_steps=True,  # Return request and response text
)
```

```
output = chain("whats the most expensive shirt?")
```

```
    > Entering new OpenAPIEndpointChain chain...


    > Entering new APIRequesterChain chain...
    Prompt after formatting:
    You are a helpful AI Assistant. Please provide JSON arguments to agentFunc() based on the user's
instructions.

    API_SCHEMA: ```typescript
    /* API for fetching Klarna product information */
    type productsUsingGET = (_: {
    /* A precise query that matches one very small category or product that needs to be searched for to find
the products the user is looking for. If the user explicitly stated what they want, use that as a query. The
query is as specific as possible to the product name or category mentioned by the user in its singular form,
and don't contain any clarifiers like latest, newest, cheapest, budget, premium, expensive or similar. The
query is always taken from the latest topic, if there is a new topic a new query is started. */
            q: string,
    /* number of products returned */
            size?: number,
    /* (Optional) Minimum price in local currency for the product searched for. Either explicitly stated by
the user or implicitly inferred from a combination of the user's request and the kind of product searched
```

```
for. */
            min_price?: number,
    /* (Optional) Maximum price in local currency for the product searched for. Either explicitly stated by
the user or implicitly inferred from a combination of the user's request and the kind of product searched
for. */
            max_price?: number,
    }) => any;
    ```

    USER_INSTRUCTIONS: "whats the most expensive shirt?"

    Your arguments must be plain json provided in a markdown block:

    ARGS: ```json
    {valid json conforming to API_SCHEMA}
    ```

    Example
    -----

    ARGS: ```json
    {"foo": "bar", "baz": {"qux": "quux"}}
    ```

    The block must be no more than 1 line long, and all arguments must be valid JSON. All string arguments
must be wrapped in double quotes.
    You MUST strictly comply to the types indicated by the provided schema, including all required args.

    If you don't have sufficient information to call the function due to things like requiring specific
uuid's, you can reply with the following message:

    Message: ```text
    Concise response requesting the additional information that would make calling the function successful.
```
```

```
    ```

    Begin

    ARGS:

    > Finished chain.
    {"q": "shirt", "size": 1, "max_price": null}
    {"products":[{"name":"Burberry Check Poplin
Shirt","url":"https://www.klarna.com/us/shopping/pl/cl10001/3201810981/Clothing/Burberry-Check-Poplin-Shirt/?
utm_source=openai&ref-site=openai_plugin","price":"$360.00","attributes":["Material:Cotton","Target
Group:Man","Color:Gray,Blue,Beige","Properties:Pockets","Pattern:Checkered"]}]}

    > Entering new APIResponderChain chain...
    Prompt after formatting:
    You are a helpful AI assistant trained to answer user queries from API responses.
    You attempted to call an API, which resulted in:
    API_RESPONSE: {"products":[{"name":"Burberry Check Poplin
Shirt","url":"https://www.klarna.com/us/shopping/pl/cl10001/3201810981/Clothing/Burberry-Check-Poplin-Shirt/?
utm_source=openai&ref-site=openai_plugin","price":"$360.00","attributes":["Material:Cotton","Target
Group:Man","Color:Gray,Blue,Beige","Properties:Pockets","Pattern:Checkered"]}]}

    USER_COMMENT: "whats the most expensive shirt?"

    If the API_RESPONSE can answer the USER_COMMENT respond with the following markdown json block:
    Response: ```json
    {"response": "Human-understandable synthesis of the API_RESPONSE"}
    ```

    Otherwise respond with the following markdown json block:
```

```
    Response Error: ```json
    {"response": "What you did and a concise statement of the resulting error. If it can be easily fixed,
provide a suggestion."}
    ```

    You MUST respond as a markdown json code block. The person you are responding to CANNOT see the
API_RESPONSE, so if there is any relevant information there you must include it in your response.

    Begin:

    > Finished chain.
    The most expensive shirt in the API response is the Burberry Check Poplin Shirt, which costs $360.00.

    > Finished chain.
```

```
# View intermediate steps
output["intermediate_steps"]
```

```
    {'request_args': '{"q": "shirt", "size": 1, "max_price": null}',
     'response_text': '{"products":[{"name":"Burberry Check Poplin
Shirt","url":"https://www.klarna.com/us/shopping/pl/cl10001/3201810981/Clothing/Burberry-Check-Poplin-Shirt/?
utm_source=openai&ref-site=openai_plugin","price":"$360.00","attributes":["Material:Cotton","Target
Group:Man","Color:Gray,Blue,Beige","Properties:Pockets","Pattern:Checkered"]}]}'}
```

