API

Learn how to use the Flowise Prediction, Vector Upsert and Message API

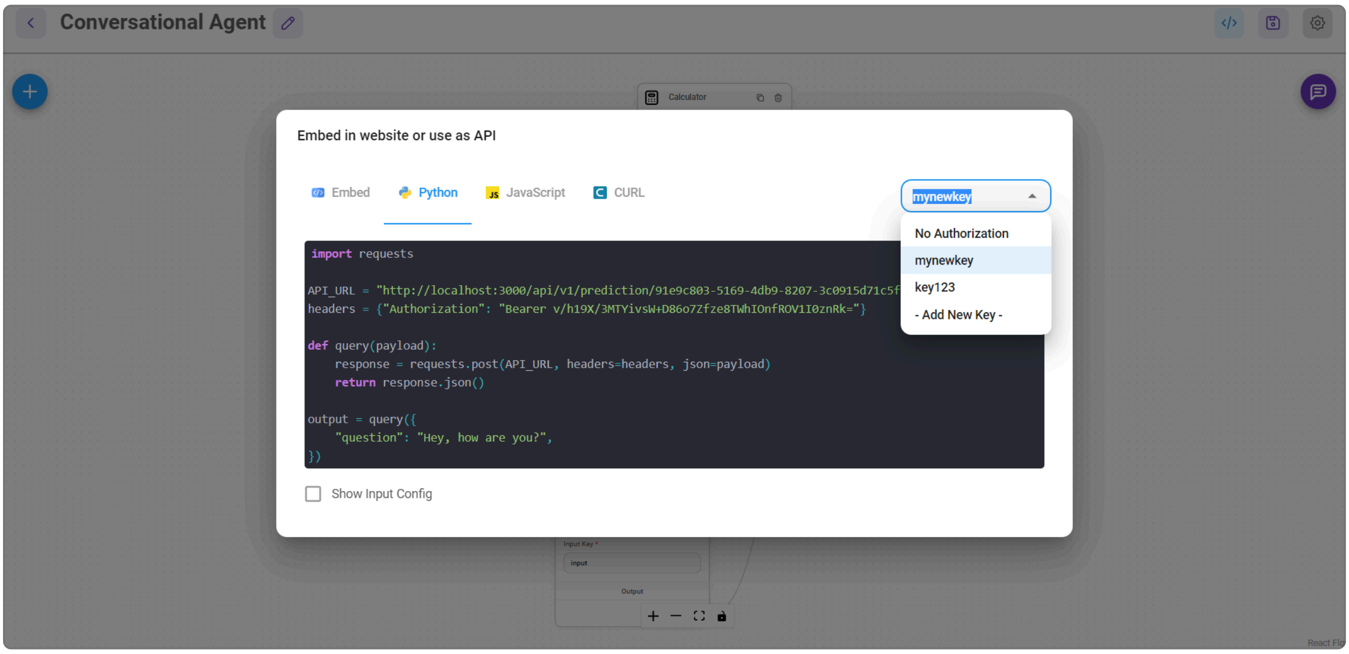
1. Prediction API

- POST `/api/v1/prediction/{your-chatflowid}`

Request Body

Key	Description	Type	Required
question	User's question	string	Yes
overrideConfig	Override existing flow configuration	object	No
history	Prepend history messages at the start of conversation	array	No

You can use the chatflow as API and connect to frontend applications.

