**GLM-4**

GLM-4 系列提供了复杂推理、超长上下文、极快推理速度等多款模型，适用于多种应用场景。

* 模型编码：glm-4-plus、glm-4-air、glm-4-air-0111 Preview 、glm-4-airx、glm-4-long 、glm-4-flashx 、glm-4-flash；
* 大规模处理文本数据，推荐使用 [Batch API](https://www.bigmodel.cn/dev/howuse/batchapi)，一次处理千万级别数据，更有 **5 折**扣优惠；
* 查看 [产品价格](https://www.bigmodel.cn/pricing) ；
* 欢迎在 [体验中心](https://www.bigmodel.cn/console/trialcenter?modelCode=glm-4-plus) 体验；
* 查看模型 [速率限制](https://www.bigmodel.cn/dev/howuse/rate-limits)；
* 查看您的 [API Key](https://www.bigmodel.cn/usercenter/proj-mgmt/apikeys)；

**同步调用**

**接口请求**

| 类型 | 说明 |
| --- | --- |
| 方法 | https |
| 请求URL | https://open.bigmodel.cn/api/paas/v4/chat/completions |
| 调用方式 | 同步调用，等待模型完成执行并返回最终结果或使用SSE调用 |
| 字符编码 | UTF-8 |
| 请求格式 | JSON |
| 响应格式 | JSON或标准Stream Event |
| 请求类型 | POST |
| 开发语言 | 任何能够发起HTTP请求的开发语言 |

