API Keys

请求地址

`https://api.lingyiwanwu.com/v1/chat/completions`

入参描述

表1：总体参数

传参方式	字段	类型	必选	描述	默认值	示例值
Header	Content-Type	string	是	内容类型。	N/A	application/js
Header	Authorization	string	是	API Key。	N/A	your-api-key
Body	model	string	是	使用的Yi Model模型ID。	N/A	yi-lightning
Body	messages	array	是	一个由历史消息组成的列表，由系统消息、用户消息和模型消息组成。	N/A	示例值，参阅「表 2 - messages 参数」
Body	tools	array	否	模型可调用的工具列表。目前只支持函数作为工具。	N/A	示例请参阅「表 4 - tools 参数
Body	tool_choice	string 或 object	否	控制模型是否会调用某个或某些工具。`none` 表示模型不会调用任何工具，而是以文字形式进行回复。`auto` 表示模型可选择以文本进行回复或者调用一个或多个工具。在调用时也可以通过将此字段设置为 `required` 或 `{"type":`	N/A	auto

				"function", "function": {"name": "some_function"}} 来更强的引导模型使用工具。		
Body	max_tokens	int or null	否	指定模型在生成内容时token的最大数量，它定义了生成的上限，但不保证每次都会产生到这个数量。	取决于模型	5000
Body	top_p	float	否	控制生成结果的随机性。数值越小，随机性越弱；数值越大，随机性越强。	0.9	取值范围：0至之间。
Body	temperature	float	否	控制生成结果的发散性和集中性。数值越小，越集中；数值越大，越发散。	0.3	取值范围：0至之间。
Body	stream	boolean	否	是否获取流式输出。	false	false

表2：messages 参数

传参方式	字段	类型	必选	描述	对象	默认值	
					系统消息 • content（string｜必选）：系统消息的内容。 • role（string 必选）：消息的发出者，此处需设置为 system。	N/A	1 { 2 "content 3 "role": 4 }

	Body	messages	array <message>	是	一个由历史消息组成的列表，由系统消息、用户消息和模型消息组成。	用户消息 • content（string｜必选）：用户消息的内容。 • role（string 必选）：消息的发出者，此处需设置为 user。	N/A	1 { 2 "content 3 "role": " 4 }
						模型消息 • content（string｜必选）：模型消息的内容。 • role（string 必选）：消息的发出者，此处需设置为 assistant。	N/A	1 { 2 "content 3 "role": " 4 }
						工具消息 • content（string｜必选）：工具消息的内容。 • role（string 必选）：消息的发出者，此处需设置为 tool。 • tool_call_id（string 必	N/A	1 { 2 "tool_ca 3 "role": " 4 "content 5 }

						选）：该条消息响应的工具调用 ID。		

表3：content 参数

传参方式	字段	类型	必选	描述	对象	默认值	子对象
Body	Content	string	是 必须定义 string 或 array	用户消息的内容。	text 输入的内容	N/A	N/A
		array			内容（dict 数据类型）的数组输入的内容	N/A	• Image part（可选项） ◦ type | string ◦ image_url | dict ▪ image_url ▪ url | string URL 或者以 Base64 图 f"data:ima {base64_i • Text part（可选项） ◦ type | string ◦ text | string 说明： • yi-vision-v2 模型支 或者 Base64 图片字符 • 当前单次调用最多可支 输入 • 图片支持以下格式： ◦ JPEG（.jpeg 和 ◦ PNG（.png） ◦ Base64（f"data:image, {base64_image

							• 输入的图片支持 2K 及 张图片大小不应超过 • 模型每理解一张图片约 700 token

表4：tools 参数

传参方式	字段	类型	必选	描述	对象

Body	tools	`List <function>`	否	一个由自定义工具组成的列表。	• type （string｜必选）：工具的类型，目前只支持 `function`。
					• function （object｜必选）：
					  ◦ description （string｜可选）：对工具函数作用的描述，用于帮助模型理解工具的调用时机和方式。
					  ◦ name （string｜必选）：要调用的工具函数的名称。必须是 a-z、A-Z、0-9，或包含下划线和破折号，最大长度为 64。
					  ◦ parameters （object｜可选）：工具函数可接受的参数，需以 JSON 模式对象的形式进行描述。

出参描述

字段	类型	子参数	描述	示例值
id	String	N/A	本次请求的系统唯一码。	cmpl-1cfb
object	String	N/A	对象类型，此处固定是 `chat.completion`。	chat.comp
created	Long	N/A	Unix 当前时间戳。	1178759
model	String	N/A	正在使用的模型名。	yi-lightnin
		index	模型生成结果的序号。0 表示第一个结果。	0
		messages	详细说明，参阅「表 2 -	示例值，参 「表 2 -

choices	List <choice>		messages参数」。	messages数」。

		finish_reason	• 段式 + 流式
			  ◦ stop：表示模型返回了完整的输出。
			  ◦ length：由于生成长度过长导致停止生成内容.
			  ◦ 以 content_filter 开头的表示安全过滤的结果。
			• 仅流式
			  ◦ null：表示正在生成内容。
				stop

usage	List <usage>	completion_tokens	内容生成的 tokens 数量。	48
		prompt_tokens	prompt 使用的 tokens 数量。	18
		total_tokens	总 tokens 用量。	66