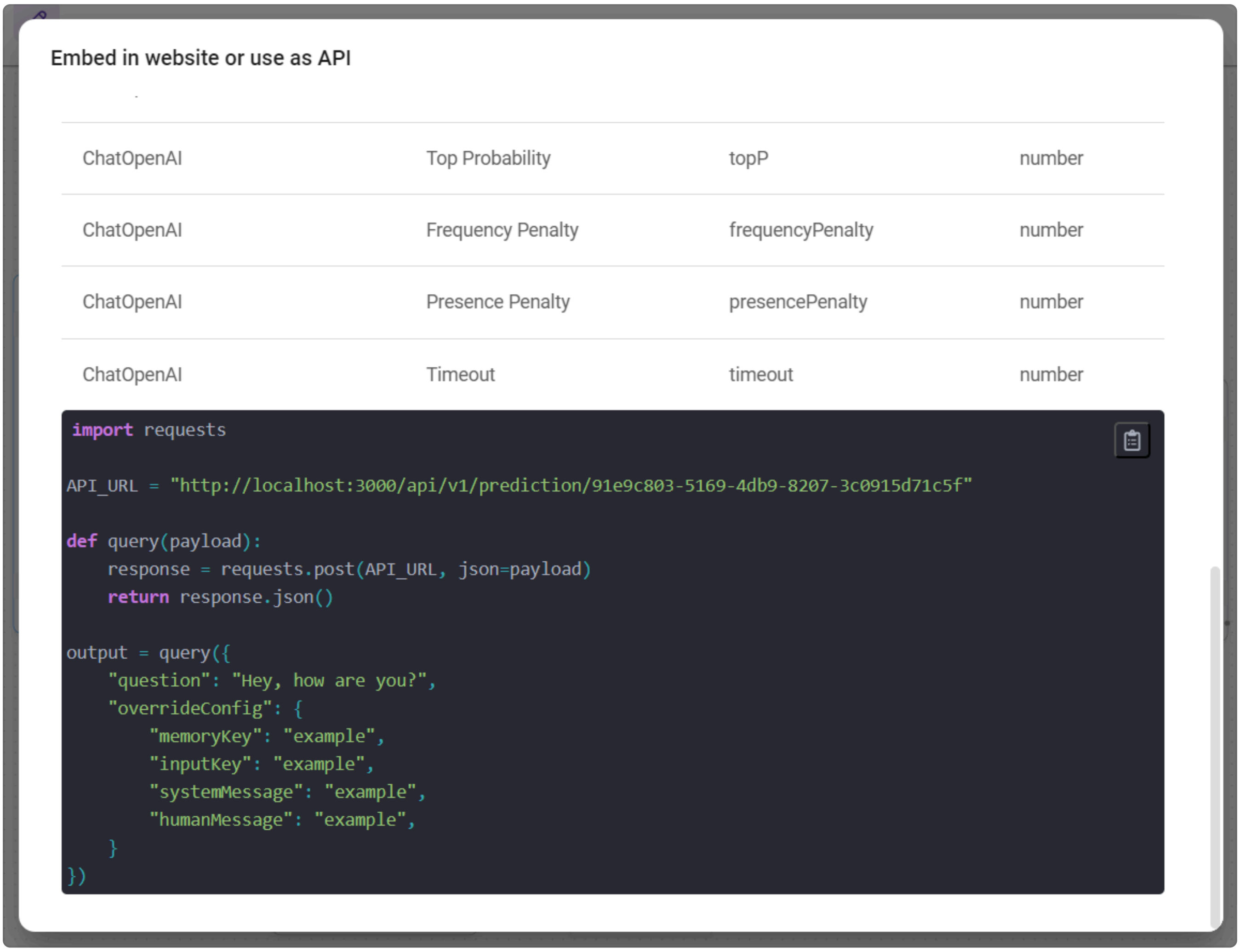


```python
import requests

API_URL = "http://localhost:3000/api/v1/prediction/91e9c803-5169-4db9-8207-3c0915d71c5f
headers = {"Authorization": "Bearer v/h19X/3MTYivsW+D86o7Zfze8TWhIOnfROV1I0znRk="}

def query(payload):
    response = requests.post(API_URL, headers=headers, json=payload)
    return response.json()

output = query({
    "question": "Hey, how are you?",
})
```

□ Show Input Config

Override Config

You also have the flexibility to override input configuration with **overrideConfig** property.



Embed in website or use as API

ChatOpenAI	Top Probability	topP	number
ChatOpenAI	Frequency Penalty	frequencyPenalty	number
ChatOpenAI	Presence Penalty	presencePenalty	number
ChatOpenAI	Timeout	timeout	number

```python
import requests

API_URL = "http://localhost:3000/api/v1/prediction/91e9c803-5169-4db9-8207-3c0915d71c5f"

def query(payload):
    response = requests.post(API_URL, json=payload)
    return response.json()

output = query({
    "question": "Hey, how are you?",
    "overrideConfig": {
        "memoryKey": "example",
        "inputKey": "example",
        "systemMessage": "example",
        "humanMessage": "example",
    }
})
```

Python Javascript

```python
import requests
API_URL = "http://localhost:3000/api/v1/prediction/&#x3C;chatlfowid>"

def query(payload):
    response = requests.post(API_URL, json=payload)
    return response.json()

output = query({
    "question": "Hey, how are you?",
    "overrideConfig": {
        "sessionId": "123",
        "returnSourceDocuments": true
    }
})
```

History

You can prepend history messages to give some context to LLM. For example, if you want the LLM to remember user's name:

Python Javascript

```
import requests
API_URL = "http://localhost:3000/api/v1/prediction/&#x3C;chatlfowid>"

def query(payload):
    response = requests.post(API_URL, json=payload)
    return response.json()

output = query({
    "question": "Hey, how are you?",
    "history": [
        {
            "role": "apiMessage",
            "content": "Hello how can I help?"
        },
        {
            "role": "userMessage",
            "content": "Hi my name is Brian"
        },
        {
            "role": "apiMessage",
            "content": "Hi Brian, how can I help?"
        },
    ]
})
```

Persists Memory

If the chatflow contains Memory nodes, you can pass a `sessionId` to persists the state of the conversation, so the every subsequent API calls will have context about previous conversation. Otherwise, a new session will be generated each time.