**请求参数**

| 参数名称 | 类型 | 必填 | 参数描述 |
| --- | --- | --- | --- |
| model | String | 是 | 要调用的模型编码。 |
| messages | List<Object> | 是 | 调用语言模型时，当前对话消息列表作为模型的提示输入，以JSON数组形式提供，例如{"role": "user", "content": "Hello"}。可能的消息类型包括系统消息、用户消息、助手消息和工具消息。 |
| request\_id | String | 否 | 由用户端传递，需要唯一；用于区分每次请求的唯一标识符。如果用户端未提供，平台将默认生成。 |
| do\_sample | Boolean | 否 | 当do\_sample为true时，启用采样策略；当do\_sample为false时，温度和top\_p等采样策略参数将不生效，模型输出随机性会大幅度降低。默认值为true。 |
| stream | Boolean | 否 | 该参数在使用同步调用时应设置为false或省略。表示模型在生成所有内容后一次性返回所有内容。默认值为false。如果设置为true，模型将通过标准Event Stream逐块返回生成的内容。当Event Stream结束时，将返回一个data: [DONE]消息。 |
| temperature | Float | 否 | 采样温度，控制输出的随机性，必须为正数 取值范围是：[0.0,1.0]， 默认值为 0.95，值越大，会使输出更随机，更具创造性；值越小，输出会更加稳定或确定 建议您根据应用场景调整 top\_p 或 temperature 参数，但不要同时调整两个参数 |
| top\_p | Float | 否 | 用温度取样的另一种方法，称为核取样 取值范围是：[0.0, 1.0]，默认值为 0.70 模型考虑具有 top\_p 概率质量 tokens 的结果 例如：0.10 意味着模型解码器只考虑从前 10% 的概率的候选集中取 tokens 建议您根据应用场景调整 top\_p 或 temperature 参数，但不要同时调整两个参数 |
| max\_tokens | Integer | 否 | 控制生成的响应的最大 token 数量， 默认值：动态计算（默认情况下，max\_tokens的值会根据上下文长度减去输入长度来自动计算） 最大值： max\_tokens 最大支持4095，设置为超过 4095，则会被自动限制为 4095。 |
| response\_format | Object | 否 | 指定模型输出格式，默认为 text, { "type": "text" }：文本输出模式，模型返回普通的文本输出。 { "type": "json\_object" }：JSON输出模式，模型返回有效的 JSON 输出。 Beta 版本采用工程实现方式，实现细节请参考[说明文档](https://www.bigmodel.cn/dev/guidelines/JsonFormat) 。 |
| stop | List | 否 | 模型遇到stop指定的字符时会停止生成。目前仅支持单个stop词，格式为["stop\_word1"]。 |
| tools | List | 否 | 模型可以调用的工具。 |
| type | String | 是 | 工具类型，目前支持 function、retrieval、web\_search。  **function**: `Object` (必需): 仅当工具类型为 function 时补充。  **name**: `String` (必需): 函数名称，只能包含 a-z、A-Z、0-9、下划线和连字符。最大长度限制为64。  **description**: `String` (必需): 用于描述函数的能力。模型将根据此描述确定函数调用的方式。  **parameters**: `Object` (必需): 参数字段必须传递一个Json Schema对象，以准确定义函数接受的参数。如果调用函数时不需要参数，则可以省略此参数。  "parameters": {   "type": "object",   "properties": { "location": {      "type": "string",     "description": "城市，例如：北京"    },    "unit": { "type": "string", "enum": ["celsius", "fahrenheit"] } }, "required": ["location"] }  建议在使用 FunctionCall 时关闭 `do\_sample`，或将 `temperature` 和 `top\_p` 调整为较低值，以提供成功率。  更多详情：[函数调用使用指南](https://www.bigmodel.cn/dev/howuse/functioncall)  **retrieval**: `Object` 描述: 仅当工具类型为 retrieval 时补充。  **knowledge\_id**: `String` (必需): 涉及知识库ID时，请前往开放平台的知识库模块创建或获取。  **prompt\_template**: `String` (非必需): 请求模型时的知识库模板，默认模板：  ```从文档 "{{ knowledge }}" 中查找问题的答案 "{{question}}" 如果找到答案，仅使用文档的陈述来回答问题；如果未找到，则使用自己的知识回答，并告知用户此信息不是来自文档。不要重复问题，直接开始回答。``` 用户自定义模板时，知识库内容占位符和用户端问题占位符必须分别为{{ knowledge }}和{{ question }};  更多详情：[Retrieval使用指南](https://www.bigmodel.cn/dev/howuse/retrieval)  **web\_search**: `Object` 描述: 仅当工具类型为 web\_search 时补充，如果tools中存在type retrieval，则web\_search将不生效。  **enable**: `Boolean` (非必需): 网络搜索功能：默认为关闭状态（False）。启用搜索：设置为 `True`。禁用搜索：设置为 `False`。  **search\_query**: `String` (非必需): 强制自定义搜索键内容。  **search\_result**: `Boolean` (非必需): 获取网页搜索来源的详细信息。默认禁用。启用：true，禁用：false。  更多详情：[web\_search使用指南](https://www.bigmodel.cn/dev/howuse/websearch) |
| tool\_choice | String或Object | 否 | 用于控制模型选择调用哪个函数的方式，仅在工具类型为function时补充。默认auto，目前仅支持auto。 |
| user\_id | String | 否 | 终端用户的唯一ID，帮助平台对终端用户的非法活动、生成非法不当信息或其他滥用行为进行干预。ID长度要求：至少6个字符，最多128个字符。 |

**message 格式**

**System Message** **格式**

| 参数名称 | 类型 | 必填 | 参数说明 |
| --- | --- | --- | --- |
| role | String | 是 | 消息的角色信息，此时应为system |
| content | String | 是 | 消息内容 |

**User Message Format**

| 参数名称 | 类型 | 必填 | 参数说明 |
| --- | --- | --- | --- |
| role | String | 是 | 消息的角色信息，此时应为user |
| content | String | 是 | 消息内容 |

**Assistant Message Format**

| 参数名称 | 类型 | 必填 | 参数说明 |
| --- | --- | --- | --- |
| role | String | 是 | 消息的角色信息，此时应为assistant |
| content | String | 是 | "content"与"tool\_calls"二必选一。 消息内容。其中包括了tool\_calls字段，content字段为空。 |
| tool\_calls | List | 是 | "content"与"tool\_calls"二必选一。 模型产生的工具调用消息 |
| id | String | 是 | 工具id |
| type | String | 是 | 工具类型, 支持web\_search、retrieval、function |
| function | Object | 否 | type为"function"时不为空 |
| name | String | 是 | 函数名称 |
| arguments | String | 是 | 模型生成的调用函数的参数列表，JSON 格式。请注意，模型可能会生成无效的JSON，也可能会虚构一些不在您的函数规范中的参数。在调用函数之前，请在代码中验证这些参数是否有效。 |

