Azure OpenAI Chat Completions API Documentation

Endpoint

```
POST https://aai02.eduhk.hk/openai/deployments/gpt-4o-mini/chat/completions
```

⚠ **IMPORTANT: Only streaming mode is enabled. Set `stream: true` in all requests.**

Authentication

- Header: `api-key: {your_api_key}`
- Required: Yes

Request Format

Headers

```
Content-Type: application/json
api-key: {your_api_key}
```

Request Body

```
{
  "messages": [
    {
      "role": "system|user|assistant",
      "content": "message content"
    }
  ],
  "model": "gpt-4o-mini",
  "temperature": 0.7,
  "max_tokens": 1000,
  "stream": true,
  "stream_options": {
    "include_usage": true
  }
}
```

Required Parameters

- `messages`: Array of message objects with `role` and `content`

Optional Parameters

- `model`: String (default: "gpt-4o-mini")
- `temperature`: Number 0-2 (default: 0.7)
- `max_tokens`: Number (default: unlimited)
- `stream`: Boolean (REQUIRED: must be true - streaming only)
- `stream_options`: Object with streaming configuration

Response Formats

Streaming Response (stream: true)

Content-Type: `text/event-stream`

Each chunk format:

```
data: {json_object}
```

Final chunk:

```
data: [DONE]
```

Example streaming chunks:

```
data: {"choices":[{"delta":
{"role":"assistant"},"index":0,"finish_reason":null}],"created":1726041600,"id":"c
hatcmpl-xyz","model":"gpt-4o-mini","object":"chat.completion.chunk"}

data: {"choices":[{"delta":
{"content":"Hello"},"index":0,"finish_reason":null}],"created":1726041600,"id":"ch
atcmpl-xyz","model":"gpt-4o-mini","object":"chat.completion.chunk"}

data: {"choices":[{"delta":{"content":"
there!"},"index":0,"finish_reason":null}],"created":1726041600,"id":"chatcmpl-
xyz","model":"gpt-4o-mini","object":"chat.completion.chunk"}

data: {"choices":[{"delta":
{},"index":0,"finish_reason":"stop"}],"created":1726041600,"id":"chatcmpl-
xyz","model":"gpt-4o-mini","object":"chat.completion.chunk"}

data: [DONE]
```

Non-Streaming Response (stream: false)

❌ **NOT SUPPORTED: Non-streaming mode is disabled. Use streaming mode only.**

Message Roles

- `system`: Instructions for the AI behavior
- `user`: Human input/questions
- `assistant`: AI responses

Error Responses

401 Unauthorized

```
{
  "error": {
    "code": "Unauthorized",
    "message": "Access denied due to missing api-key header"
  }
}
```

400 Bad Request

```
{
  "error": {
    "code": "BadRequest",
    "message": "Invalid request format"
  }
}
```

Implementation Notes for AI Systems

For Streaming Requests:

1. Set `stream: true` in request body
2. Parse each `data:` line as separate JSON
3. Extract `choices[0].delta.content` for text content
4. Stop when receiving `data: [DONE]`
5. Concatenate all content chunks for complete response

Note: Only streaming mode is supported. Do not attempt non-streaming requests.

Example Usage Pattern:

```
import requests
import json

# Streaming (ONLY supported mode)
response = requests.post(
    'https://aai02.eduhk.hk/openai/deployments/gpt-4o-mini/chat/completions',
```

```
    headers={'Content-Type': 'application/json', 'api-key': 'your-key'},
    json={
        'messages': [{'role': 'user', 'content': 'Hello'}],
        'stream': True
    },
    stream=True
)

content = ""
for line in response.iter_lines():
    if line.startswith(b'data: '):
        data = line[6:].decode('utf-8')
        if data == '[DONE]':
            break
        try:
            chunk = json.loads(data)
            delta_content = chunk['choices'][0]['delta'].get('content', '')
            content += delta_content
        except:
            continue
```

Rate Limits and Constraints

- No specific rate limits documented
- Standard HTTP timeout applies
- Maximum context length varies by model

Available Models

- gpt-4o-mini (recommended)
- gpt-4
- gpt-35-turbo

Note: Replace `gpt-4o-mini` in the URL path with desired model name.