Python Javascript

```python
import requests
API_URL = "http://localhost:3000/api/v1/prediction/&#x3C;chatlfowid>"

def query(payload):
    response = requests.post(API_URL, json=payload)
    return response.json()

output = query({
    "question": "Hey, how are you?",
    "overrideConfig": {
        "sessionId": "123"
    }
})
```

Variables

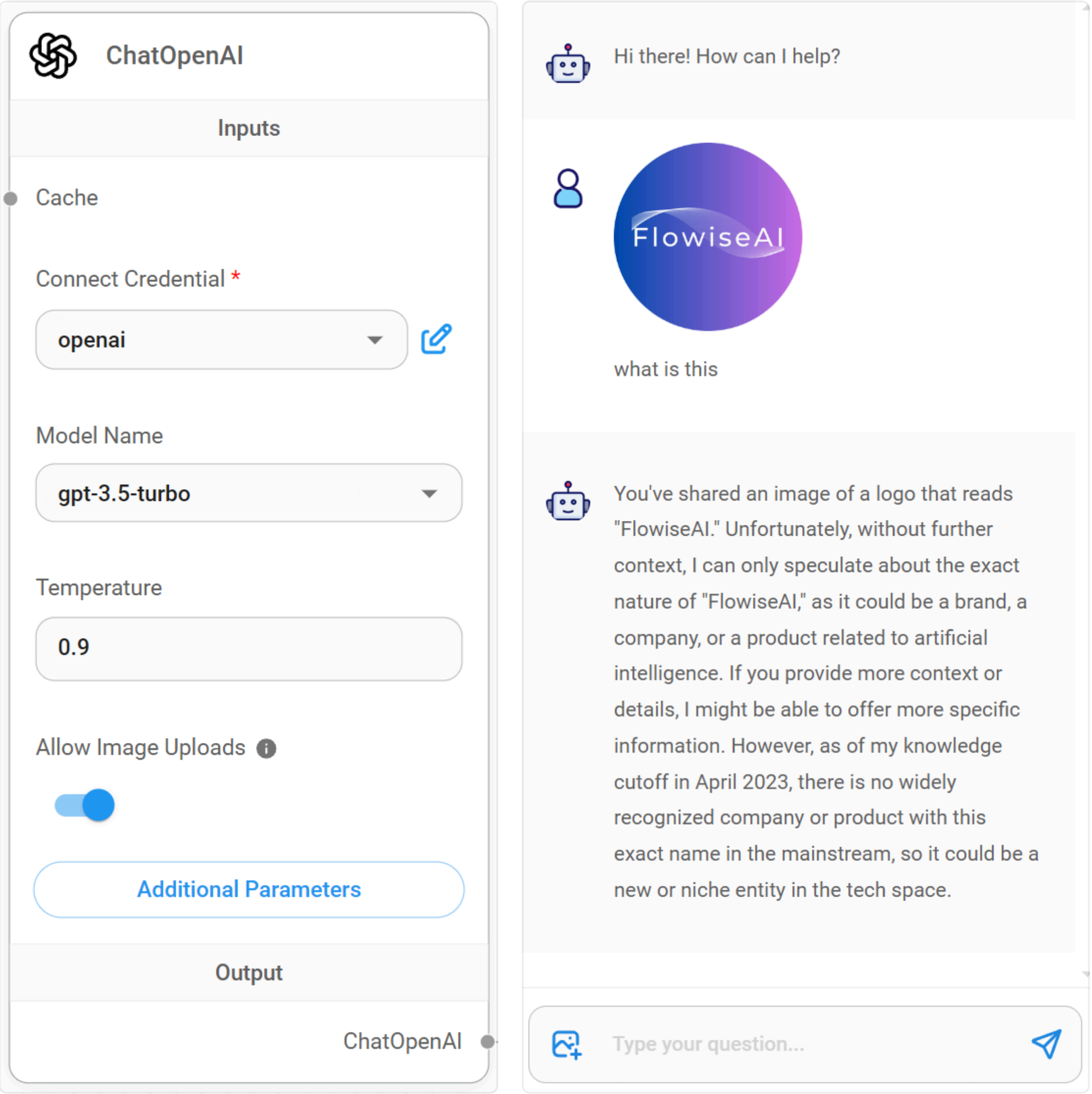
You can pass variable in the API to be used by the nodes in the flow. See more: Variables

Python Javascript

```python
import requests
API_URL = "http://localhost:3000/api/v1/prediction/&#x3C;chatlfowid>"

def query(payload):
    response = requests.post(API_URL, json=payload)
    return response.json()

output = query({
    "question": "Hey, how are you?",
    "overrideConfig": {
        "vars": {
            "foo": "bar"
        }
    }
})
```

Image Uploads

When **Allow Image Upload** is enabled, images can be uploaded from chat interface.