**Tool Message格式**

Tool Message表示调用工具后的返回结果。模型然后根据工具消息输出自然语言格式的消息给用户。

| 参数名称 | 类型 | 必填 | 参数描述 |
| --- | --- | --- | --- |
| role | String | 是 | 消息的角色信息，此时应为tool。 |
| content | String | 是 | 工具消息的内容，调用工具后的返回结果。 |
| tool\_call\_id | String | 是 | 工具调用的记录。 |

**响应参数**

| 参数名称 | 类型 | 参数描述 |
| --- | --- | --- |
| id | String | 任务ID |
| created | Long | 请求创建时间，为Unix时间戳，单位为秒 |
| model | String | 模型名称 |
| choices | List | 当前对话的模型输出内容 |
| index | Integer | 结果索引 |
| finish\_reason | String | 模型推理终止的原因。可以是’stop’、‘tool\_calls’、‘length’、‘sensitive’或’network\_error’。 |
| message | Object | 模型返回的文本消息 |
| role | String | 当前对话角色，默认为’assistant’（模型） |
| content | String | 当前对话内容。命中函数时为null，否则返回模型推理结果。 |
| tool\_calls | List<Object> | 模型生成的应调用的函数名称和参数。 |
| function | Object | 包含模型生成的函数名称和JSON格式的参数。 |
| name | String | 模型生成的函数名称。 |
| arguments | String | 模型生成的函数调用参数的JSON格式。调用函数前请验证参数。 |
| id | String | 命中函数的唯一标识符。 |
| type | String | 模型调用的工具类型，目前仅支持’function’。 |
| usage | Object | 模型调用结束时返回的token使用统计。 |
| prompt\_tokens | Integer | 用户输入的token数量 |
| completion\_tokens | Integer | 模型输出的token数量 |
| total\_tokens | Integer | 总token数量 |
| web\_search | List | 返回与网页搜索相关的信息。 |
| icon | String | 来源网站的图标 |
| title | String | 搜索结果的标题 |
| link | String | 搜索结果的网页链接 |
| media | String | 搜索结果网页的媒体来源名称 |
| content | String | 搜索结果网页引用的文本内容 |

**请求示例**

from zhipuai import ZhipuAI

client = ZhipuAI(api\_key="") # 请填写您自己的APIKey

response = client.chat.completions.create(

model="glm-4-plus", # 请填写您要调用的模型名称

messages=[

{"role": "user", "content": "作为一名营销专家，请为我的产品创作一个吸引人的口号"},

{"role": "assistant", "content": "当然，要创作一个吸引人的口号，请告诉我一些关于您产品的信息"},

{"role": "user", "content": "智谱AI开放平台"},

{"role": "assistant", "content": "点燃未来，智谱AI绘制无限，让创新触手可及！"},

{"role": "user", "content": "创作一个更精准且吸引人的口号"}

],

)

print(response.choices[0].message)

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13

**响应示例**

{

"created": 1703487403,

"id": "8239375684858666781",

"model": "glm-4-plus",

"request\_id": "8239375684858666781",

"choices": [

{

"finish\_reason": "stop",

"index": 0,

"message": {

"content": "以AI绘蓝图 — 智谱AI，让创新的每一刻成为可能。",

"role": "assistant"

