华清远见 HQYJ.COM

对话机器人fastapi

fastapi介绍

FastAPI 是一个用于构建API的现代化、快速（高性能）的Web框架，使用Python 3.7+的标准类型提示。它的性能媲美Node.js和Go，是基于Python的框架中最快的之一。

主要特点：

- 高性能：与Starlette、Pydantic等框架深度集成，性能优异。
- 简洁明了：基于类型提示（Type Hints），使得代码更加简洁且具备良好的可读性。
- 自动生成文档：自动生成Swagger UI和ReDoc文档，方便开发者查看和测试API。
- 异步支持：原生支持Python的async和await，适合处理异步任务。

fastapi安装

```
pip install fastapi uvicorn
```

代码示例

以下是一个基于 FastAPI 框架的简单对话机器人示例，结合了OpenAI的API，用于处理用户的输入并生成对话。
首先我们需要一个可以进行模型对话的api，这里使用之前学过的vllm封装方式。
启动vllm大模型

```
python -m vllm.entrypoints.openai.api_server --model
/root/hqyj_ai/models/models/Qwen/Qwen2___5-0___5B-Instruct --
served-model-name Qwen2___5-0___5B-Instruct &
```

导入依赖：

```
from typing import List
from fastapi import FastAPI, Body
from openai import OpenAI
```

初始化 FastAPI 应用:
app = FastAPI() 用于创建一个FastAPI应用实例。

```
# 初始化 FastAPI 应用
app = FastAPI()
```

初始化 OpenAI 客户端:
aclient = AsyncOpenAI(api_key=api_key, base_url=base_url) 初始化了异步OpenAI客户端，其中 api_key 和 base_url 是用于访问API的凭证和地址，我们使用vllm的端口地址。

```
# 初始化 OpenAI 客户端
api_key = "EMPTY"
base_url = "http://localhost:8000/v1"
aclient = AsyncOpenAI(api_key=api_key, base_url=base_url)
```

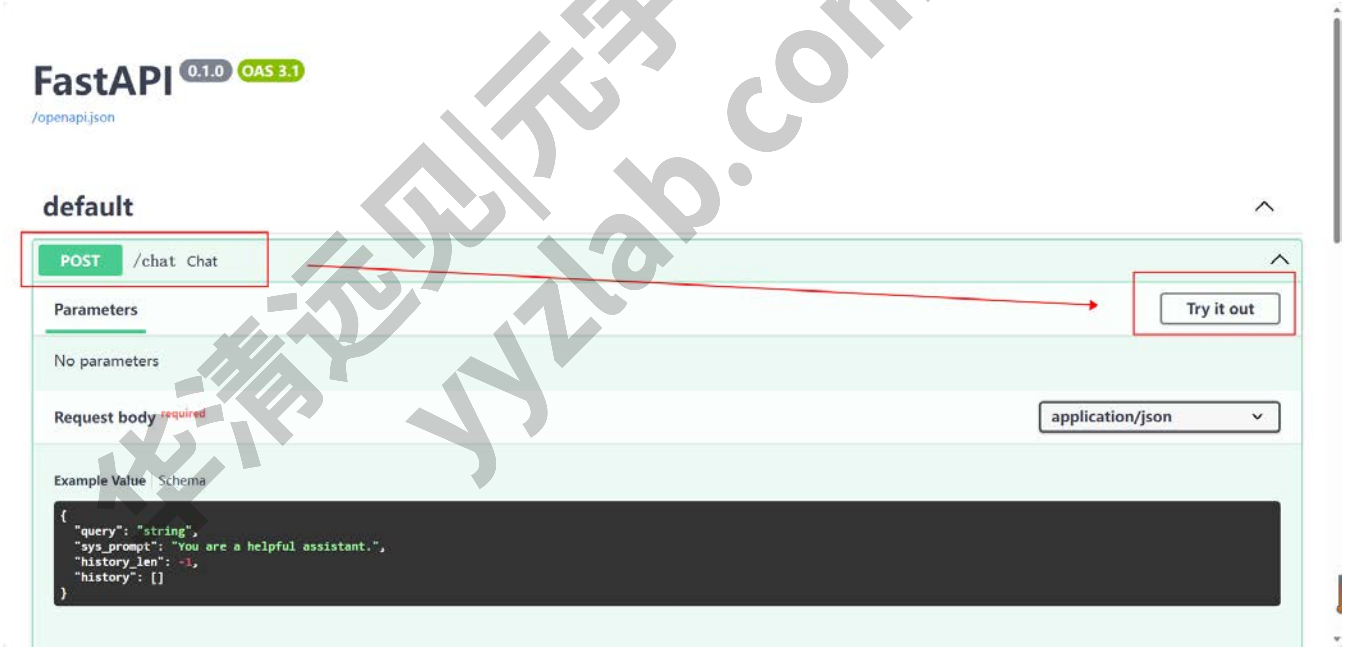
/chat API 接口:
@app.post("/chat") 定义了一个POST请求的API接口，用于接收用户的对话请求。