ChatOpenAI

Inputs

Cache

Connect Credential *

openai

Model Name

gpt-3.5-turbo

Temperature

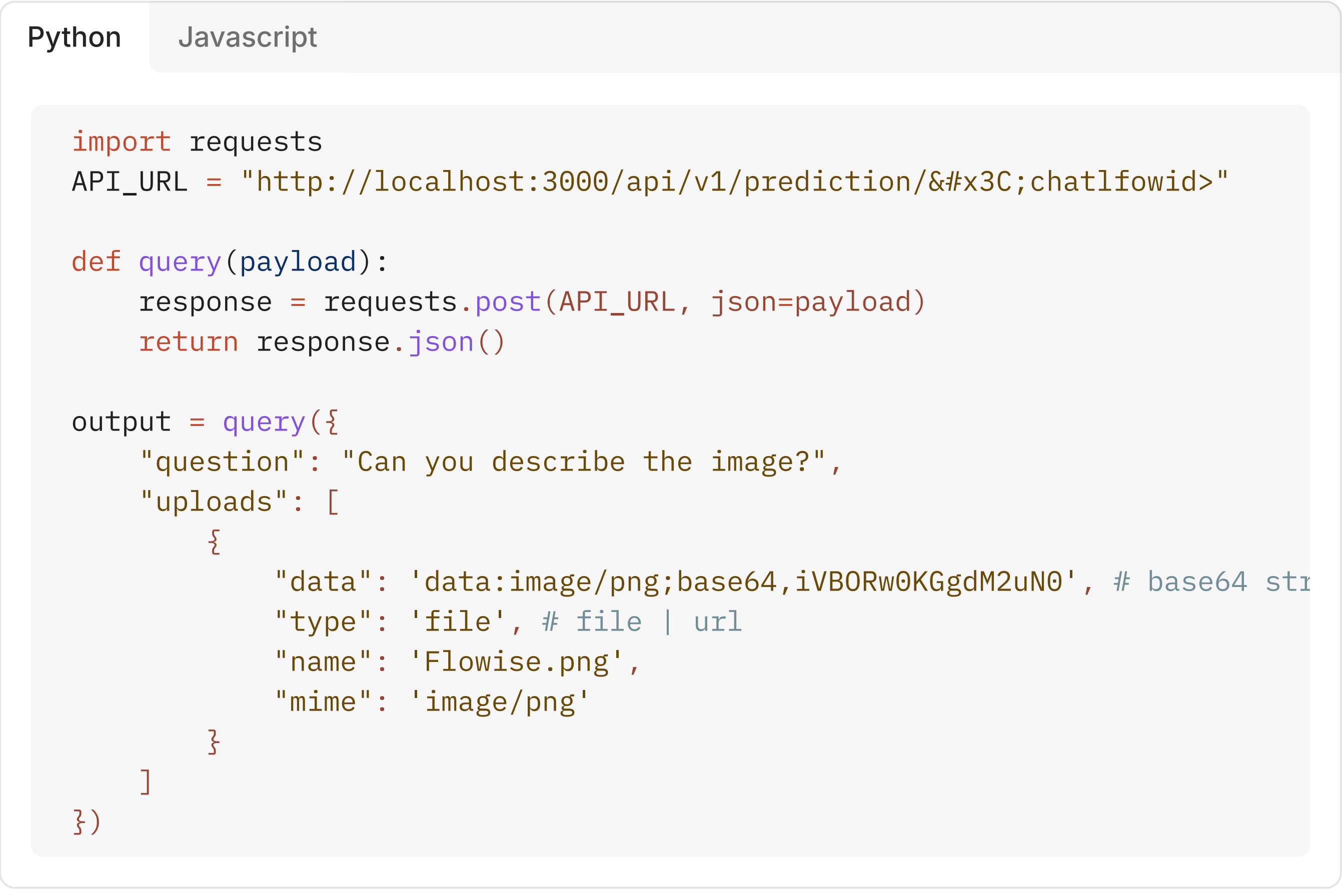
0.9

Allow Image Uploads

Hi there! How can I help?

what is this

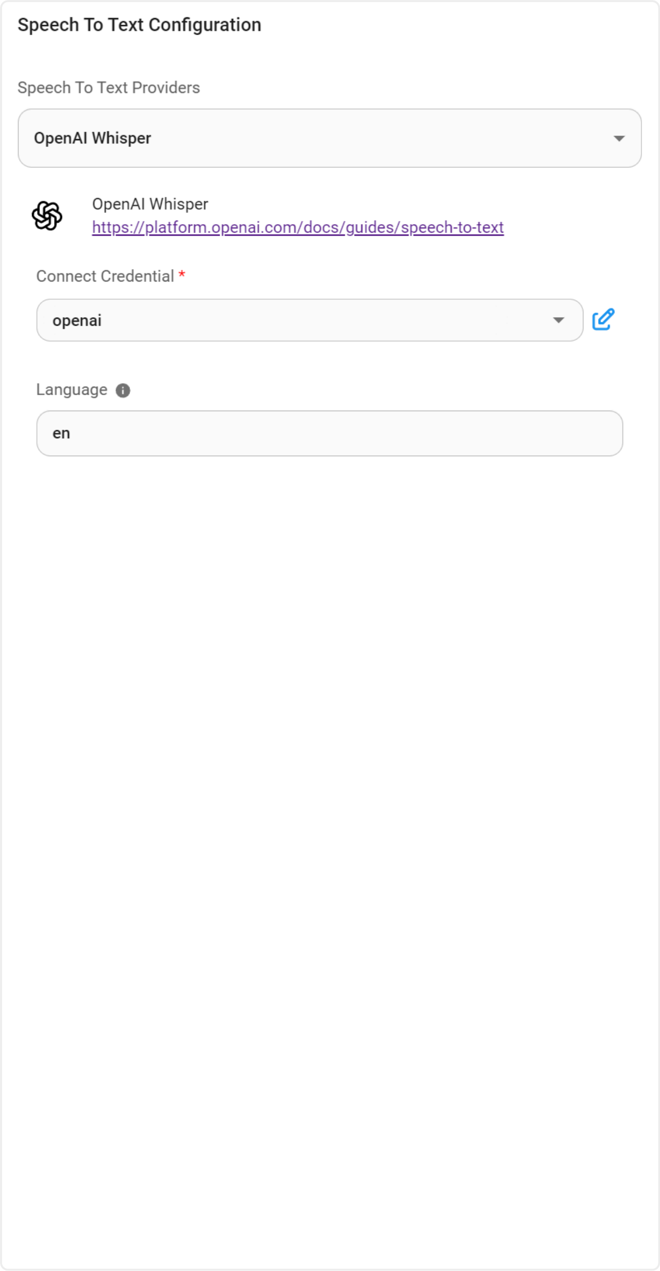
You've shared an image of a logo that reads "FlowiseAI." Unfortunately, without further context, I can only speculate about the exact nature of "FlowiseAI," as it could be a brand, a company, or a product related to artificial intelligence. If you provide more context or details, I might be able to offer more specific information. However, as of my knowledge cutoff in April 2023, there is no widely recognized company or product with this exact name in the mainstream, so it could be a new or niche entity in the tech space.



```python
import requests
API_URL = "http://localhost:3000/api/v1/prediction/&#x3C;chatlfowid>"

def query(payload):
    response = requests.post(API_URL, json=payload)
    return response.json()

output = query({
    "question": "Can you describe the image?",
    "uploads": [
        {
            "data": 'data:image/png;base64,iVBORw0KGgdM2uN0', # base64 str
            "type": 'file', # file | url
            "name": 'Flowise.png',
            "mime": 'image/png'
        }
    ]
})
```