}

}

],

"usage": {

"completion\_tokens": 217,

"prompt\_tokens": 31,

"total\_tokens": 248

}

}

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21

**流式输出**

**响应参数**

| 参数名称 | 类型 | 参数描述 |
| --- | --- | --- |
| id | String | 智谱AI开放平台生成的任务序号，调用请求结果接口时请使用此序号 |
| created | Long | 请求创建时间，为Unix时间戳，单位为秒 |
| choices | List | 当前对话的模型输出内容 |
| index | Integer | 结果索引 |
| finish\_reason | String | 模型推理终止的原因。'stop’表示自然结束或触发stop词，'tool\_calls’表示模型命中函数，'length’表示达到token长度限制，'sensitive’表示内容被安全审核接口拦截（用户应判断并决定是否撤回公开内容），'network\_error’表示模型推理异常。 |
| delta | Object | 模型增量返回的文本信息 |
| role | String | 当前对话角色，默认为’assistant’（模型） |
| content | String | 当前对话内容。命中函数时为null，否则返回模型推理结果。 |
| tool\_calls | List | 模型生成的应调用的函数名称和参数。 |
| function | Object | 包含模型生成的函数名称和JSON格式的参数。 |
| name | String | 模型生成的函数名称。 |
| arguments | String | 模型生成的函数调用参数的JSON格式。调用函数前请验证参数。 |
| id | String | 命中函数的唯一标识符。 |
| type | String | 模型调用的工具类型，目前仅支持’function’。 |
| usage | Object | 模型调用结束时返回的token使用统计。 |
| prompt\_tokens | Integer | 用户输入的token数量 |
| completion\_tokens | Integer | 模型输出的token数量 |
| total\_tokens | Integer | 总token数量 |
| web\_search | List | 返回与网页搜索相关的信息。icon |
| icon | String | 来源网站的图标 |
| title | String | 搜索结果的标题 |
| link | String | 搜索结果的网页链接 |
| media | String | 搜索结果网页的媒体来源名称 |
| content | String | 搜索结果网页引用的文本内容 |

**请求示例**

最新的模型GLM-4系列模型支持系统提示、函数调用、检索、Web\_Search等新功能。要使用这些新功能，需要升级到最新版本的Python SDK。如果您安装了旧版本的SDK，请更新到最新版本。

pip install --upgrade zhipuai

1

from zhipuai import ZhipuAI

client = ZhipuAI(api\_key="") # 请填写您自己的APIKey

response = client.chat.completions.create(

model="glm-4-plus", # 请填写您要调用的模型名称

messages=[

{"role": "system", "content": "你是一个乐于回答各种问题的小助手，你的任务是提供专业、准确、有洞察力的建议。"},

{"role": "user", "content": "我对太阳系的行星非常感兴趣，尤其是土星。请提供关于土星的基本信息，包括它的大小、组成、环系统以及任何独特的天文现象。"},

],

stream=True,

)

for chunk in response:

print(chunk.choices[0].delta)

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12

响应示例:

data: {"id":"8313807536837492492","created":1706092316,"model":"glm-4-plus","choices":[{"index":0,"delta":{"role":"assistant","content":"土"}}]}

data: {"id":"8313807536837492492","created":1706092316,"model":"glm-4-plus","choices":[{"index":0,"delta":{"role":"assistant","content":"星"}}]}

....

data: {"id":"8313807536837492492","created":1706092316,"model":"glm-4-plus","choices":[{"index":0,"delta":{"role":"assistant","content":"，"}}]}

data: {"id":"8313807536837492492","created":1706092316,"model":"glm-4-plus","choices":[{"index":0,"delta":{"role":"assistant","content":"主要由"}}]}

data: {"id":"8313807536837492492","created":1706092316,"model":"glm-4-plus","choices":[{"index":0,"finish\_reason":"length","delta":{"role":"assistant","content":""}}],"usage":{"prompt\_tokens":60,"completion\_tokens":100,"total\_tokens":160}}

data: [DONE]

1  
2  
3  
4  
5  
6  
7

**函数调用**

**请求示例**

from zhipuai import ZhipuAI

client = ZhipuAI(api\_key="") # 请填写您自己的APIKey

tools = [

{

"type": "function",

"function": {

"name": "query\_train\_info",

"description": "根据用户提供的信息查询火车时刻",

"parameters": {

"type": "object",

"properties": {

"departure": {

"type": "string",

"description": "出发城市或车站",

},

"destination": {

"type": "string",

"description": "目的地城市或车站",

},

"date": {

"type": "string",

"description": "要查询的火车日期",

},

},

"required": ["departure", "destination", "date"],

},

}

}

]

messages = [

{

"role": "user",

"content": "你能帮我查一下2024年1月1日从北京南站到上海的火车票吗？"

}

]

response = client.chat.completions.create(

model="glm-4-plus", # 请填写您要调用的模型名称

messages=messages,

tools=tools,

tool\_choice="auto",

)

print(response.choices[0].message)

