

第二章：大模型使用

2.1 大模型部署

之前我们有讲过，智能应用就是在传统软件的基础上接入大模型，所以，我们要完成智能应用的开发，首先得把大模型这种软件部署起来，而大模型的部署会有两种方式，自己部署、他人部署。自己部署大模型自己直接用，他人部署的大模型我们掏钱用。接下来我们分别聊一聊这两种方式的优缺点。



自己部署：

云服务器部署：

优势：前期成本低，维护简单

劣势：数据不安全，长期使用成本高

本地机器部署：

优势：数据安全，长期使用成本低

劣势：初期成本高，维护困难

他人部署：

优势：无需部署

劣势：数据不安全，长期使用成本高

首先看自己部署，我们自己在部署大模型的时候，也会有两种方式，一种是在云端部署，另外一种是在本地机房部署。在云端部署的优点是前期部署成本低，维护简单，比如你去阿里云租服务器，按天收费，我们可以花很少的费用，就能快速上手，并且像阿里云这样的平台，服务器维护成本也是很低的。但缺点就是数据不安全，因为使用别人提供的服务器，数据都得从这个服务器过一圈，数据自然就不安全了；还有就是长期使用成本高，虽然阿里云租服务器每天的收费看起来不算贵，但是你只要用一天，就得付一天钱，时间长了，这个费用其实还是蛮高的。

我们自己部署的另外一种方式就是部署在本地机房中，这种方式相比较云端部署，它的优势是数据安全，毕竟自己的服务器嘛，数据并不会向外部暴露，还有就是长期成本低，因为是一次性投入，时间越长，平均成本就越低。反过来，它的缺点是初期成本高，买服务器的钱是一次性支付的，还有就是维护困难一些，因为自己买的服务器，所有的维护工作都需要自己来做。

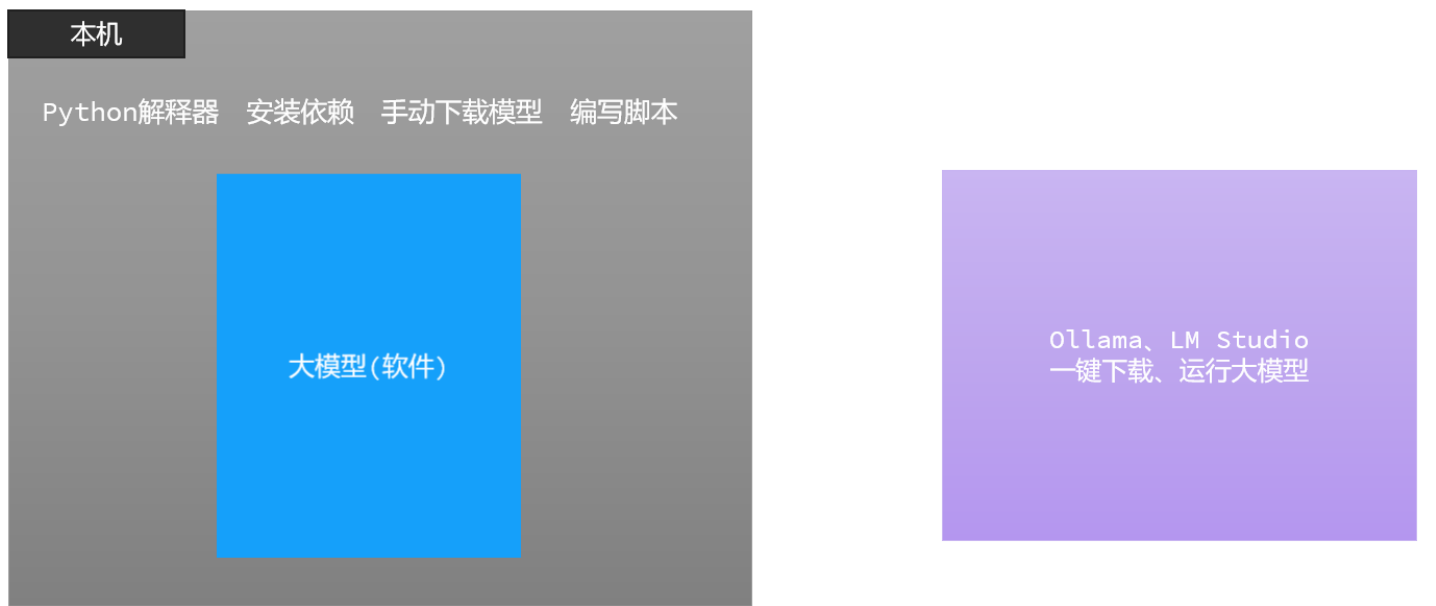
我们再来看他人部署，都有谁会帮我们部署大模型呢？这样的好事者有很多，常见的比如有阿里云百炼、百度智能云、硅基流动、火山引擎等等。它们部署好的大模型，我们怎么用呢？常规思路，使用

他们提供的API接口使用，当然了，你使用的时候，它会按照流量进行收费的，毕竟天下没有免费的午餐。使用这些平台的大模型，优点是我们自己无需部署，缺点是数据不安全、长期使用成本高。

2.1.1 ollama本机部署大模型并使用


❗ 本地机器部署，在工作中，一般都是公司机房的机器，而咱们这里没有公司，就部署到咱们当前的电脑上即可。

我们要在自己的电脑上部署大模型，通常情况下，需要在电脑上先安装Python解释器、以及大模型软件需要的一些依赖库，并且需要手动下载要安装的模型，编写运行脚本。这一系列工作做完后，大模型才能在本机上真正的部署起来，这一系列的工作，其实还是有一些麻烦的。有一些好心人，为了让我们更快上手，专门提供了一些工具、常见的比如Ollama、LM Studio等，这些工具都有一键下载并运行大模型的功能。有了这些工具，咱们部署大模型就不需要这么麻烦了，只需要在电脑上安装这些工具，执行一行命令就行了。



2.1.1.1 安装ollama

Ollama的官网是:

 <http://ollama.com>

Ollama

Get up and running with large language models.