Speech to Text

When **Speech to Text** is enabled, users can speak directly into microphone and speech will be transcribed into text.



Speech To Text Configuration

Speech To Text Providers

OpenAI Whisper

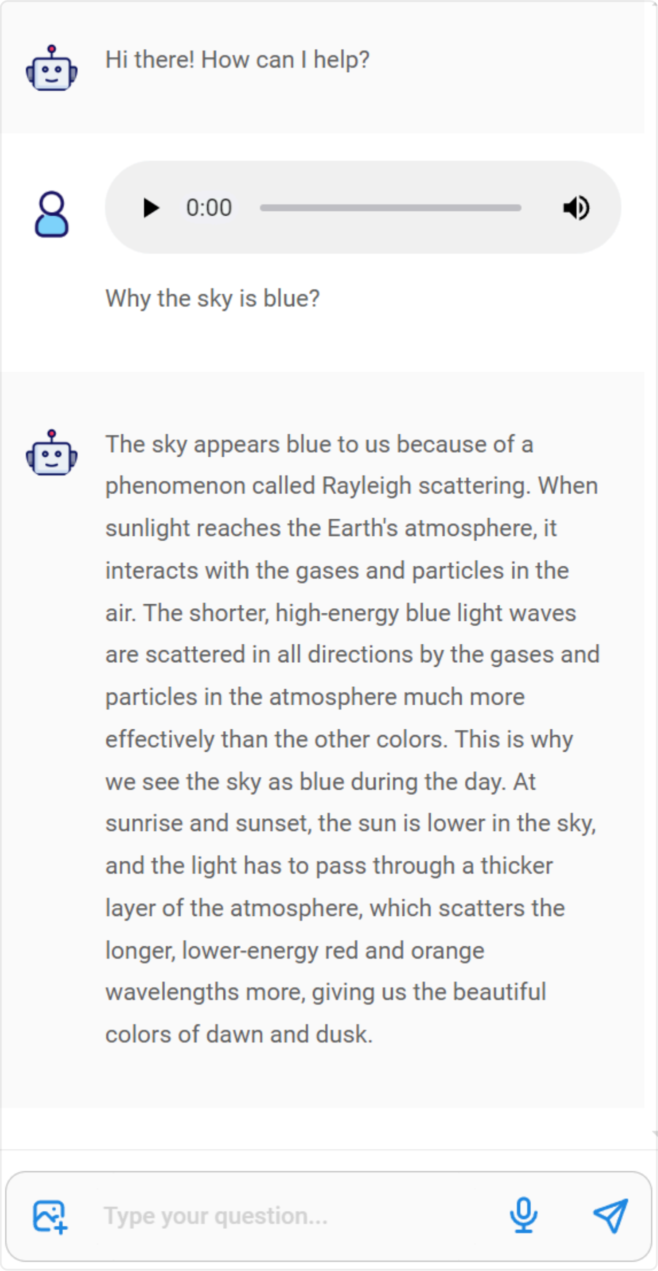
OpenAI Whisper
https://platform.openai.com/docs/guides/speech-to-text

Connect Credential *

openai

Language

en



Hi there! How can I help?

Why the sky is blue?

The sky appears blue to us because of a phenomenon called Rayleigh scattering. When sunlight reaches the Earth's atmosphere, it interacts with the gases and particles in the air. The shorter, high-energy blue light waves are scattered in all directions by the gases and particles in the atmosphere much more effectively than the other colors. This is why we see the sky as blue during the day. At sunrise and sunset, the sun is lower in the sky, and the light has to pass through a thicker layer of the atmosphere, which scatters the longer, lower-energy red and orange wavelengths more, giving us the beautiful colors of dawn and dusk.