参数:
query: 用户的输入问题。
sys_prompt: 系统提示（角色设定）。
history_len: 历史记录长度，默认值为-1，表示不限制历史记录。
history: 历史对话记录，用于提供上下文记忆。
历史消息的管理:
使用 messages 列表来管理对话，消息包括用户的输入和系统提示。
在接口的实现中，允许根据 history_len 参数控制保留历史消息的数量，以便优化对话的上下文记忆。

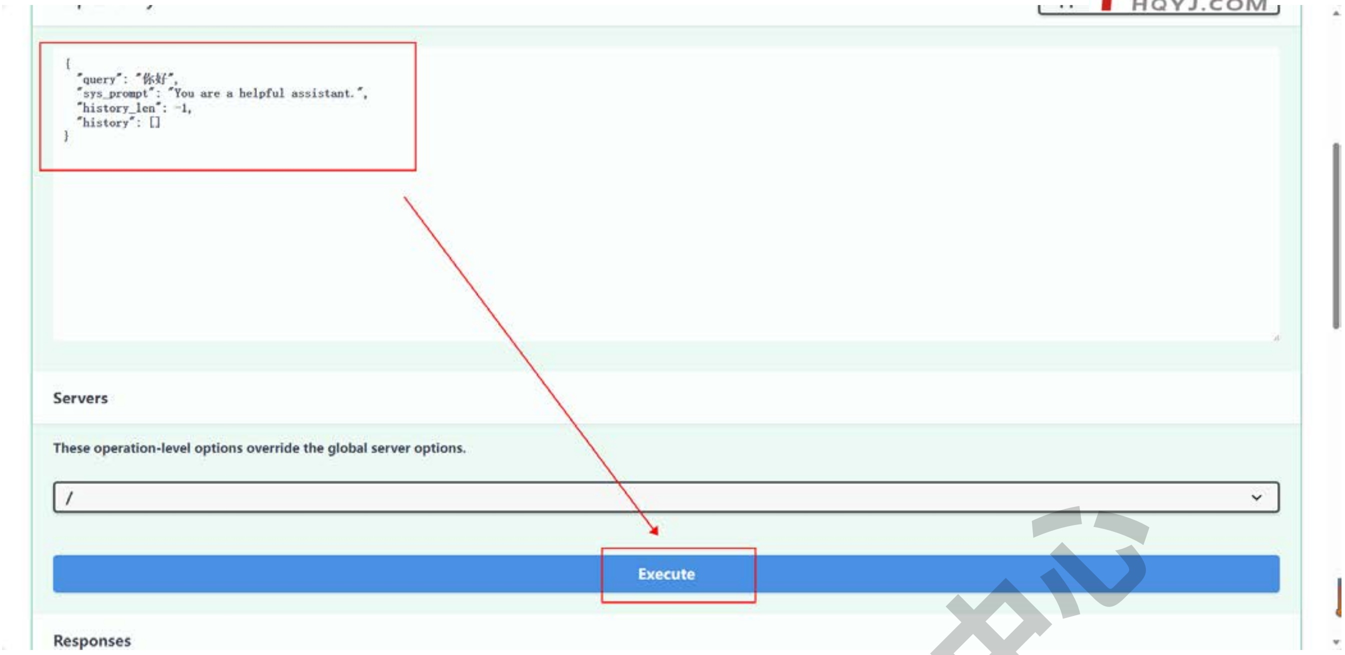
```
# 初始化历史对话列表
messages = []


@app.post("/chat")
async def chat(
        query: str = Body(..., description="用户输入"),
        sys_prompt: str = Body("You are a helpful assistant.",
description="系统提示"),
        history_len: int = Body(-1, description="保留历史消息的数
量"),
        history: List = Body([], description="历史对话"),
        temperature: float = Body(0.7, description="LLM采样温度"),
        top_p: float = Body(0.7, description="LLM采样概率"),
        max_tokens: int = Body(None, description="LLM最大token数配
置"),
):
    global messages

    # 控制历史记录长度，保留指定数量的消息
    if history_len > 0:
        history = history[-2 * history_len:]

    # 在消息列表中加入系统提示
    messages.clear()
    messages.append({"role": "system", "content": sys_prompt})

    # 在消息列表中加入历史消息
    messages.extend(history)

    # 在消息列表中加入用户输入
    messages.append({"role": "user", "content": query})
    print(messages)

    # 生成异步生成器，用于流式输出助手响应
    async def generate_response():
        # 发送请求
        response = await aclient.chat.completions.create(
            model='Qwen2___5-0___5B-Instruct',  # 使用的模型
            messages=messages,
            max_tokens=max_tokens,  # 返回文本的最大长度
            temperature=temperature,  # 控制生成文本的随机性
            top_p=top_p,
            stream=True,  # 启用流式响应
        )
        if stream:
            # 逐步获取助手的响应内容并流式返回
            async for chunk in response:
                chunk_message = chunk.choices[0].delta.content
                if chunk_message:
                    # 函数中看到 yield 关键字时，这意味着该函数是一个生成
器函数。
                    # 如果存在有效消息chunk_message，则使用 yield 将这
条消息返回给调用者。
```

```
                        yield chunk_message

    # 返回 StreamingResponse，以流的形式发送数据
    return StreamingResponse(generate_response(),
media_type="text/plain")


# 程序主入口
if __name__ == "__main__":
    # 导入unicorn服务器的包
    import uvicorn

    # 运行服务器
    uvicorn.run(app, host="127.0.0.1", port=6605,
log_level="info")
```