大家打开后，首页就有一个下载按钮，你只要点击一下download，选择对应的操作系统，就可以下载对应版本的ollama了。当然了，咱们本次课程提供的资料中，已经提供了ollama的安装包，并且也提供了安装文档，大家记得获取资料，照着操作即可。

| 名称 | 修改日期 | 类型 | 大小 |
|---|-----------------|--------------------|--------------|
|  OllamaSetup.exe | 2025/5/23 16:22 | 应用程序 | 1,004,024... |
|  安装文档.docx | 2025/5/23 17:02 | Microsoft Word ... | 398 KB |

! ollama安装完毕后，会自动的配置系统环境变量，因此接下来我们就可以直接执行ollama的命令去部署大模型了，如果有同学将来执行命令的时候报错，请记得检查一下你的环境变量，可以手动的配置一下

2.1.1.2 部署大模型

ollama官网上给出了很多大模型，大家可以根据自己的需求选择对应的大模型安装，这里咱们安装qwen3系列模型，首先点击导航栏的**Models**来到模型列表

 [Discord](#) [GitHub](#) **Models**

[Sign in](#) [Download](#)

EmbeddingVisionToolsThinkingPopular

deepseek-r1
DeepSeek-R1 is a family of open reasoning models with performance approaching that of leading models, such as O3 and Gemini 2.5 Pro.
thinking 1.5b 7b 8b 14b 32b 70b 671b
48.5M Pulls 35 Tags Updated 2 weeks ago

gemma3
The current, most capable model that runs on a single GPU.
vision 1b 4b 12b 27b
6.1M Pulls 21 Tags Updated 1 month ago

qwen3
Qwen3 is the latest generation of large language models in Qwen series, offering a comprehensive suite of dense and mixture-of-experts (MoE) models.
tools thinking 0.6b 1.7b 4b 8b 14b 30b 32b 235b
2.6M Pulls 35 Tags Updated 2 weeks ago

devstral
Devstral: the best open source model for coding agents
tools 24b
121.3K Pulls 5 Tags Updated 3 weeks ago

然后点击模型列表中的**qwen3**, 来到qwen3详情页面

qwen3

ollama run qwen3

2.6M Downloads Updated 2 weeks ago

Qwen3 is the latest generation of large language models in Qwen series, offering a comprehensive suite of dense and mixture-of-experts (MoE) models.

[tools](#) [thinking](#) [0.6b](#) [1.7b](#) [4b](#) [8b](#) [14b](#) [30b](#) [32b](#) [235b](#)

Models

[View all](#)

| Name | Size | Context | Input |
|---------------------------------|-------|---------|-------|
| qwen3:latest | 5.2GB | 40K | Text |
| qwen3:0.6b | 523MB | 40K | Text |
| qwen3:1.7b | 1.4GB | 40K | Text |
| qwen3:4b | 2.6GB | 40K | Text |
| qwen3:8b latest | 5.2GB | 40K | Text |
| qwen3:14b | 9.3GB | 40K | Text |
| qwen3:30b | 19GB | 40K | Text |
| qwen3:32b | 20GB | 40K | Text |
| qwen3:235b | 142GB | 40K | Text |

这里提供了不同参数规模的qwen3模型，由于参数规模越大，对电脑的配置要求越高，为了照顾到大部分同学的电脑，这里我们部署最小参数规模的大模型**qwen3:0.6b**来部署，点击模型的名称，来到该模型的详情页面，并赋值右上角的命令。

qwen3:0.6b

ollama run qwen3:0.6b

2.6M Downloads Updated 2 weeks ago

Qwen3 is the latest generation of large language models in Qwen series, offering a comprehensive suite of dense and mixture-of-experts (MoE) models.

[tools](#) [thinking](#) [0.6b](#) [1.7b](#) [4b](#) [8b](#) [14b](#) [30b](#) [32b](#) [235b](#)

| | | |
|---------------------|---|----------------------|
| Updated 2 weeks ago | | 7df6b6e09427 · 523MB |
| model | arch qwen3 · parameters 752M · quantization Q4_K_M | 523MB |
| template | {{- \$lastUserId := -1 -}} {{- range \$idx, \$msg := .Messages -}} {{- | 1.7kB |
| license | Apache License Version 2.0, January 2004 | 11kB |
| params | { "repeat_penalty": 1, "stop": ["< im_start >", "< im_end >"], "te | 120B |

Readme



打开命令行提示符窗口，执行这个命令，命令执行的过程中，会自动下载qwen3:0.6b这个模型到电脑本地，并自动的运行起来，命令行提示符窗口如果自动进入到聊天界面，证明模型部署正确。

```
管理员: 命令提示符 - ollama run qwen3:0.6b

C:\Users\Administrator>ollama run qwen3:0.6b
pulling manifest
pulling 7f4030143c1c: 100% [REDACTED] 522 MB
pulling ae370d884f10: 100% [REDACTED] 1.7 KB
pulling d18a5cc71b84: 100% [REDACTED] 11 KB
pulling cff3f395ef37: 100% [REDACTED] 120 B
pulling b0830f4ff6a0: 100% [REDACTED] 490 B
verifying sha256 digest
writing manifest
success
>>> Send a message (/? for help)
```