Return raw response
```

We can also run this chain without synthesizing the response. This will have the effect of just returning the raw API output.

```
chain = OpenAPIEndpointChain.from_api_operation(
    operation,
    llm,
    requests=Requests(),
    verbose=True,
    return_intermediate_steps=True,  # Return request and response text
    raw_response=True,  # Return raw response
)
```

```
output = chain("whats the most expensive shirt?")
```

```


    > Entering new OpenAPIEndpointChain chain...


    > Entering new APIRequesterChain chain...
    Prompt after formatting:
    You are a helpful AI Assistant. Please provide JSON arguments to agentFunc() based on the user's
instructions.

    API_SCHEMA: ```typescript
    /* API for fetching Klarna product information */
    type productsUsingGET = (_: {
    /* A precise query that matches one very small category or product that needs to be searched for to find
the products the user is looking for. If the user explicitly stated what they want, use that as a query. The
query is as specific as possible to the product name or category mentioned by the user in its singular form,
and don't contain any clarifiers like latest, newest, cheapest, budget, premium, expensive or similar. The
```

```
query is always taken from the latest topic, if there is a new topic a new query is started. */
            q: string,
    /* number of products returned */
            size?: number,
    /* (Optional) Minimum price in local currency for the product searched for. Either explicitly stated by
the user or implicitly inferred from a combination of the user's request and the kind of product searched
for. */
            min_price?: number,
    /* (Optional) Maximum price in local currency for the product searched for. Either explicitly stated by
the user or implicitly inferred from a combination of the user's request and the kind of product searched
for. */
            max_price?: number,
    }) => any;
    ```

    USER_INSTRUCTIONS: "whats the most expensive shirt?"

    Your arguments must be plain json provided in a markdown block:

    ARGS: ```json
    {valid json conforming to API_SCHEMA}
    ```

    Example
    -----

    ARGS: ```json
    {"foo": "bar", "baz": {"qux": "quux"}}
    ```

    The block must be no more than 1 line long, and all arguments must be valid JSON. All string arguments
must be wrapped in double quotes.
    You MUST strictly comply to the types indicated by the provided schema, including all required args.
```
```

```
    If you don't have sufficient information to call the function due to things like requiring specific
uuid's, you can reply with the following message:

    Message: ```text
    Concise response requesting the additional information that would make calling the function successful.
    ```

    Begin

    ARGS:

    > Finished chain.
    {"q": "shirt", "max_price": null}
    {"products":[{"name":"Burberry Check Poplin
Shirt","url":"https://www.klarna.com/us/shopping/pl/cl10001/3201810981/Clothing/Burberry-Check-Poplin-Shirt/?
utm_source=openai&ref-site=openai_plugin","price":"$360.00","attributes":["Material:Cotton","Target
Group:Man","Color:Gray,Blue,Beige","Properties:Pockets","Pattern:Checkered"]},{"name":"Burberry Vintage Check
Cotton Shirt - Beige","url":"https://www.klarna.com/us/shopping/pl/cl359/3200280807/Children-s-
Clothing/Burberry-Vintage-Check-Cotton-Shirt-Beige/?utm_source=openai&ref-
site=openai_plugin","price":"$229.02","attributes":
["Material:Cotton,Elastane","Color:Beige","Model:Boy","Pattern:Checkered"]},{"name":"Burberry Vintage Check
Stretch Cotton Twill
Shirt","url":"https://www.klarna.com/us/shopping/pl/cl10001/3202342515/Clothing/Burberry-Vintage-Check-
Stretch-Cotton-Twill-Shirt/?utm_source=openai&ref-site=openai_plugin","price":"$309.99","attributes":
["Material:Elastane/Lycra/Spandex,Cotton","Target
Group:Woman","Color:Beige","Properties:Stretch","Pattern:Checkered"]},{"name":"Burberry Somerton Check Shirt
- Camel","url":"https://www.klarna.com/us/shopping/pl/cl10001/3201112728/Clothing/Burberry-Somerton-Check-
Shirt-Camel/?utm_source=openai&ref-site=openai_plugin","price":"$450.00","attributes":
["Material:Elastane/Lycra/Spandex,Cotton","Target Group:Man","Color:Beige"]},{"name":"Magellan Outdoors
Laguna Madre Solid Short Sleeve Fishing
Shirt","url":"https://www.klarna.com/us/shopping/pl/cl10001/3203102142/Clothing/Magellan-Outdoors-Laguna-
```
```