运行代码启动服务器，通过`http://127.0.0.1:6605/docs`可以进入自动生成的接口文档，
在文档中可以测试api接口。



```
{
  "query": "string",
  "sys_prompt": "You are a helpful assistant.",
  "history_len": -1,
  "history": []
}
```



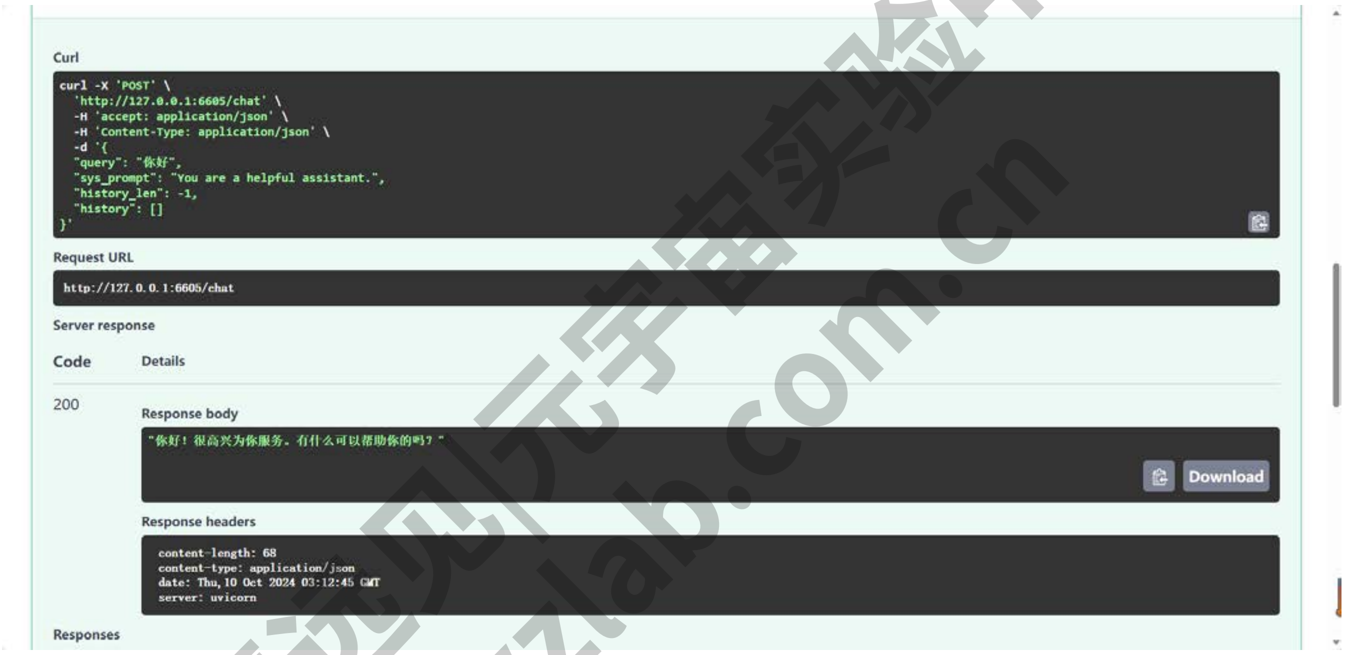
```
{
  "query": "你好",
  "sys_prompt": "You are a helpful assistant.",
  "history_len": -1,
  "history": []
}
```

Servers

These operation-level options override the global server options.

/

Responses



Curl

```
curl -X 'POST' \
  'http://127.0.0.1:6605/chat' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "query": "你好",
  "sys_prompt": "You are a helpful assistant.",
  "history_len": -1,
  "history": []
}'
```

Request URL

```
http://127.0.0.1:6605/chat
```

Server response

Code Details

200

Response body

```
"你好！很高兴为你服务。有什么可以帮助你的吗？"
```

Response headers

```
content-length: 68
content-type: application/json
date: Thu, 10 Oct 2024 03:12:45 GMT
server: uvicorn
```