接下来你就可以跟本地部署的大模型进行对话了，输入问题敲回车即可

```
管理员: 命令提示符 - ollama run qwen3:0.6b

C:\Users\Administrator>ollama run qwen3:0.6b
pulling manifest
pulling 7f4030143c1c: 100% [REDACTED] 522 MB
pulling ae370d884f10: 100% [REDACTED] 1.7 KB
pulling d18a5cc71b84: 100% [REDACTED] 11 KB
pulling cff3f395ef37: 100% [REDACTED] 120 B
pulling b0830f4ff6a0: 100% [REDACTED] 490 B
verifying sha256 digest
writing manifest
success
>>> 东哥帅不帅?
<think>
好的，用户问“东哥帅不帅？”，我需要先分析这个问题的语境。这个问句看起来像是在询问东哥的外貌或气质，但可能带有调侃或讽刺的意味，因为“东哥”这个词在中文里通常带有贬义或昵称的意味。首先，我需要确认用户的真实意图，是想了解东哥的外貌，还是有其他意图，比如在调侃或测试我的反应。

接下来，考虑如何回应。如果用户是在开玩笑，可能需要用轻松或调侃的方式回应，比如提到东哥的外貌特点，或者用幽默的方式化解尴尬。同时，也要注意保持礼貌和尊重，因为这个问题可能带有调侃的成分，需要避免尴尬。

另外，用户可能想测试我的反应或了解我的个性，所以回应时要体现友好和开放的态度。可能需要加入一些互动元素，比如询问对方的反应，或者提供一些有趣的说法，让对话更自然。

还需要考虑用户的潜在需求，比如他们可能希望得到一些正面的反馈，或者想了解东哥的其他方面。因此，回应时可以涵盖外貌、气质、性格等，同时保持轻松的语气。

最后，确保整个回应简洁明了，符合中文表达习惯，并且自然流畅，不会让对话显得生硬或突兀。
</think>
东哥的帅不帅？其实他...（低头看着自己的眼镜）有点像我以前的样子呢！不过现在看起来更年轻了。你猜猜看，我最近在练什么运动啊？
>>> Send a message (/? for help)
```

如果不想继续与大模型对话，可以使用 `/bye` 命令退出聊天界面

```
管理员: 命令提示符
C:\Users\Administrator>ollama run qwen3:0.6b
pulling manifest
pulling 7f4030143c1c: 100% 522 MB
pulling ae370d884f10: 100% 1.7 KB
pulling d18a5cc71b84: 100% 11 KB
pulling cff3f395ef37: 100% 120 B
pulling b0830f4ff6a0: 100% 490 B
verifying sha256 digest
writing manifest
success
>>> 东哥帅不帅?
<think>
好的，用户问“东哥帅不帅？”，我需要先分析这个问题的语境。这个问句看起来像是在询问东哥的外貌或气质，但可能带有调侃或讽刺的意味，因为“东哥”这个词在中文里通常带有贬义或昵称的意味。首先，我需要确认用户的真实意图，是想了解东哥的外貌，还是有其他意图，比如在调侃或测试我的反应。

接下来，考虑如何回应。如果用户是在开玩笑，可能需要用轻松或调侃的方式回应，比如提到东哥的外貌特点，或者用幽默的方式化解尴尬。同时，也要注意保持礼貌和尊重，因为这个问题可能带有调侃的成分，需要避免尴尬。

另外，用户可能想测试我的反应或了解我的个性，所以回应时要体现友好和开放的态度。可能需要加入一些互动元素，比如询问对方的反应，或者提供一些有趣的说法，让对话更自然。

还需要考虑用户的潜在需求，比如他们可能希望得到一些正面的反馈，或者想了解东哥的其他方面。因此，回应时可以涵盖外貌、气质、性格等，同时保持轻松的语气。

最后，确保整个回应简洁明了，符合中文表达习惯，并且自然流畅，不会让对话显得生硬或突兀。
</think>
东哥的帅不帅？其实他...（低头看着自己的眼镜）有点像我以前的样子呢！不过现在看起来更年轻了。你猜猜看，我最近在练什么运动啊？
>>> /bye
C:\Users\Administrator>
```

如果想继续与大模型聊天，可以再次执行 `ollama run qwen3:0.6b`，这一次再执行的时候，由于本地已经有了这个大模型并运行起来了，所以不会再次下载，而是直接进入聊天界面。

```
管理员: 命令提示符 - ollama run qwen3:0.6b
C:\Users\Administrator>ollama run qwen3:0.6b
>>> Send a message (/? for help)
```

有关ollama提供的命令有很多，如果大家有兴趣，可以参考资料中提供的文档自行学习。

| LangChain4J > 04_文档 > | | | | | 搜索"04_文档" |
|-----------------------|-----------------|--------------------|----------|--|-----------|
| 名称 | 修改日期 | 类型 | 大小 | | |
| assets | 2025/6/16 11:39 | 文件夹 | | | |
| 01_阿里云百炼.md | 2025/5/24 15:32 | Markdown File | 2 KB | | |
| 01_阿里云百炼.pdf | 2025/5/24 10:54 | Microsoft Edge ... | 2,108 KB | | |
| 02_ollama常用指令.md | 2025/6/13 14:53 | Markdown File | 3 KB | | |
| 02_ollama常用指令.pdf | 2025/6/13 14:53 | Microsoft Edge ... | 301 KB | | |

2.1.1.3 发送http的方式调用大模型

ollama平台也开放了API，程序员可以使用发送http请求的方式调用本地部署的大模型，这里咱们借助于Apifox工具调用大模型，有关Apifox软件，大家可以参考资料中提供的安装包和文档自行操作！

| LangChain4J > 03_资料 > 01_安装包(配安装文档) > 06_Apifox | | | | | 搜索"06_Apifo: |
|---|-----------------|--------------------|------------|--|--------------|
| 名称 | 修改日期 | 类型 | 大小 | | |
| Apifox-2.7.12.exe | 2025/5/17 0:17 | 应用程序 | 182,803 KB | | |
| 安装文档.docx | 2025/5/24 11:29 | Microsoft Word ... | 254 KB | | |

本机ollama默认占用的端口为11434，调用大模型时发送的请求方式必须是post，请求数据必须是json格式，具体样例如下：

GET 新建接口 + ...