Python Javascript

```python
import requests
API_URL = "http://localhost:3000/api/v1/prediction/&#x3C;chatlfowid>"

def query(payload):
    response = requests.post(API_URL, json=payload)
    return response.json()

output = query({
    "uploads": [
        {
            "data": 'data:audio/webm;codecs=opus;base64,GkXf', #base64 str
            "type": 'audio',
            "name": 'audio.wav',
            "mime": 'audio/webm'
        }
    ]
})
```

Authentication

You can assign an API key to the prediction API from the UI. Refer Chatflow Level for more details.

The Authorization header must be provided with the correct API key specified during a HTTP call.

```
"Authorization": "Bearer <your-api-key>"
```

2. Vector Upsert API

- POST `/api/v1/vector/upsert/{your-chatflowid}`

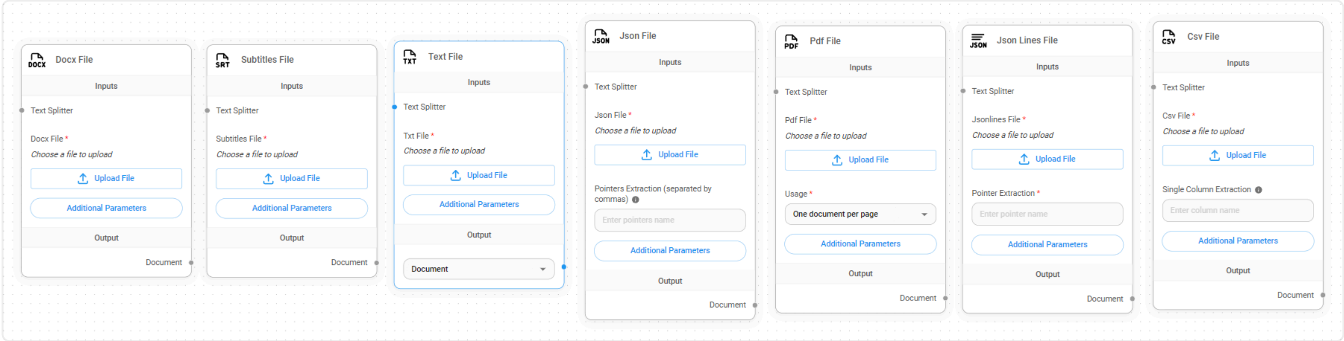
Request Body

Key	Description	Type	Required
overrideConfig	Override existing flow configuration	object	No
stopNodeId	Node ID of the vector store. When you have multiple vector stores in a flow, you might not want to upsert all of them. Specifying	array	No

Key	Description	Type	Required
	`stopNodeId` will ensure only that specific vector store node is upserted.		

Document Loaders with Upload

Some document loaders in Flowise allow user to upload files:



If the flow contains Document Loaders with Upload File functionality, the API looks slightly different. Instead of passing body as **JSON**, **form-data** is being used. This allows you to upload any files to the API.

> ⓘ It is user's responsibility to make sure the file type is compatible with the expected file type from document loader. For example, if a Text File Loader is being used, you should only upload file with `.txt` extension.

Python | Javascript

```
import requests

API_URL = "http://localhost:3000/api/v1/vector/upsert/<chatlfowid>"

# use form data to upload files
form_data = {
    "files": ('state_of_the_union.txt', open('state_of_the_union.txt', 'rb
}

body_data = {
    "returnSourceDocuments": True
}

def query(form_data):
    response = requests.post(API_URL, files=form_data, data=body_data)
    print(response)
    return response.json()

output = query(form_data)
print(output)
```

Document Loaders without Upload

For other Document Loaders nodes without Upload File functionality, the API body is in **JSON** format similar to Prediction API.

Python Javascript

```
import requests

API_URL = "http://localhost:3000/api/v1/vector/upsert/<chatlfowid>"

def query(form_data):
    response = requests.post(API_URL, json=payload)
    print(response)
    return response.json()

output = query({
    "overrideConfig": { # optional
        "returnSourceDocuments": true
    }
})
print(output)
```

Authentication

You can assign an API key to the prediction API from the UI. Refer Chatflow Level for more details.

The Authorization header must be provided with the correct API key specified during a HTTP call.

```
"Authorization": "Bearer <your-api-key>"
```

3. Message API

- GET `/api/v1/chatmessage/{your-chatflowid}`
- DELETE `/api/v1/chatmessage/{your-chatflowid}`

Query Parameters

Param	Type	Value
sessionId	string	
sort	enum	ASC or DESC
startDate	string	
endDate	string	