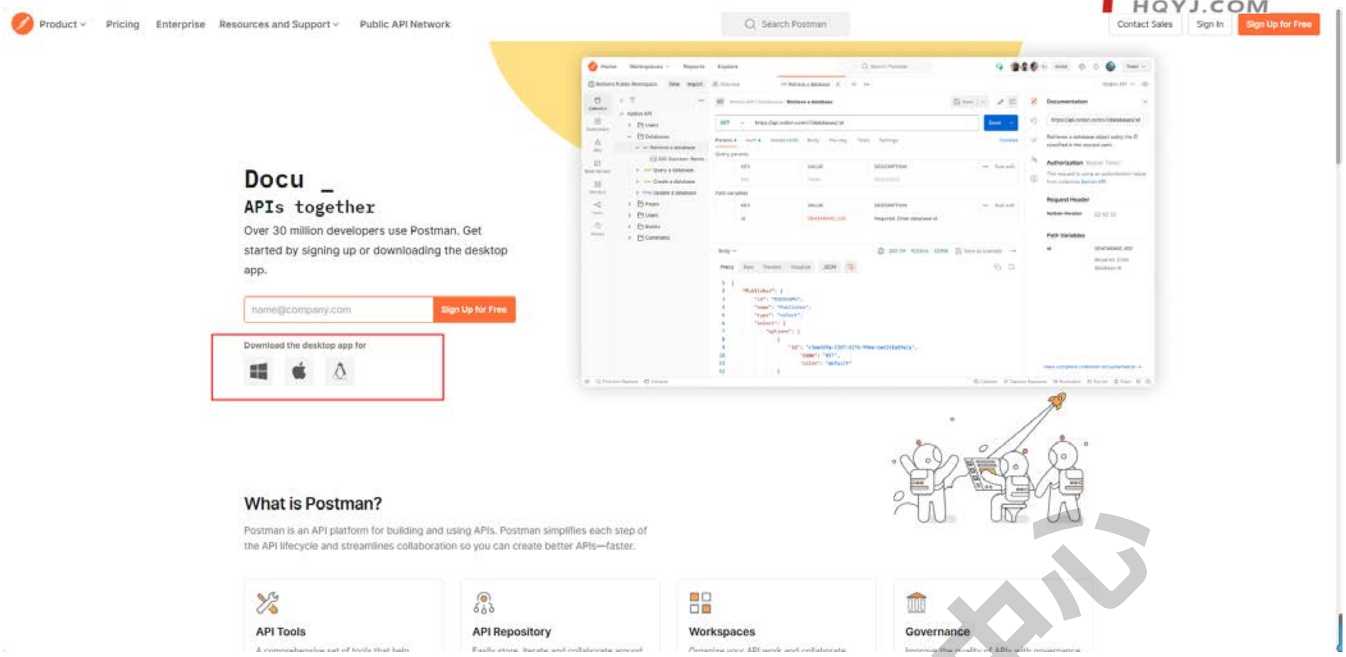
Responses

使用postman测试接口

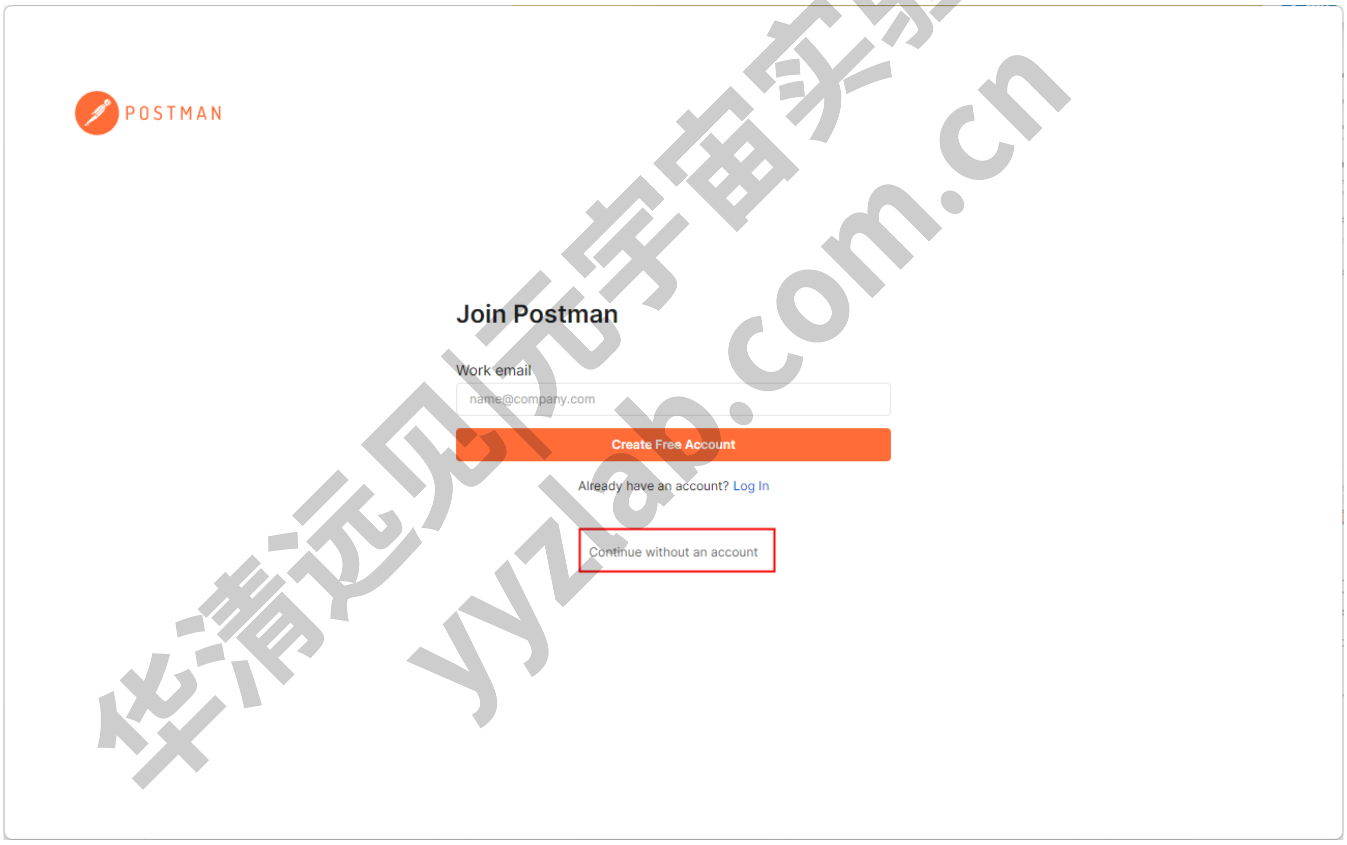
下载postman

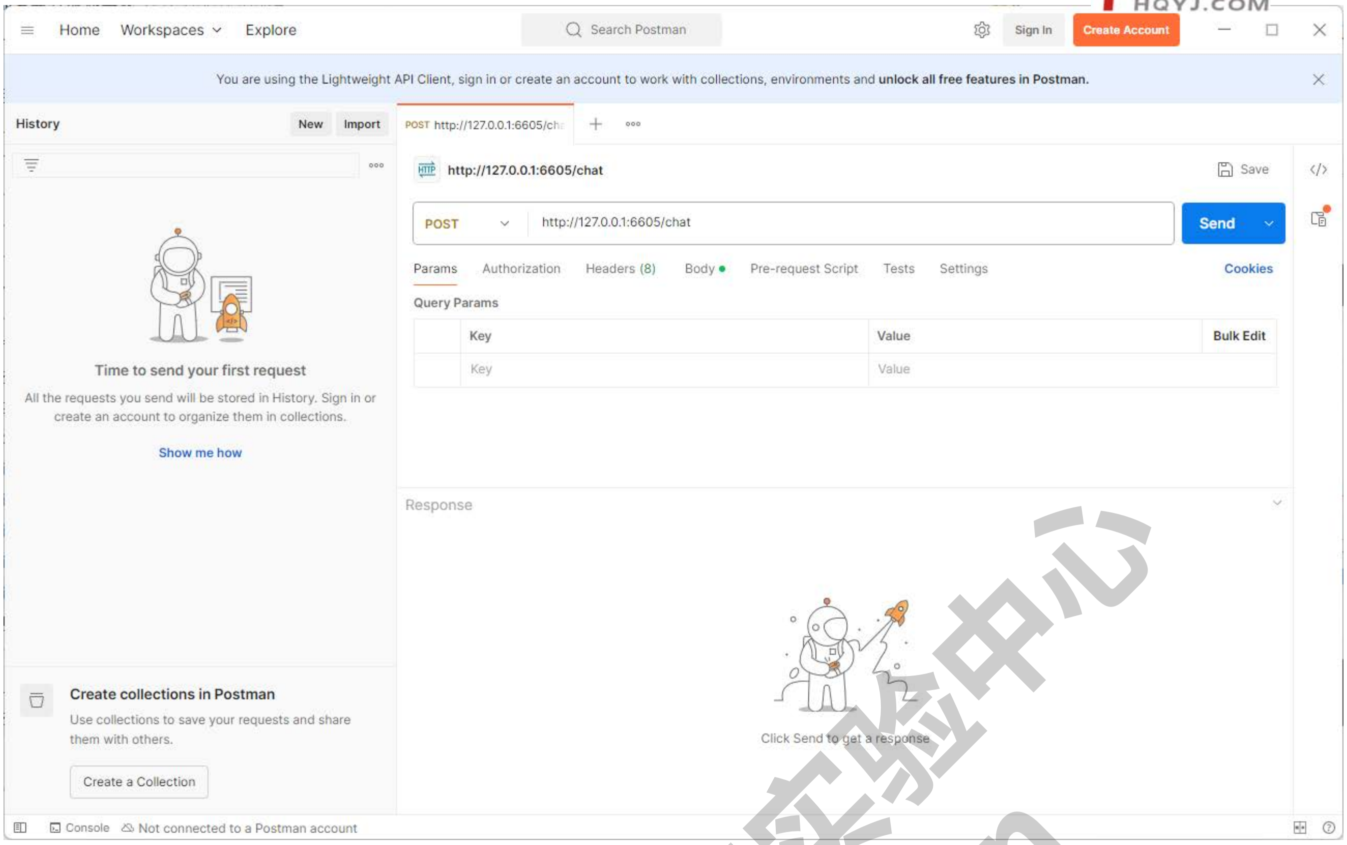
链接：https://www.postman.com/。

进入官网下载对应版本


Product
Pricing
Enterprise
Resources and Support
Public API Network
Search Postman
Contact Sales
Sign In
Sign Up for Free
Docu _
APIs together
Over 30 million developers use Postman. Get started by signing up or downloading the desktop app.
name@company.com
Sign Up for Free
Download the desktop app for
What is Postman?
Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.
API Tools
API Repository
Workspaces
Governance