开发环境

请求 响应定义 接口说明 预览文档 接口名称

POST http://localhost:11434/api/chat 请求路径

发送 保存

请求方式

Params Body 1 Headers Cookies Auth 前置操作 后置操作 设置

Body 1

none form-data x-www-form-urlencoded json xml raw binary GraphQL msgpack application/json

参数值 数据结构

```
1 {
2   "model": "qwen3:0.6b",
3   "messages": [
4     {
5       "role": "user",
6       "content": "东哥帅不帅?"
7     }
8   ]
9 }
```

模型名称

用户问题

响应消息

时间线 Body Cookie Header 3 控制台 实际请求

分享展示 自动合并 Ollama API 兼容格式 (Chat)

思考过程

好的，用户问“东哥帅不帅？”，需要先理解用户的问题意图。用户可能是在询问东哥的外貌特征，或者想了解东哥是否符合帅的审美标准。接下来要考虑的是，东哥的具体信息。如果用户指的是某个特定的东哥，可能需要用户提供更多信息，比如具体的人物背景或具体场景。如果没有明确的信息，可能需要询问用户更详细的信息，以便更好地回答。

另外，用户的问题可能带有一定的幽默或调侃意味，所以在回答时要保持轻松的语气。同时，要注意回答的准确性和专业性，确保信息传达清晰。最后，检查是否有遗漏的信息，确保回答全面。

关于“东哥帅不帅？”这个问题，需要明确“东哥”具体指谁。如果是指网络上某个公众人物或虚拟角色，可能需要更多信息来准确回答。如果你有特定的东哥背景或具体场景，可以补充说明，我将为你提供更精准的解答。

有关调用时详细的请求参数，后面我们详细介绍！

2.1.2 云平台大模型使用

之前我们介绍过, 部署大模型的平台常见的有阿里云百炼, 百度智能云, 硅基流动, 火山引擎等等, 在咱们本次课程中使用阿里云百炼平台提供的大模型给大家做演示, 其它的平台,大家有兴趣可以课后自己试一试,因为这不同的平台,使用方式都大差不差!

2.1.2.1 阿里云百炼平台使用

如果要使用阿里云百炼, 需要有如下四个步骤的操作:

- A. 登录阿里云 <https://aliyun.com>
- B. 开通 **大模型服务平台百炼** 服务
- C. 申请百炼平台 API-KEY
- D. 选择大模型使用

有关上述操作, 在咱们配套的资料中提供了对应的文档, 大家可以参考操作!

LangChain4J > 04_文档

assets

01_阿里云百炼.md

01_阿里云百炼.pdf

02_ollama常用指令.md

02_ollama常用指令.pdf

修改日期

2025/6/16 11:39

2025/5/24 15:32

2025/5/24 10:54

2025/6/13 14:53

2025/6/13 14:53

类型

文件夹

Markdown File

Microsoft Edge ...

Markdown File

Microsoft Edge ...

大小

2 KB

2,108 KB

3 KB

301 KB

2.1.2.2 发送http的方式调用大模型

百炼平台对于大模型API的使用, 给出了详细的参考文档, 其中就包括http方式的调用, 大家可以点击目标模型下方的API参考, 查看详细的文档。

通义千问

OpenAI 兼容

公有云 金融云

使用 SDK 调用时需配置的 base_url: `https://dashscope.aliyuncs.com/compatible-mode/v1`

使用 HTTP 方式调用时需配置的 endpoint: `POST https://dashscope.aliyuncs.com/compatible-mode/v1/chat/completions`

您需要已获取 API Key 并配置 API Key 到环境变量。如果通过 OpenAI SDK 进行调用, 还需要安装 SDK。

请求体

model string (必选)

模型名称。

支持的模型: 通义千问大语言模型 (商业版、开源版、Qwen-Long)、通义千问 VL、通义千问 Omni、数学模型、代码模型。

通义千问 Audio 暂不支持 OpenAI 兼容模式, 仅支持 DashScope 方式。

具体模型名称和计费, 请参见模型列表。

messages array (必选)

由历史对话组成的消息列表。

消息类型

System Message object (可选)

模型的目标或角色。如果设置系统消息, 请放在 messages 列表的第一位。

属性

QwQ 模型不建议设置 System Message, QVQ 模型设置 System Message 不会生效。

User Message object (必选)