```
Madre-Solid-Short-Sleeve-Fishing-Shirt/?utm_source=openai&ref-
site=openai_plugin","price":"$19.99","attributes":["Material:Polyester,Nylon","Target
Group:Man","Color:Red,Pink,White,Blue,Purple,Beige,Black,Green","Properties:Pockets","Pattern:Solid
Color"]}]}

    > Finished chain.
```

```
output
```

```
    {'instructions': 'whats the most expensive shirt?',
     'output': '{"products":[{"name":"Burberry Check Poplin
Shirt","url":"https://www.klarna.com/us/shopping/pl/cl10001/3201810981/Clothing/Burberry-Check-Poplin-Shirt/?
utm_source=openai&ref-site=openai_plugin","price":"$360.00","attributes":["Material:Cotton","Target
Group:Man","Color:Gray,Blue,Beige","Properties:Pockets","Pattern:Checkered"]},{"name":"Burberry Vintage Check
Cotton Shirt - Beige","url":"https://www.klarna.com/us/shopping/pl/cl359/3200280807/Children-s-
Clothing/Burberry-Vintage-Check-Cotton-Shirt-Beige/?utm_source=openai&ref-
site=openai_plugin","price":"$229.02","attributes":
["Material:Cotton,Elastane","Color:Beige","Model:Boy","Pattern:Checkered"]},{"name":"Burberry Vintage Check
Stretch Cotton Twill
Shirt","url":"https://www.klarna.com/us/shopping/pl/cl10001/3202342515/Clothing/Burberry-Vintage-Check-
Stretch-Cotton-Twill-Shirt/?utm_source=openai&ref-site=openai_plugin","price":"$309.99","attributes":
["Material:Elastane/Lycra/Spandex,Cotton","Target
Group:Woman","Color:Beige","Properties:Stretch","Pattern:Checkered"]},{"name":"Burberry Somerton Check Shirt
- Camel","url":"https://www.klarna.com/us/shopping/pl/cl10001/3201112728/Clothing/Burberry-Somerton-Check-
Shirt-Camel/?utm_source=openai&ref-site=openai_plugin","price":"$450.00","attributes":
["Material:Elastane/Lycra/Spandex,Cotton","Target Group:Man","Color:Beige"]},{"name":"Magellan Outdoors
Laguna Madre Solid Short Sleeve Fishing
Shirt","url":"https://www.klarna.com/us/shopping/pl/cl10001/3203102142/Clothing/Magellan-Outdoors-Laguna-
Madre-Solid-Short-Sleeve-Fishing-Shirt/?utm_source=openai&ref-
site=openai_plugin","price":"$19.99","attributes":["Material:Polyester,Nylon","Target
Group:Man","Color:Red,Pink,White,Blue,Purple,Beige,Black,Green","Properties:Pockets","Pattern:Solid
```