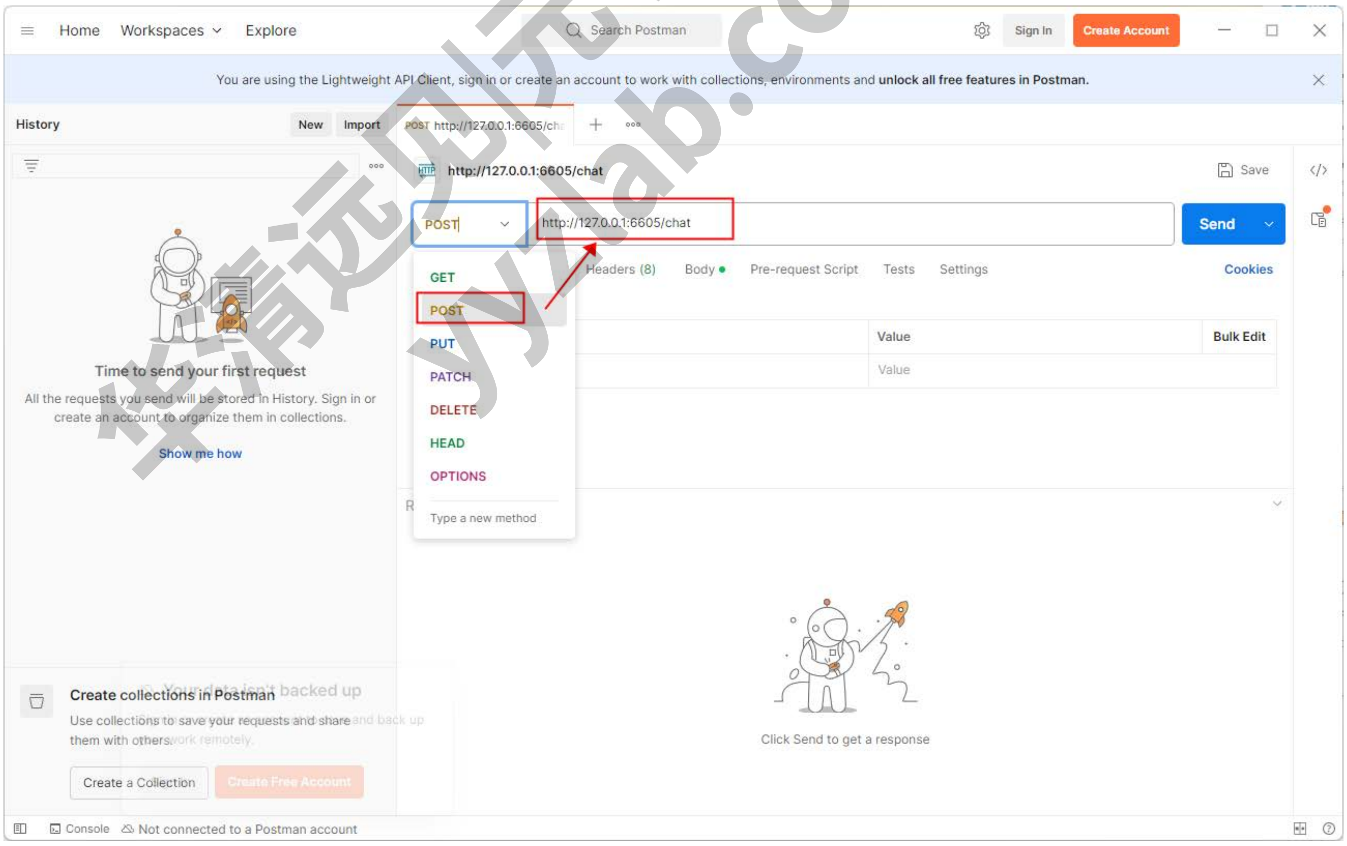
下载后打开文件，免安装免登录。




Home
Workspaces
Explore
Search Postman
Sign In
Create Account
You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman.
History
New
Import
POST http://127.0.0.1:6605/chat
http://127.0.0.1:6605/chat
Save
POST
http://127.0.0.1:6605/chat
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
Query Params
Key
Value
Bulk Edit
Time to send your first request
All the requests you send will be stored in History. Sign in or create an account to organize them in collections.
Show me how
Response
Click Send to get a response
Create collections in Postman
Use collections to save your requests and share them with others.
Create a Collection
Console
Not connected to a Postman account