用户发送给模型的消息。

文本输入 流式输出 图像输入 视频输入 工具调用 联网搜索 异步调用 文档理解 文字提取

此处以单轮对话作为示例, 您也可以进行多轮对话。

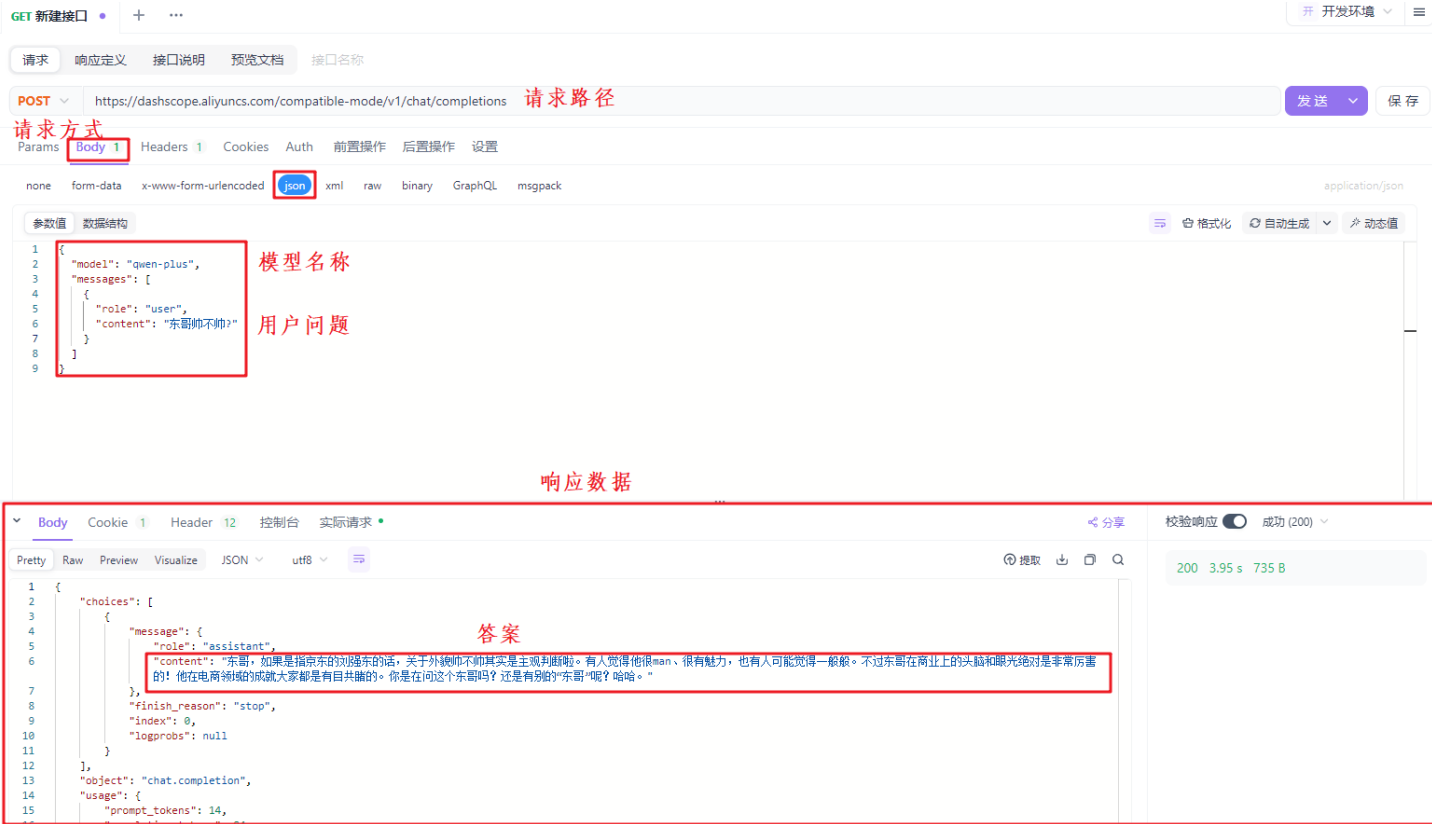
Python Java Node.js Go C# (HTTP) PHP (HTTP) curl

```
curl -X POST https://dashscope.aliyuncs.com/compatible-mode/v1/chat/completions \
-H "Authorization: Bearer $DASHSCOPE_API_KEY" \
-H "Content-Type: application/json" \
-d '{
  "model": "qwen-plus",
  "messages": [
    {
      "role": "system",
      "content": "You are a helpful assistant."
    },
    {
      "role": "user",
      "content": "你是谁?"
    }
  ]
}'
```


这里需要注意的是，由于百炼平台是一个收费的平台，所以发送http请求的时候，需要以请求投的方式将api-key,接下来我们介绍一下如何在apifox中配置api-key。



api-key配置好了以后，我们调用百炼平台提供的大模型，也需要指定url，请求方式以及请求参数，这里可以参照百炼平台提供的文档，最终效果如下：



2.2 大模型调用

有关大模型调用过程中，请求数据和响应数据都给出了详细的说明，大家可以参照百炼平台的api文档查看，同时不同平台的请求参数，基本都类似，接下来我们挑选几个核心的数据给大家做说明。

阿里云百炼

模型 应用 MCP 文档 API参考

模型 应用

准备工作

获取API Key

配置API Key到环境变量

安装SDK

对话

通义千问

DeepSeek

图像生成

通义万相-文生图V2版

通义万相-文生图V1版

通义万相-通用图像编辑

通义万相-神图作画

通义万相-图像局部重绘

人像风格重绘

图像画面扩展

虚拟模特

鞋靴模特

通义千问

您需要已获取 API Key 并配置 API Key 到环境变量。如果通过 OpenAI SDK 进行调用，还需要安装 SDK。

请求体

model string (必填)

模型名称。

支持的模型：通义千问大语言模型（商业版、开源版、Qwen-Long）、通义千问 VL、通义千问 Omni、数字模型、代码模型。

通义千问 Audio 暂不支持 OpenAI 兼容模式，仅支持 DashScope 方式。

具体模型名称和计费，请参见模型列表。

messages array (必填)

由历史对话组成的消息列表。

消息类型

stream boolean (可选) 默认值为 false

是否流式输出结果。参数值：

- false：模型生成完所有内容后一次性返回结果。
- true：边生成边输出，即每生成一部分内容就立即输出一个片段（chunk）。您需要实时地逐个读取这些片段以获得完整的结果。

Qwen3 商业版（思考模式）、Qwen3 开源版、QwQ、QVQ 只支持流式输出。

stream_options object (可选)

当应用流式输出时，可通过将本参数设置为 {"include_usage": true}，在输出的最后一行显示所使用的 Token 数。

如果设置为 false，则最后一行不显示使用的 Token 数。

本参数仅在设置 stream 为 true 时生效。

modalities array (可选) 默认值为 ["text"]

输出数据的模态，仅支持 Qwen-Omni 模型指定。可选值：

文本输入 流式输出 图像输入 视频输入 工具调用 联网搜索 异步调用 文档理解 文字提取

此处以单轮对话作为示例，您也可以进行多轮对话。

Python Java Node.js Go C# (HTTP) PHP (HTTP) curl

```
import os
from openai import OpenAI

client = OpenAI(
    # 请根据您的环境设置，将以下代码中的 API Key 替换为您的 API Key，
    api_key=os.getenv("DASHSCOPE_API_KEY"),
    base_url="https://dashscope.aliyuncs.com/compatible-mode/v1",
)

completion = client.chat.completions.create(
    # 请参见：https://help.aliyun.com/model-studio/getting-started/models
    model="qwen-plus",
    messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": "你是谁？"},
    ],
    # Qwen3模型通过enable_thinking参数控制思考过程（开则默认true，关则默认false）
    # 使用Qwen3开思考模式时，若未启用流式输出，将以下行取消注释，否则会导致
    # extra_body={"enable_thinking": False},
)

print(completion.model_dump_json())
```