Authentication

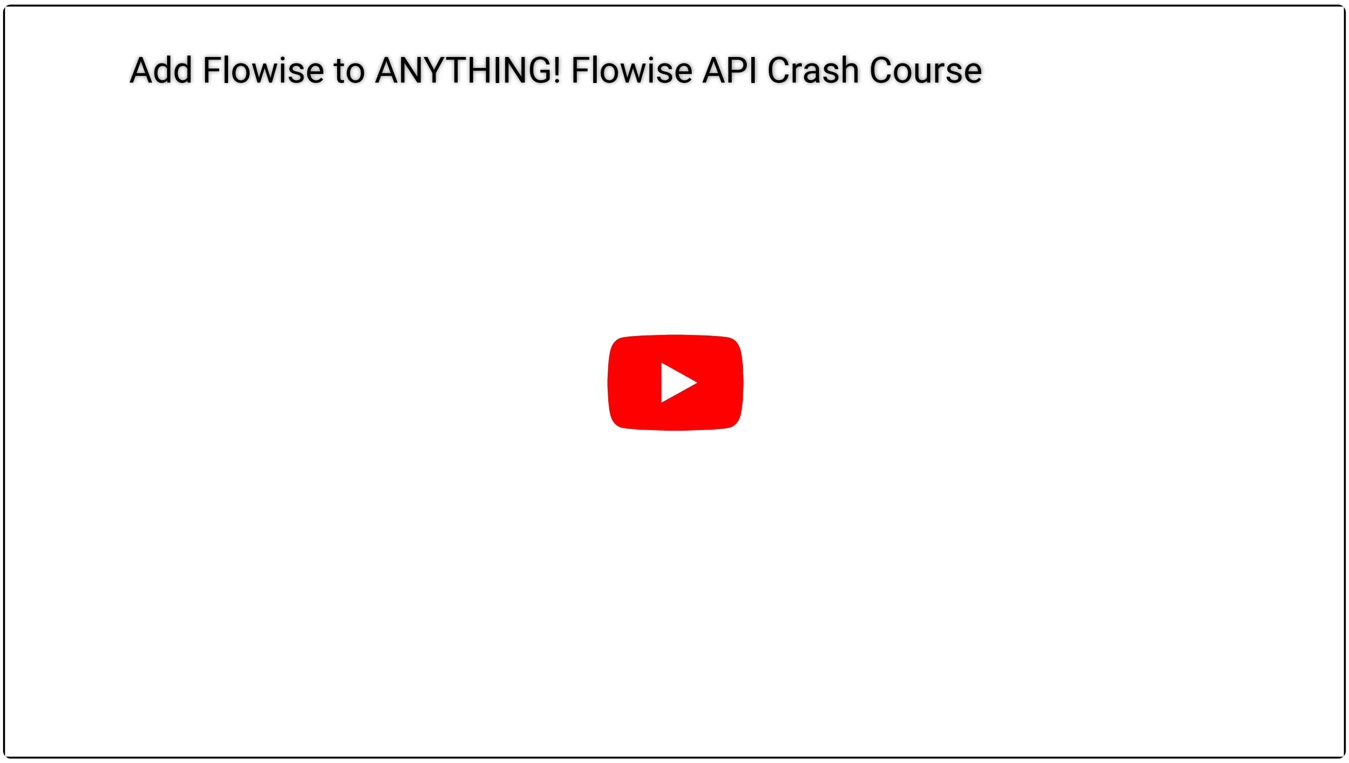
Message API is restricted to only Flowise admin user. Basic authentication must be provided in the headers if Flowise instance has been configured with `FLOWISE_USERNAME` and `FLOWISE_PASSWORD`. Refer App Level for more details.

Node | Browser | Python

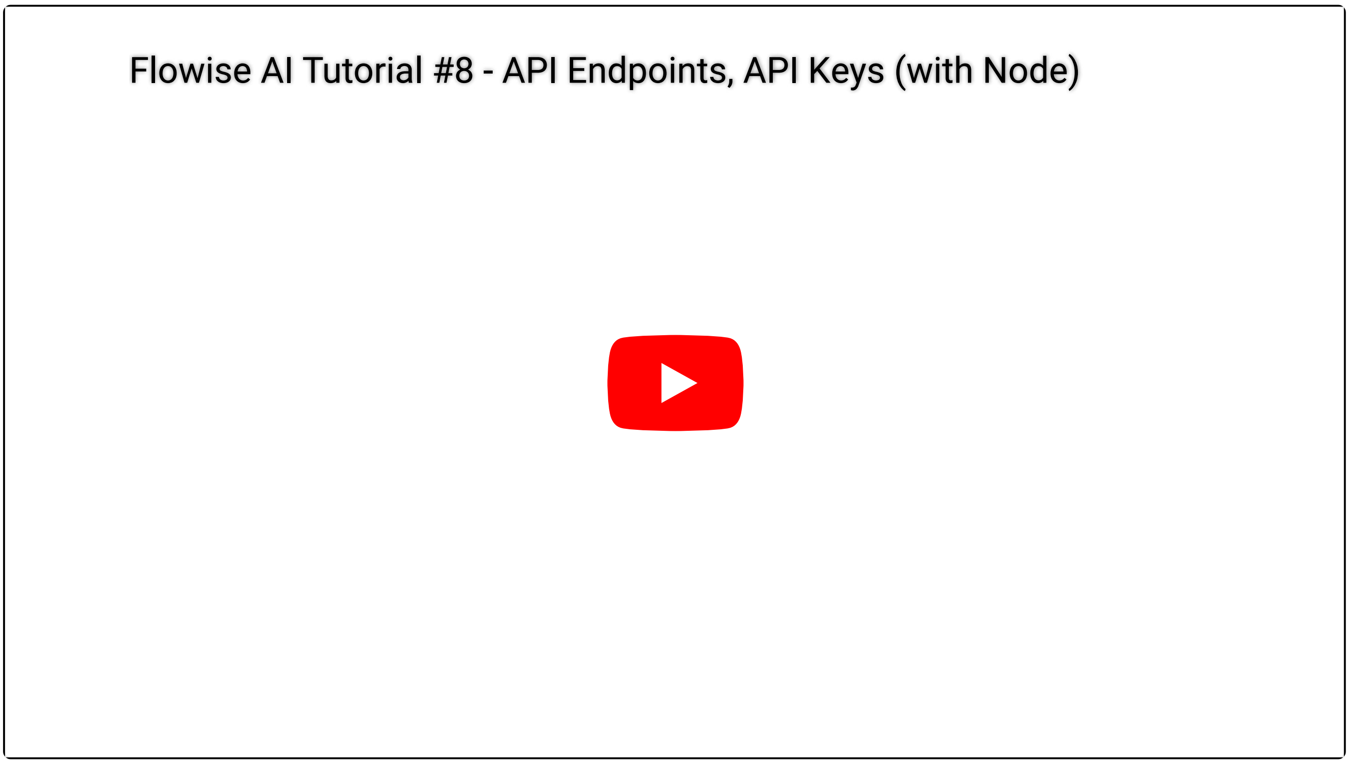
```
"Authorization": "Basic " + Buffer.from(username + ":" + password).toStrir
```