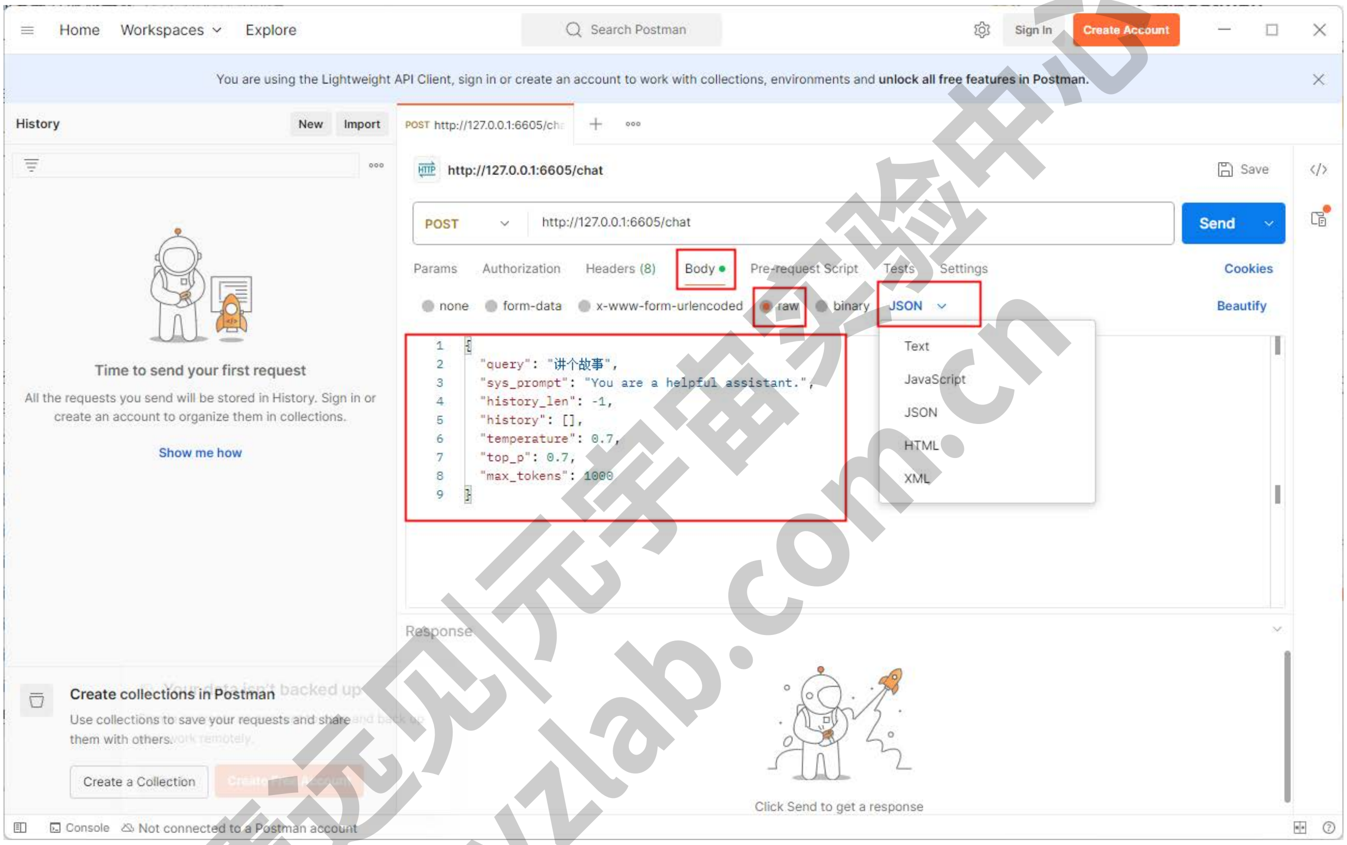
使用postman

选择对应的请求方式和url


Home
Workspaces
Explore
Search Postman
Sign In
Create Account
You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman.
History
New
Import
POST http://127.0.0.1:6605/chat
http://127.0.0.1:6605/chat
Save
POST
http://127.0.0.1:6605/chat
Send
GET
POST
PUT
PATCH
DELETE
HEAD
OPTIONS
Type a new method
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
Value
Bulk Edit
Time to send your first request
All the requests you send will be stored in History. Sign in or create an account to organize them in collections.
Show me how
Click Send to get a response
Create collections in Postman
Use collections to save your requests and share them with others.
Create a Collection
Console
Not connected to a Postman account