```
Color"]}]}',
     'intermediate_steps': {'request_args': '{"q": "shirt", "max_price": null}',
      'response_text': '{"products":[{"name":"Burberry Check Poplin
Shirt","url":"https://www.klarna.com/us/shopping/pl/cl10001/3201810981/Clothing/Burberry-Check-Poplin-Shirt/?
utm_source=openai&ref-site=openai_plugin","price":"$360.00","attributes":["Material:Cotton","Target
Group:Man","Color:Gray,Blue,Beige","Properties:Pockets","Pattern:Checkered"]},{"name":"Burberry Vintage Check
Cotton Shirt - Beige","url":"https://www.klarna.com/us/shopping/pl/cl359/3200280807/Children-s-
Clothing/Burberry-Vintage-Check-Cotton-Shirt-Beige/?utm_source=openai&ref-
site=openai_plugin","price":"$229.02","attributes":
["Material:Cotton,Elastane","Color:Beige","Model:Boy","Pattern:Checkered"]},{"name":"Burberry Vintage Check
Stretch Cotton Twill
Shirt","url":"https://www.klarna.com/us/shopping/pl/cl10001/3202342515/Clothing/Burberry-Vintage-Check-
Stretch-Cotton-Twill-Shirt/?utm_source=openai&ref-site=openai_plugin","price":"$309.99","attributes":
["Material:Elastane/Lycra/Spandex,Cotton","Target
Group:Woman","Color:Beige","Properties:Stretch","Pattern:Checkered"]},{"name":"Burberry Somerton Check Shirt
- Camel","url":"https://www.klarna.com/us/shopping/pl/cl10001/3201112728/Clothing/Burberry-Somerton-Check-
Shirt-Camel/?utm_source=openai&ref-site=openai_plugin","price":"$450.00","attributes":
["Material:Elastane/Lycra/Spandex,Cotton","Target Group:Man","Color:Beige"]},{"name":"Magellan Outdoors
Laguna Madre Solid Short Sleeve Fishing
Shirt","url":"https://www.klarna.com/us/shopping/pl/cl10001/3203102142/Clothing/Magellan-Outdoors-Laguna-
Madre-Solid-Short-Sleeve-Fishing-Shirt/?utm_source=openai&ref-
site=openai_plugin","price":"$19.99","attributes":["Material:Polyester,Nylon","Target
Group:Man","Color:Red,Pink,White,Blue,Purple,Beige,Black,Green","Properties:Pockets","Pattern:Solid
Color"]}]}'}}
```


Example POST message

For this demo, we will interact with the speak API.

```
spec = OpenAPISpec.from_url("https://api.speak.com/openapi.yaml")
```

```
    Attempting to load an OpenAPI 3.0.1 spec.  This may result in degraded performance. Convert your OpenAPI
spec to 3.1.* spec for better support.
    Attempting to load an OpenAPI 3.0.1 spec.  This may result in degraded performance. Convert your OpenAPI
spec to 3.1.* spec for better support.
```

```
operation = APIOperation.from_openapi_spec(
    spec, "/v1/public/openai/explain-task", "post"
)
```

```
llm = OpenAI()
chain = OpenAPIEndpointChain.from_api_operation(
    operation, llm, requests=Requests(), verbose=True, return_intermediate_steps=True
)
```

```
output = chain("How would ask for more tea in Delhi?")
```

```


    > Entering new OpenAPIEndpointChain chain...


    > Entering new APIRequesterChain chain...
    Prompt after formatting:
    You are a helpful AI Assistant. Please provide JSON arguments to agentFunc() based on the user's
```

```
instructions.

    API_SCHEMA: ```typescript
    type explainTask = (_: {
    /* Description of the task that the user wants to accomplish or do. For example, "tell the waiter they
messed up my order" or "compliment someone on their shirt" */
      task_description?: string,
    /* The foreign language that the user is learning and asking about. The value can be inferred from
question - for example, if the user asks "how do i ask a girl out in mexico city", the value should be
"Spanish" because of Mexico City. Always use the full name of the language (e.g. Spanish, French). */
      learning_language?: string,
    /* The user's native language. Infer this value from the language the user asked their question in.
Always use the full name of the language (e.g. Spanish, French). */
      native_language?: string,
    /* A description of any additional context in the user's question that could affect the explanation -
e.g. setting, scenario, situation, tone, speaking style and formality, usage notes, or any other qualifiers.
*/
      additional_context?: string,
    /* Full text of the user's question. */
      full_query?: string,
    }) => any;
    ```

    USER_INSTRUCTIONS: "How would ask for more tea in Delhi?"