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44

**响应示例**

{

"id": "8231168139794583938",

"model": "glm-4-plus",

"request\_id": "8231168139794583938",

"created": 1703490288,

"choices": [

{

"finish\_reason": "tool\_calls",

"index": 0,

"message": {

"role": "assistant",

"tool\_calls": [

{

"id": "call\_8231168139794583938",

"index": 0,

"type": "function",

"function": {

"arguments": '{"date": "2024-01-01","departure": "北京南站","destination": "上海"}',

"name": "query\_train\_info"

}

}

]

}

}

],

"usage": {

"completion\_tokens": 31,

"prompt\_tokens": 120,

"total\_tokens": 151

}

}

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31

**异步调用**

**接口请求**

| 类型 | 说明 |
| --- | --- |
| 传输方式 | HTTPS |
| 请求URL | https://open.bigmodel.cn/api/paas/v4/async/chat/completions |
| 调用方式 | 异步，结果必须通过查询接口获取 |
| 字符编码 | UTF-8 |
| 请求格式 | JSON |
| 响应格式 | JSON |
| HTTP方法 | POST |
| 开发语言 | 任何能够发起HTTP请求的开发语言 |

**请求参数**

请求参数与同步API调用相同。

**响应参数**

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| request\_id | String | 请求发起时客户端提交的任务号或平台生成的任务号。 |
| id | String | 智谱AI开放平台生成的任务序号，查询结果时使用此序号。 |
| model | String | API请求时调用的模型名称。 |
| task\_status | string | 请求的处理状态：PROCESSING（处理中），SUCCESS（成功），FAIL（失败）。此状态必须查询才能确定结果。 |

**调用示例**

from zhipuai import ZhipuAI

client = ZhipuAI(api\_key="") # 请填写您自己的APIKey

response = client.chat.asyncCompletions.create(

model="glm-4-plus", # 请填写您要调用的模型名称

messages=[

{

"role": "user",

"content": "

作为童话之王，请以始终保持一颗善良的心为主题，写一篇简短的童话故事。故事应能激发孩子们的学习兴趣和想象力，同时帮助他们更好地理解和接受故事中蕴含的道德和价值观。"

}

],

)

print(response)

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14

**响应示例**

id='123456789' request\_id='654321' model='glm-4-plus' task\_status='PROCESSING'

1

**任务结果查询**

**接口请求**

| 类型 | 说明 |
| --- | --- |
| 传输方式 | https |
| 请求URL | https://open.bigmodel.cn/api/paas/v4/async-result/{id} |
| 调用方式 | 同步调用，等待模型完全执行并返回最终结果 |
| 字符编码 | UTF-8 |
| 请求格式 | JSON |
| 响应格式 | JSON |
| HTTP方法 | GET |
| 开发语言 | 同步调用，等待模型完全执行并返回最终结果 |

**请求参数**

| 参数名称 | 类型 | 必填 | 描述 |
| --- | --- | --- | --- |
| id | String | 是 | 任务id |

**响应参数**

| 参数名称 | 类型 | 描述 |
| --- | --- | --- |
| model | String | 模型名称 |
| choices | List | 当前对话模型输出内容，目前仅返回一个 |
| index | Integer | 结果索引 |
| finish\_reason | String | 模型推理终止的原因。"stop"表示自然结束或触发stop词，"length"表示达到token长度限制。 |
| message | Object | 模型返回的文本消息 |
| role | String | 当前对话角色，目前默认为assistant（模型） |
| content | String | 当前对话内容 |
| tool\_calls | List | 模型生成的应调用的函数名称和参数。 |
| function | Object |  |
| name | String | 模型生成的应调用的函数名称。 |
| arguments | String | 模型生成的函数调用参数的JSON格式。注意，模型生成的JSON不一定有效，可能包含函数模式中未定义的参数。在调用函数前，请在你的代码中验证参数。 |
| id | String | 命中函数的唯一标识符。 |
| type | String | 模型调用的工具类型，目前仅支持’function’。" |
| task\_status | String | 处理状态：PROCESSING（处理中），SUCCESS（成功），FAIL（失败） |
| request\_id | String | 客户端请求时提交的任务号或平台生成的任务号 |
| id | String | 智谱AI开放平台生成的任务序号，调用请求结果接口时使用此序号 |
| usage | Object | 本次模型调用的token统计 |
| prompt\_tokens | int | 用户输入的token数量 |
| completion\_tokens | int | 模型输出的token数量 |
| total\_tokens | int | 总token数量 |