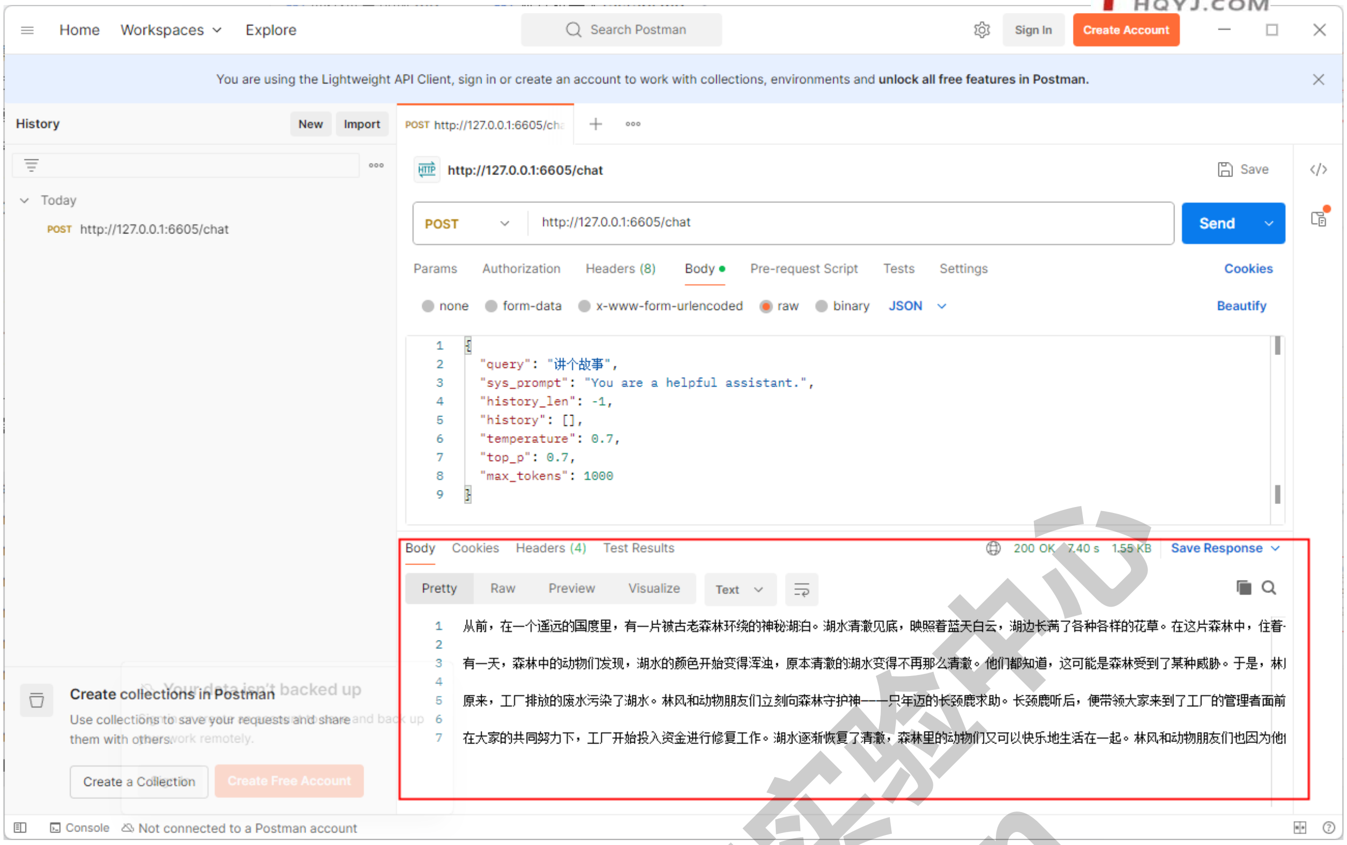
选择body->raw->json，填入请求体：

```
{
  "query":"讲个故事",
  "sys_prompt": "You are a helpful assistant.",
  "history_len": -1,
  "history":[],
  "temperature": 0.7,
  "top_p":0.7,
  "max tokens": 1000
}
```


Home
Workspaces
Explore
Search Postman
Sign In
Create Account
You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman.
History
New
Import
POST http://127.0.0.1:6605/chat
http://127.0.0.1:6605/chat
Save
POST
http://127.0.0.1:6605/chat
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
JSON
Beautify
Text
JavaScript
JSON
HTML
XML
Time to send your first request
All the requests you send will be stored in History. Sign in or create an account to organize them in collections.
Show me how
Create collections in Postman
Use collections to save your requests and share them with others.
Create a Collection
Response
Click Send to get a response
Console
Not connected to a Postman account

```
{
  "query": "讲个故事",
  "sys_prompt": "You are a helpful assistant.",
  "history_len": -1,
  "history": [],
  "temperature": 0.7,
  "top_p": 0.7,
  "max_tokens": 1000
}
```

点击发送send



You are using the Lightweight API Client, sign in or create an account to work with collections, environments and **unlock all free features in Postman.**

POST http://127.0.0.1:6605/chat

Body: raw, JSON

```
{
  "query": "讲个故事",
  "sys_prompt": "You are a helpful assistant.",
  "history_len": -1,
  "history": [],
  "temperature": 0.7,
  "top_p": 0.7,
  "max_tokens": 1000
}
```

Body   Cookies   Headers (4)   Test Results   200 OK   7.40 s   1.55 KB   Save Response

```
从前，在一个遥远的国度里，有一片被古老森林环绕的神秘湖泊。湖水清澈见底，映照着蓝天白云，湖边长满了各种各样的花草。在这片森林中，住着

有一天，森林中的动物们发现，湖水的颜色开始变得浑浊，原本清澈的湖水变得不再那么清澈。他们都知道，这可能是森林受到了某种威胁。于是，林

原来，工厂排放的废水污染了湖水。林风和动物朋友们立刻向森林守护神——一只年迈的长颈鹿求助。长颈鹿听后，便带领大家来到了工厂的管理者面前

在大家的共同努力下，工厂开始投入资金进行修复工作。湖水逐渐恢复了清澈，森林里的动物们又可以快乐地生活在一起。林风和动物朋友们也因为他
```

Create collections in Postman
Use collections to save your requests and share them with others.
Create a Collection
Create Free Account
Console
Not connected to a Postman account