    Your arguments must be plain json provided in a markdown block:

    ARGS: ```json
    {valid json conforming to API_SCHEMA}
    ```

    Example
    -----
```

```
    ARGS: ```json
    {"foo": "bar", "baz": {"qux": "quux"}}
    ```

    The block must be no more than 1 line long, and all arguments must be valid JSON. All string arguments must be wrapped in double quotes.
    You MUST strictly comply to the types indicated by the provided schema, including all required args.

    If you don't have sufficient information to call the function due to things like requiring specific uuid's, you can reply with the following message:

    Message: ```text
    Concise response requesting the additional information that would make calling the function successful.
    ```

    Begin
    -----
    ARGS:



    > Finished chain.
    {"task_description": "ask for more tea", "learning_language": "Hindi", "native_language": "English", "full_query": "How would I ask for more tea in Delhi?"}
    {"explanation":"<what-to-say language=\"Hindi\" context=\"None\">\nऔर चाय लाओ। (Aur chai lao.) \n</what-to-say>\n\n<alternatives context=\"None\">\n1. \"चाय थोड़ी ज्यादा मिल सकती है?\" *(Chai thodi zyada mil sakti hai? - Polite, asking if more tea is available)*\n2. \"मुझे महसूस हो रहा है कि मुझे कुछ अन्य प्रकार की चाय पीनी चाहिए। \" *(Mujhe mehsoos ho raha hai ki mujhe kuch anya prakar ki chai peeni chahiye. - Formal, indicating a desire for a different type of tea)*\n3. \"क्या मुझे or cup में milk/tea powder मिल सकता है?\" *(Kya mujhe aur cup mein milk/tea powder mil sakta hai? - Very informal/casual tone, asking for an extra serving of milk or tea powder)*\n</alternatives>\n\n<usage-notes>\nIn India and Indian culture, serving guests with food and beverages holds great importance in hospitality. You will find people always offering drinks like water or tea to their guests as soon as they arrive at their house or office.\n</usage-notes>\n\n<example-convo
```

```
language=\"Hindi\">\n<context>At home during breakfast.</context>\nPreeti: सर, क्या main aur cups chai lekar aaun? (Sir,kya main aur cups chai lekar aaun? - Sir, should I get more tea cups?)\nRahul: हां,बिल्कुल। और चाय की मात्रा में भी थोड़ा सा इजाफा करना। (Haan,bilkul. Aur chai ki matra mein bhi thoda sa eejafa karna. - Yes, please. And add a little extra in the quantity of tea as well.)\n</example-convo>\n\n*[Report an issue or leave feedback](https://speak.com/chatgpt?rid=d4mcapbkopo164pqpbk321oc})*","extra_response_instructions":"Use all information in the API response and fully render all Markdown.\nAlways end your response with a link to report an issue or leave feedback on the plugin."}