**请求示例**

import time

from zhipuai import ZhipuAI

client = ZhipuAI(api\_key="") # 请填写您自己的APIKey

response = client.chat.asyncCompletions.create(

model="glm-4-plus", # 请填写您要调用的模型名称

messages=[

{

"role": "user",

"content": "

作为童话之王，请以始终保持一颗善良的心为主题，写一篇简短的童话故事。故事应能激发孩子们的学习兴趣和想象力，同时帮助他们更好地理解和接受故事中蕴含的道德和价值观。"

}

],

)

task\_id = response.id

task\_status = ''

get\_cnt = 0

while task\_status != 'SUCCESS' and task\_status != 'FAILED' and get\_cnt <= 40:

result\_response = client.chat.asyncCompletions.retrieve\_completion\_result(id=task\_id)

print(result\_response)

task\_status = result\_response.task\_status

time.sleep(2)

get\_cnt += 1

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27

**响应示例:**

{"id":"123456789","request\_id":"123123123","model":null,"task\_status":"PROCESSING"}

{"id":"123456789","request\_id":"123123123","model":null,"task\_status":"PROCESSING"}

... ...

{"id":"123456789","request\_id":"123123123","model":null,"task\_status":"PROCESSING"}

{

"id": "123456789",

"request\_id": "123123123",

"model": "glm-4-plus",

"task\_status": "SUCCESS",

"choices": [

{

"index": 0,

"finish\_reason": "stop",

"message": {

"content": "从前，有一个美丽的村庄，那里的孩子们都喜欢一起玩耍、学习和探索。其中有一个小男孩叫小明，他有一颗善良的心，总是乐于帮助别人。

一天，小明在森林里发现了一只翅膀受伤的小鸟，无法飞翔。小明心疼这只小鸟，便把它带回家，用温暖和爱心照顾它。在小明的细心照料下，小鸟的翅膀渐渐康复，开始在房间里飞来飞去。

看到小鸟奇迹般的康复，小明对鸟类产生了浓厚的兴趣，想要了解更多关于鸟类的知识。他开始阅读有关鸟类的书籍，学习它们的习性和生活方式。通过学习，小明对鸟类有了深刻的了解，他和那只小鸟也建立了深厚的友谊。

一天，小明在森林里散步时，发现了一只被困在猎人陷阱里的小兔子。小明毫不犹豫地救出了小兔子。小兔子感激地看着小明，并告诉他森林里有一个神秘的宝藏——一颗能实现愿望的魔法宝石。

充满好奇心的小明决定去寻找这颗宝石。他带着小鸟和小兔子踏上了冒险之旅。在旅途中，他们遇到了许多挑战，但小明始终保持着善良的心，勇敢地面对每一个困难。他不仅学会了如何与森林里的动物相处，还学会了许多生存技能。

经过一段时间的努力，小明终于找到了那颗魔法宝石。宝石发出耀眼的光芒，将小明和他的朋友们带到了一个美丽的世界。在那里，他们遇到了一位智慧的老人，老人告诉小明，宝石的力量来自于一个人的善良之心。只有拥有善良之心的人才能激活宝石的力量，实现自己的愿望。

小明明白了这个道理，感谢了老人，带着宝石回到了现实世界。他用宝石的力量帮助他人，让村庄变得更加美好。小明成为了村庄里的榜样，通过他的行动，孩子们明白了始终保持一颗善良的心的重要性。

从此以后，小明和村民们幸福地生活在一起。听过小明故事的孩子们明白了善良之心的重要性。他们以小明为榜样，努力成为一个有爱心、有责任感的人。在这个过程中，他们的学习兴趣和想象力也被激发，成长为优秀的孩子。

这个故事告诉我们，始终保持一颗善良的心，用我们的行动去影响周围的人。只有拥有善良之心的人，才能解锁自己的潜力，实现梦想。让我们都努力成为一个有善良之心的人。",

"role": "assistant",

"tool\_calls": null

}

}

],

"usage": {

"prompt\_tokens": 52,

"completion\_tokens": 470,

"total\_tokens": 522

}

}

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44