Video Tutorials

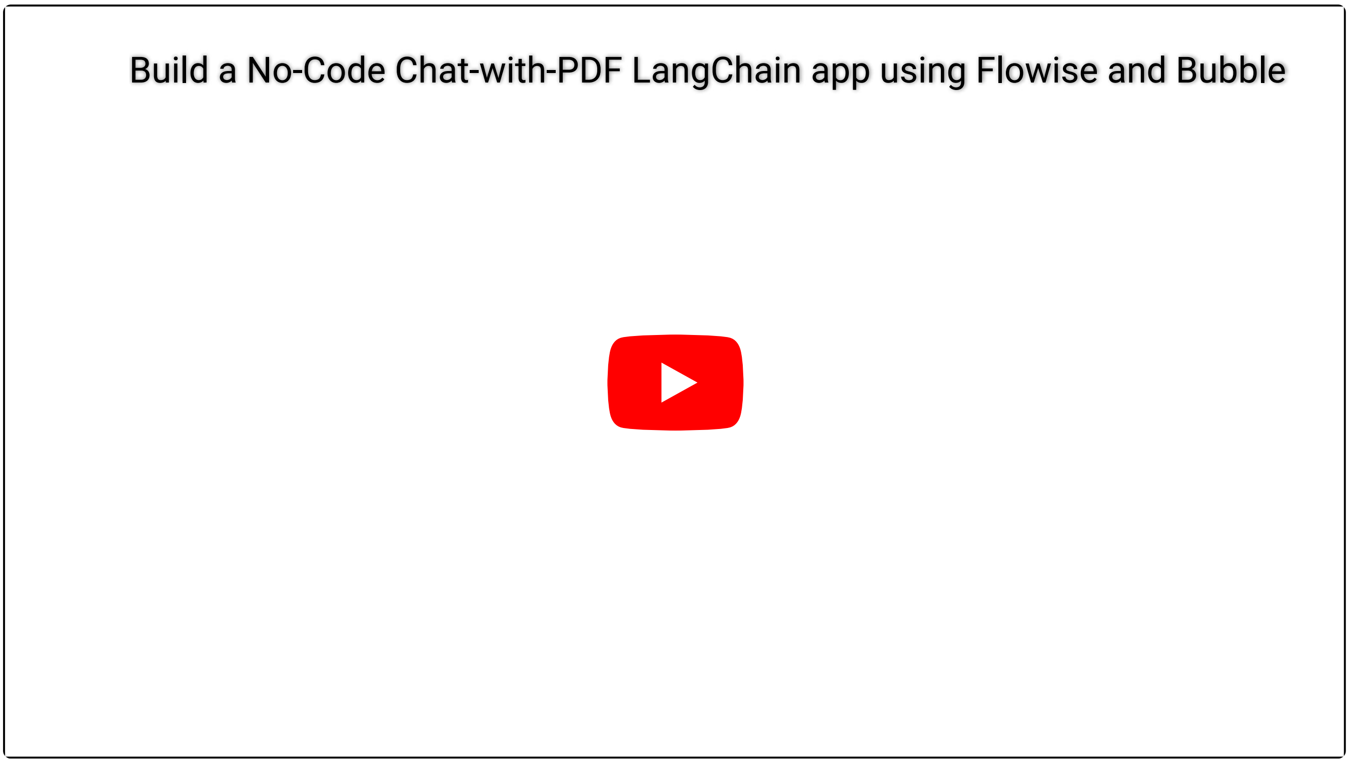
Those video tutorials cover the main use cases for implementing the Flowise API.



Add Flowise to ANYTHING! Flowise API Crash Course



Flowise AI Tutorial #8 - API Endpoints, API Keys (with Node)



Build a No-Code Chat-with-PDF LangChain app using Flowise and Bubble

Previous
Video Tutorials

Next
Analytic

Last updated 13 days ago