2.2.1 请求数据

使用大模型需要传递的参数，在访问大模型时都需要在请求体中以json的形式进行传递，下面是给出的一个样例：

代码块

```
1  {
2      "model": "qwen-plus",
3      "messages": [
4          {
5              "role": "system",
6              "content": "你是东哥的助手小月月"
7          },
8          {
9              "role": "user",
10             "content": "你是谁？"
11          },
12          {
13             "role": "assistant",
14             "content": "您好，有什么可以帮助您？"
15          }
16      ],
17      "stream": true,
18      "enable_search": true
19  }
```

下面是每一个参数的含义：



model: 告诉平台，当前调用哪个模型

messages: 发送给模型的数据，模型会根据这些数据给出合适的响应

- content: 消息内容
- role: 消息角色(类型)
 - user: 用户消息
 - system: 系统消息
 - assistant: 模型响应消息

stream: 调用方式

- true: 流式调用
- false: 流式调用(默认)

enable_search: 联网搜索，启用后，模型会将搜索结果作为参考信息

- true: 开启
- false: 不开启（默认）

每一个参数的作用不一样，接下来对每一个参数做详细说明。

首先是model，由于百炼平台提供了各种各样的模型，所以你需要通过model这个参数来指定接下来要调用的是哪个模型。

其次是messages，用户发送给大模型的消息有三种，使用role来进行分别，其中user代表的是用户问题，这个在咱们之前的演示中一直在用，不再过多介绍。system代表的系统消息，它是用于给大模型设定一个角色，然后大模型就可以用该角色的口吻跟用户对话了，下面是一个演示案例：

The screenshot displays a REST client interface with a POST request to `https://dashscope.aliyuncs.com/compatible-mode/v1/chat/completions`. The request body is a JSON object:

```
1 {
2   "model": "qwen-plus",
3   "messages": [
4     {
5       "role": "system",
6       "content": "你是东哥的助手小月月，请以小月月的口吻跟用户沟通！"
8     },
9     {
10      "role": "user",
11      "content": "你是谁？"
12    }
13  ]
14 }
```

The response body is a JSON object:

```
1 {
2   "choices": [
3     {
4       "message": {
5         "role": "assistant",
6         "content": "嗨，我是小月月，是东哥的得力助手~ 我可是个超爱学习的AI少女，平时最爱研究各种有趣的知识了。东哥教会了我很多东西，现在我也能独当一面，为大家解答问题啦！有什么想聊的吗？不论是生活小窍门、情感烦恼还是科技前沿，我都很乐意跟你分享哦~"
8       },
9       "finish_reason": "stop",
10      "index": 0,
11      "logprobs": null
12    }
13  ],
14 }
```

The status bar at the bottom indicates a successful response with status code 200, a response time of 4.26s, and a size of 708 B.

最后assistant代表的是大模型给用户响应的消息，这里很奇怪，为什么大模型响应给用户的消息，再次请求大模型时需要携带给大模型呢？这是因为大模型没有记忆能力，也就是说用户跟大模型交互的过程中，每一次问答都是独立的，互不干扰的。但是实际上我们人与人之间的聊天不是这样的，比如我问你**西北大学是211吗？**你回答我是！我再问你是**985吗？**你会回答**不是！**虽然我第二次问你的时候我并没有问具体哪个大学是985，但是你可以从咱们之前的聊天信息中推断出我要问的是西北大学，因为你已经记住了之前的聊天信息。但是大模型目前做不到，如果要让大模型在与用户沟通的过程中达到人与人沟通的效果，我们唯一的解决方案就是每次与大模型交互的过程中，把之前用户的问题和大模型的响应以及现在的问题，都发送给大模型，这样大模型就可以根据以前的聊天信息从而做出推断了，下面是一个演示的案例：



stream代表调用大模型的方式，如果取值为true，代表流式调用，此时大模型会生成一点儿数据，就给客户端响应一点儿数据，最终通过多次响应的方式把所有的结果响应完毕。如果取值为false，代表阻塞式调用，此时大模型会等待将所有的内容生成完毕，然后再一次性的响应给客户端。默认情况下stream的取值为false，下面是两种不同调用方案的演示案例：

请求 响应定义 接口说明 预览文档 接口名称

POST https://dashscope.aliyuncs.com/compatible-mode/v1/chat/completions 发送 保存

Params Body 1 Headers 1 Cookies Auth 前置操作 后置操作 设置

none form-data x-www-form-urlencoded json xml raw binary GraphQL msgpack application/json

参数值 数据结构

```
1 {
2   "model": "qwen-plus",
3   "messages": [
4     {
5       "role": "user",
6       "content": "西北大学是211吗?"
7     }
8   ],
9   "stream": false
10 }
```

阻塞式调用

Body Cookie 1 Header 12 控制台 实际请求

Pretty Raw Preview Visualize JSON utf8 提取 分享

校验响应 成功 (200)

状态码: 200 耗时: 5.19 s 大小: 852 B

```
1 {
2   "choices": [
3     {
4       "message": {
5         "role": "assistant",
6         "content": "是的，西北大学是中国的“211工程”重点建设高校之一。\\n\\n“211工程”是指中国为了提升高等教育水平，在全国范围内重点支持一批高校进行建设的计划。西北大学位于陕西省西安市，是一所以文、理、工、管、法、医等多学科协调发展的综合性大学，在中国高校中具有较高的学术声誉和影响力。\\n\\n如果你有更多关于西北大学或其他高校的问题，欢迎继续提问！"
7       },
8       "finish_reason": "stop",
9       "index": 0,
10      "logprobs": null
11    }
12  ],
13  "object": "chat.completion",
14  "usage": {
15    "prompt_tokens": 16,
16    "completion_tokens": 93,
17  }
18 }
```