    > Entering new APIResponderChain chain...
    Prompt after formatting:
    You are a helpful AI assistant trained to answer user queries from API responses.
    You attempted to call an API, which resulted in:
    API_RESPONSE: {"explanation":"<what-to-say language=\"Hindi\" context=\"None\">\nऔर चाय लाओ। (Aur chai lao.) \n</what-to-say>\n\n<alternatives context=\"None\">\n1. \"चाय थोड़ी ज्यादा मिल सकती है?\" *(Chai thodi zyada mil sakti hai? - Polite, asking if more tea is available)*\n2. \"मुझे महसूस हो रहा है कि मुझे कुछ अन्य प्रकार की चाय पीनी चाहिए।\" *(Mujhe mehsoos ho raha hai ki mujhe kuch anya prakar ki chai peeni chahiye. - Formal, indicating a desire for a different type of tea)*\n3. \"क्या मुझे or cup में milk/tea powder मिल सकता है?\" *(Kya mujhe aur cup mein milk/tea powder mil sakta hai? - Very informal/casual tone, asking for an extra serving of milk or tea powder)*\n</alternatives>\n\n<usage-notes>\nIn India and Indian culture, serving guests with food and beverages holds great importance in hospitality. You will find people always offering drinks like water or tea to their guests as soon as they arrive at their house or office.\n</usage-notes>\n\n<example-convo language=\"Hindi\">\n<context>At home during breakfast.</context>\nPreeti: सर, क्या main aur cups chai lekar aaun? (Sir,kya main aur cups chai lekar aaun? - Sir, should I get more tea cups?)\nRahul: हां,बिल्कुल। और चाय की मात्रा में भी थोड़ा सा इजाफा करना। (Haan,bilkul. Aur chai ki matra mein bhi thoda sa eejafa karna. - Yes, please. And add a little extra in the quantity of tea as well.)\n</example-convo>\n\n*[Report an issue or leave feedback](https://speak.com/chatgpt?rid=d4mcapbkopo164pqpbk321oc})*","extra_response_instructions":"Use all information in the API response and fully render all Markdown.\nAlways end your response with a link to report an issue or leave feedback on the plugin."}

    USER_COMMENT: "How would ask for more tea in Delhi?"
```

```
    If the API_RESPONSE can answer the USER_COMMENT respond with the following markdown json block:
    Response: ```json
    {"response": "Concise response to USER_COMMENT based on API_RESPONSE."}
    ```

    Otherwise respond with the following markdown json block:
    Response Error: ```json
    {"response": "What you did and a concise statement of the resulting error. If it can be easily fixed, provide a suggestion."}
    ```

    You MUST respond as a markdown json code block.

    Begin:
    ---



    > Finished chain.
    In Delhi you can ask for more tea by saying 'Chai thodi zyada mil sakti hai?'

    > Finished chain.
```

```
# Show the API chain's intermediate steps
output["intermediate_steps"]
```

```
    ['{"task_description": "ask for more tea", "learning_language": "Hindi", "native_language": "English", "full_query": "How would I ask for more tea in Delhi?"}',
     '{"explanation":"<what-to-say language=\\"Hindi\\" context=\\"None\\">\\nऔर चाय लाओ। (Aur chai lao.) \\n</what-to-say>\\n\\n<alternatives context=\\"None\\">\\n1. \\"चाय थोड़ी ज्यादा मिल सकती है?\\" *(Chai thodi zyada mil sakti hai? - Polite, asking if more tea is available)*\\n2. \\"मुझे महसूस हो रहा है कि मुझे कुछ अन्य प्रकार की चाय पीनी चाहिए।\\" *(Mujhe mehsoos ho raha hai ki mujhe kuch anya prakar ki chai peeni chahiye. - Formal,
```

indicating a desire for a different type of tea)*\\n3. \\"क्या मुझे or cup में milk/tea powder मिल सकता है?\\" * (Kya mujhe aur cup mein milk/tea powder mil sakta hai? - Very informal/casual tone, asking for an extra serving of milk or tea powder)*\\n</alternatives>\\n\\n<usage-notes>\\nIn India and Indian culture, serving guests with food and beverages holds great importance in hospitality. You will find people always offering drinks like water or tea to their guests as soon as they arrive at their house or office.\\n</usage-notes>\\n\\n<example-convo language=\\"Hindi\\">\\n<context>At home during breakfast.</context>\\nPreeti: सर, क्या main aur cups chai lekar aaun? (Sir,kya main aur cups chai lekar aaun? - Sir, should I get more tea cups?)\\nRahul: हां,बिल्कुल। और चाय की मात्रा में भी थोड़ा सा इजाफा करना। (Haan,bilkul. Aur chai ki matra mein bhi thoda sa eejafa karna. - Yes, please. And add a little extra in the quantity of tea as well.)\\n</example-convo>\\n\\n*[Report an issue or leave feedback](https://speak.com/chatgpt?rid=d4mcapbkopo164pqpbk321oc})*","extra_response_instructions":"Use all information in the API response and fully render all Markdown.\\nAlways end your response with a link to report an issue or leave feedback on the plugin."}']