☰

API

Streaming messages

If you are interested in implementing the streaming functionality, the following code can be used to output the response of the Chatbase API in a stream format, giving the appearance of natural, hand typed text.

Replace `<Your-Secret-Key>` and `<Your Chatbot ID>` before implementation.

```
// streamer.js

const axios = require('axios')
const {Readable} = require('stream')

const apiKey = '<Your-Secret-Key>'
const chatId = '<Your Chatbot ID>'
const apiUrl = 'https://www.chatbase.co/api/v1/chat'

const messages = [{content: '<Your query here>', role: 'user'}]

const authorizationHeader = `Bearer ${apiKey}`

async function readChatbotReply() {
  try {
    const response = await axios.post(
      apiUrl,
      {
        messages,
        chatId,
        stream: true,
        temperature: 0,
```

```
      },
      headers: {
        Authorization: authorizationHeader,
        'Content-Type': 'application/json',
      },
      responseType: 'stream',
    }
  )

  const readable = new Readable({
    read() {},
  })

  response.data.on('data', (chunk) => {
    readable.push(chunk)
  })

  response.data.on('end', () => {
    readable.push(null)
  })

  const decoder = new TextDecoder()
  let done = false

  readable.on('data', (chunk) => {
    const chunkValue = decoder.decode(chunk)

    // Process the chunkValue as desired
    // Here we just output it as in comes in without \n
    process.stdout.write(chunkValue)
  })

  readable.on('end', () => {
    done = true
  })
} catch (error) {
  console.log('Error:', error.message)
}
}
```

```python
## streamer.py

import requests

api_url = 'https://www.chatbase.co/api/v1/chat'
api_key = '<Your-Secret-Key>'
chat_id = '<Your Chatbot ID>'

messages = [
    { 'content': '<Your query here>', 'role': 'user' }
]

authorization_header = f'Bearer {api_key}'

def read_chatbot_reply():
    try:
        headers = {
            'Authorization': authorization_header,
            'Content-Type': 'application/json'
        }

        data = {
            'messages': messages,
            'chatId': chat_id,
            'stream': True,
            'temperature': 0
        }

        response = requests.post(api_url, json=data, headers=headers, stream=True)
        response.raise_for_status()

        decoder = response.iter_content(chunk_size=None)
        for chunk in decoder:
            chunk_value = chunk.decode('utf-8')
            print(chunk_value, end='', flush=True)
```

```
except requests.exceptions.RequestException as error:
        print('Error:', error)


read_chatbot_reply()
```