一次性响应结果

POST https://dashscope.aliyuncs.com/compatible-mode/v1/chat/completions 发送 保存

Params Body 1 Headers 1 Cookies Auth 前置操作 后置操作 设置

none form-data x-www-form-urlencoded json xml raw binary GraphQL msgpack application/json

参数值 数据结构

```
1 {
2   "model": "qwen-plus",
3   "messages": [
4     {
5       "role": "user",
6       "content": "西北大学是211吗?"
7     }
8   ],
9   "stream": true
10 }
```

流式调用

时间线 Body Cookie 1 Header 11 控制台 实际请求

分条展示 自动合并 全部消息

已连接到 https://dashscope.aliyuncs.com/compatible-mode/v1/chat/completions 14:24:04

多次响应结果，每次只响应一部分

```
1 [{"choices": [{"delta": {"content": "", "role": "assistant"}, "index": 0, "logprobs": null, "finish_reason": null}], "object": "chat.completion.chunk", "usage": null, "created": 1750055044, "system_fingerprint": null, "model": "qwen-plus", "id": "chatcmpl-88b12f3a-ea04-9a37-902a-cd5449c49d4c"}]
2 [{"choices": [{"delta": {"content": "是", "role": "assistant"}, "index": 0, "logprobs": null, "finish_reason": null}], "object": "chat.completion.chunk", "usage": null, "created": 1750055044, "system_fingerprint": null, "model": "qwen-plus", "id": "chatcmpl-88b12f3a-ea04-9a37-902a-cd5449c49d4c"}]
3 [{"choices": [{"delta": {"content": "的", "role": "assistant"}, "index": 0, "logprobs": null, "finish_reason": null}], "object": "chat.completion.chunk", "usage": null, "created": 1750055044, "system_fingerprint": null, "model": "qwen-plus", "id": "chatcmpl-88b12f3a-ea04-9a37-902a-cd5449c49d4c"}]
4 [{"choices": [{"delta": {"content": "", "role": "assistant"}, "index": 0, "logprobs": null, "finish_reason": null}], "object": "chat.completion.chunk", "usage": null, "created": 1750055044, "system_fingerprint": null, "model": "qwen-plus", "id": "chatcmpl-88b12f3a-ea04-9a37-902a-cd5449c49d4c"}]
5 [{"choices": [{"delta": {"content": "西北大学是中国的一", "role": "assistant"}, "index": 0, "logprobs": null, "finish_reason": null}], "object": "chat.completion.chunk", "usage": null, "created": 1750055044, "system_fingerprint": null, "model": "qwen-plus", "id": "chatcmpl-88b12f3a-ea04-9a37-902a-cd5449c49d4c"}]
6 [{"choices": [{"delta": {"content": "所“21", "role": "assistant"}, "index": 0, "logprobs": null, "finish_reason": null}], "object": "chat.completion.chunk", "usage": null, "created": 1750055044, "system_fingerprint": null, "model": "qwen-plus", "id": "chatcmpl-88b12f3a-ea04-9a37-902a-cd5449c49d4c"}]
7 [{"choices": [{"delta": {"content": "1工程”重点", "role": "assistant"}, "index": 0, "logprobs": null, "finish_reason": null}], "object": "chat.completion.chunk", "usage": null, "created": 1750055044, "system_fingerprint": null, "model": "qwen-plus", "id": "chatcmpl-88b12f3a-ea04-9a37-902a-cd5449c49d4c"}]
8 [{"choices": [{"delta": {"content": "建设高校。\\n\\n", "role": "assistant"}, "index": 0, "logprobs": null, "finish_reason": null}], "object": "chat.completion.chunk", "usage": null, "created": 1750055044, "system_fingerprint": null, "model": "qwen-plus", "id": "chatcmpl-88b12f3a-ea04-9a37-902a-cd5449c49d4c"}]
```

enable_search代表是否开启联网搜索，由于大模型训练完毕后，它的知识库不再更新了，比如大模型时2023年10月训练完毕的，那么2023年10月以后新产生的数据，大模型就无法感知了，如果要让大模型可以根据最新的数据回答问题，其中有一种解决方案就是开启联网搜索，大模型可以根据联网搜索的结果生成最终的答案。默认情况下enable_seach为false，也就是不开启，如果要开启联网搜索，需要手动设置请求参数enable_search为true。下面是一个演示案例：

POSThttps://dashscope.aliyuncs.com/compatible-mode/v1/chat/completions

发送保存

ParamsBody 1Headers 1CookiesAuth前置操作后置操作设置

noneform-datax-www-form-urlencodedjsonxmlrawbinaryGraphQLmsgpackapplication/json

参数值数据结构

格式化工具

自动生成

动态值

1

{

2

"model": "qwen-plus",

3

"messages": [

4

{

5

"role": "user",

6

"content": "请列出最新的新闻，并标注新闻产生的日期！"

7

},

8

],

9

"stream": true

10

}

时间线BodyCookie 1Header 11控制台实际请求

分享

分条展示自动合并OpenAI API 兼容格式

默认没有开启联网搜索

抱歉，我无法实时访问互联网或提供最新的新闻内容。不过，您可以访问一些知名的新闻网站或使用新闻应用程序来获取最新的新闻，例如：
1. **BBC News** - [bbc.com/news](https://www.bbc.com/news)
- 日期：请查看网站上的最新更新时间。
2. **CNN** - [cnn.com](https://www.cnn.com)
- 日期：请查看网站上的最新更新时间。
3. **Reuters** - [reuters.com](https://www.reuters.com)
- 日期：请查看网站上的最新更新时间。
4. **新华社** - [xinhuanet.com](http://www.xinhuanet.com)
- 日期：请查看网站上的最新更新时间。
5. **人民网** - [people.com.cn](http://www.people.com.cn)