请求和响应示例

同步调用

HTTP 示例

- **请求**

```
curl https://api.lingyiwanwu.com/v1/chat/completions \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $API_KEY" \
  -d '{
        "model": "yi-lightning",
        "messages": [{"role": "user", "content": "Hi, who are you?"}],
        "temperature": 0.3
  }'
```

- **响应**


```
{
  "id": "cmpl-c730301f",
  "object": "chat.completion",
  "created": 7825887,
  "model": "yi-lightning",
  "usage": {
    "completion_tokens": 65,
    "prompt_tokens": 15,
    "total_tokens": 80
  },
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "Hello! My name is Yi, and I am a language model based on the transfor
      },
      "finish_reason": "stop"
    }
  ]
}
```


SDK 示例

- **请求**

```
import openai
from openai import OpenAI
API_BASE = "https://api.lingyiwanwu.com/v1"
API_KEY = "your key"
client = OpenAI(
  api_key=API_KEY,
  base_url=API_BASE
)
completion = client.chat.completions.create(
```

```
  model="yi-lightning",
  messages=[{"role": "user", "content": "Hi, who are you?"}]
)
print(completion)
```

- **响应**

```
ChatCompletion(id = 'cmpl-8062fda5',
 choices = [
    Choice(
      finish_reason = 'stop',
      index = 0,
      logprobs = None,
      message = ChatCompletionMessage(
        content = 'Hello! My name is Yi, and I am a language model based on the transform
        role = 'assistant',
        function_call = None,
        tool_calls = None
      )
    )
  ],
  created = 7826404,
  model = 'yi-lightning',
  object = 'chat.completion',
  system_fingerprint = None,
  usage = CompletionUsage(
    completion_tokens = 65,
    prompt_tokens = 15,
    total_tokens = 80
  )
)
```

流式调用

HTTP 示例

- **请求**

```
curl https://api.lingyiwanwu.com/v1/chat/completions \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $API_KEY" \
  -d '{
        "model": "yi-lightning",
        "messages": [{"role": "user", "content": "Hi, who are you?"}],
        "temperature": 0.3,
        "stream": true
  }'
```

- **响应**

```
data: {"id":"cmpl78796a05","object":"chat.completion.chunk","created":7828777,"model":"yi
data: {"id":"cmpl78796a05","object":"chat.completion.chunk","created":7828777,"model":"yi
...
data: {"id":"cmpl78796a05","object":"chat.completion.chunk","created":7828777,"model":"yi
data: [DONE]
```

SDK 示例

- **请求**

```
import openai
from openai import OpenAI
API_BASE = "https://api.lingyiwanwu.com/v1"
API_KEY = "your key"
client = OpenAI(
  api_key=API_KEY,
  base_url=API_BASE
)
completion = client.chat.completions.create(
  model="yi-lightning",
  messages=[{"role": "user", "content": "Hi, who are you?"}],
  stream=True
)
for chunk in completion:
```

```
  print(chunk.choices[0].delta.content or "", end="", flush=True)
```

- **响应**

```
Hello! My name is Yi, and I am a language model based on the transformers arcl
```

List models

功能描述

显示可用的模型。

请求地址

`https://api.lingyiwanwu.com/v1/models`

出参描述

字段	类型	描述	示例值
id	String	可用的模型ID。	yi-lightning
object	String	对象类型，此处固定是`model`。	model
created	Long	Unix当前时间戳。	1178759
ownedBy	String	模型所属的公司，此处固定是01.ai。	01.ai

请求和响应提示

HTTP 示例

- **请求**

```
curl --location 'https://api.lingyiwanwu.com/v1/models' \
  --header "Authorization: Bearer $API_KEY"
```

- **响应**

```
{
  "data": [
    {
      "id": "yi-lightning",
      "object": "model",
      "created": 1708258504,
      "ownedBy": "01.ai",
      "root": "",
      "parent": ""
    }
  ],
  "object": "list"
}
```

SDK 示例

- **请求**

```
import openai
from openai import OpenAI
API_BASE = 'https://api.lingyiwanwu.com/v1'
API_KEY = "your key"
client = OpenAI(
  api_key=API_KEY,
  base_url=API_BASE,
  timeout=300
)
models = client.models.list()
print(models)
```

- **响应**

```
SyncPage[Model](
  data=[
    Model(
      id='yi-lightning',
      created=1708671653,
      object='model',
      owned_by=None,
      ownedBy='01.ai',
      root='',
      parent=''
    )
  ],
  object='list'
)
```

状态码

HTTP 返回码	错误代码	原因	解决方案
400	Bad request	模型的输入+输出（max_tokens）超过了模型的最大上下文。	减少模型的输入，或将 max_tokens 参数值设置更小。
		输入格式错误。	检查输入格式，确保正确。例如，模型名必须全小写，yi-lightning。
401	Authentication Error	API Key缺失或无效。	请确保你的 API Key 有效。
404	Not found	无效的 Endpoint URL 或模型名。	确保使用正确的 Endpoint URL 或模型名。
429	Too Many Requests	在短时间内发出的请求太多。	控制请求速率。
500	Internal Server Error	服务端内部错误。	请稍后重试。
529	System busy	系统繁忙，请重试。	请 1 分钟后重试。