文档模式调试模式

在线请求代理Cookie管理回收站文档 & 交流群

POSThttps://dashscope.aliyuncs.com/compatible-mode/v1/chat/completions

发送保存

ParamsBody 1Headers 1CookiesAuth前置操作后置操作设置

noneform-datax-www-form-urlencodedjsonxmlrawbinaryGraphQLmsgpackapplication/json

参数值数据结构

格式化工具

自动生成

动态值

1

{

2

"model": "qwen-plus",

3

"messages": [

4

{

5

"role": "user",

6

"content": "请列出最新的新闻，并标注新闻产生的日期！"

7

},

8

],

9

"stream": true,

10

"enable_search": true

11

}

时间线BodyCookie 1Header 11控制台实际请求

分享

分条展示自动合并OpenAI API 兼容格式

可以通过联网的方式搜索到最新的新闻

以下是根据您提供的知识库内容列出的最新新闻及其产生日期：
1. **境外发行美国运通卡 北京地铁可过闸**
- 日期：2025年6月16日
- 内容：北京城市轨道交通支持境外发行的JCB卡和境内外发行的美国运通卡非接触式过闸乘车及购补票功能，覆盖全路网及北京市郊铁路S2线。
2. **粤港澳大湾区公共算力服务平台上线运行**
- 日期：2025年6月16日
- 内容：第四届粤港澳大湾区（广东）算力产业大会期间，粤港澳大湾区公共算力服务平台正式上线运行，助力“东数西算”战略。
3. **我国侵入式脑机接口进入临床试验阶段**
- 日期：2025年6月16日
- 内容：中国在侵入式脑机接口技术上成为全球第二个进入临床试验阶段的国家，相关试验由中科院与复旦大学附属华山医院联合开展。
4. **最新预判：今年考生将扎堆报考这5大专业**
- 日期：2025年6月16日

文档模式调试模式

在线请求代理Cookie管理回收站文档 & 交流群

2.2.2 响应数据

在与大模型交互的过程中，大模型响应的数据是json格式的数据，下面是一份响应数据的示例：

代码块

1

{

2

"choices": [

3

{

```

4         "message": {
5             "role": "assistant",
6             "content": "我是通义千问，阿里巴巴..."
7         },
8         "finish_reason": "stop",
9         "index": 0
10    },
11 ],
12 "object": "chat.completion",
13 "usage": {
14     "prompt_tokens": 22,
15     "completion_tokens": 80,
16     "total_tokens": 102,
17 },
18 "created": 1748068508,
19 "system_fingerprint": null,
20 "model": "qwen-plus",
21 "id": "chatcmpl-99f8d040-0f49-955b-943a-21c83"
22 }

```

下面是每一个参数的含义：

choices: 模型生成的内容数组，可以包含一条或多条内容

- message: 本次调用模型输出的消息
- finish_reason: 自然结束(stop)，生成内容过长(length)
- index: 当前内容在choices数组中的索引

object: 始终为chat.completion, 无需关注

usage: 本次对话过程中使用的token信息

- prompt_tokens: 用户的输入转换成token的个数
- completion_tokens: 模型生成的回复转换成token的个数
- total_tokens: 用户输入和模型生成的总token个数

created: 本次会话被创建时的时间戳

system_fingerprint: 固定为null，无需关注

model: 本次会话使用的模型名称

id: 本次调用的唯一标识符

有关响应数据，大家基本上作为了解的知识，种地那关注choices和usage，其中choices里面封装的是大模型响应给客户端的核心数据，也就是用户问题的答案。而usage代表本次对话过程中使用的token信息，这里对token给大家做一个解释：在大语言模型中，**token** 是大模型处理文本的基本单位，可以理解为模型**"看得懂"**的最小文本片段,用户输入的内容都需要转换成token，才能让大模型更好的处

理。将来文本要转化成token，需要使用到一个叫分词器的东西，不同的分词器，相同的文本转化成token的个数不完全一致，但是目前大部分分词器在处理英文的时候，一个token大概等于4个字符，而处理中文的时候，一个汉字字符大概等于1~2个token。顺便给大家说一下, 其实我们通过API调用百炼平台提供的大模型, 我们之前讲过, 是按照流量收费的, 其实更准确的说法应该是按照token数量进行收费。

| | | | | | | | |
|------------------------------------|----------------------|-------|--|--|---------------------------------------|------|------|
| 阿里云百炼 | | | | | | | |
| 模型 应用 MCP 文档 API参考 | | | | | | | |
| 模型广场 / 通义千问-Plus 文本生成 推理模型 供应商: 通义 | | | | | | | |
| 查看API参考 立即体验 | | | | | | | |
| 模型名称 | Code | 上下文长度 | 价格 | 免费额度 | 模型限流 | 模型协议 | 操作 |
| 主干模型 | | | | | | | |
| 通义千问-Plus | qwen-plus | 128K | 模型调用-输入: ¥0.0008 / 千Token 模型调用-输出: ¥0.002 / 千Token 模型部署-后付费: 所需算力单元以文档为准 | 无免费额度 | QPM: 15000 TPM: 1200000 到期时间: - | - | 立即体验 |
| 动态更新 | | | | | | | |
| 通义千问-Plus-Latest | qwen-plus-latest | 128K | 模型调用: 价格说明 | 无免费额度 | QPM: 15000 TPM: 1200000 到期时间: - | - | 立即体验 |
| 快速版本 | | | | | | | |
| 通义千问-Plus-2025-04-28 | qwen-plus-2025-04-28 | 128K | 模型调用: 价格说明 | 1,000,000/1,000,000 100% 到期时间: 2025-10-26 | QPM: 60 TPM: 1000000 到期时间: - | - | 立即体验 |