

环境：

Vmware-----win10 专业版

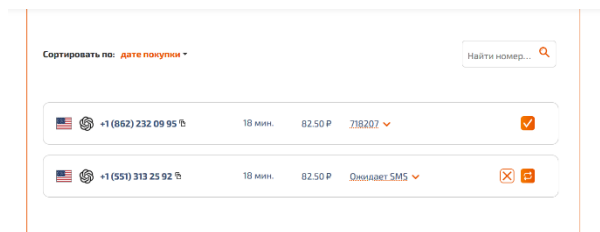
Python 3.8 VScode

V2ray-----freecat

Pip list

```
C:\Users\加菲>
C:\Users\加菲>pip list
Package           Version
-----
annotated-types    0.6.0
anyio               4.3.0
certifi             2024.2.2
colorama            0.4.6
distro              1.9.0
exceptiongroup      1.2.1
h11                 0.14.0
httpcore            1.0.5
httpx               0.27.0
idna                3.7
openai              1.26.0
pip                 24.0
pydantic            2.7.1
pydantic_core       2.18.2
setuptools          41.2.0
sniffio             1.3.1
tqdm                4.66.4
typing_extensions  4.11.0
```

1.一开始就遇到了 获取 key 需要验证身份，需要外网 phong 2 美刀-----密钥：sk-proj-npj1HvrWxQ35IfWjmmBkT3BlbkFJKQGgqZAuJVJ0WJG0SQHF



2 在虚拟机上安装 python3.8.0 以及 pycharm2024，且配对环境

3 使用 pip install openai 安装包

```
C:\Users\加菲>pip install openai
Collecting openai
```

```
C:\Windows\system32\cmd.exe
Downloading https://files.pythonhosted.org/packages/28/78/d31230046e58c207284c6b2c4e8d96e6d3cb4e52354721b944d3e1ee4aa5/annotated_types-0.6.0-py3-none-any.whl
Collecting pydantic-core==2.18.2 (from pydantic<3.>=1.9.0->openai)
Downloading https://files.pythonhosted.org/packages/05/43/7b662e634c8c92266d559ad9205335178b985ed26f35de0f24cebb0e12af/pydantic_core-2.18.2-cp38-none-win_amd64.whl (1.9MB)
1.9MB 6.4MB/s
Collecting certifi (from httpx<1.>=0.23.0->openai)
Downloading https://files.pythonhosted.org/packages/ba/06/a07f096c664aeb9f01624f858c3add0a4e913d6c96257acb4fce61e7de14/certifi-2024.2.2-py3-none-any.whl (163kB)
163kB 6.8MB/s
Collecting httpcore==1.* (from httpx<1.>=0.23.0->openai)
Downloading https://files.pythonhosted.org/packages/78/d4/e5d7e4f2174f8a4d63c8897d79eb8fe2503f7ecc03282feelfa2719c2704/httpcore-1.0.5-py3-none-any.whl (77kB)
81kB ...
Collecting colorama; platform_system == "Windows" (from tqdm>4->openai)
Downloading https://files.pythonhosted.org/packages/d1/d6/3965ed04c63042e047cb6a3e6ed1a63a35087b6a609aa3a15ed8ac56c221/colorama-0.4.6-py2.py3-none-any.whl
Collecting h11<0.15,>=0.13 (from httpcore==1.*->httpx<1.>=0.23.0->openai)
Downloading https://files.pythonhosted.org/packages/95/04/ff642e65ad6b90db43e668d70ffb6736436c7ce41fcc549f4e9472234127/h11-0.14.0-py3-none-any.whl (58kB)
61kB 4.1MB/s
Installing collected packages: exceptiongroup, sniffio, idna, typing_extensions, anyio, annotated-types, pydantic-core, pydantic, certifi, h11, httpcore, httpx, colorama, tqdm, distro, openai
Successfully installed annotated-types-0.6.0 anyio-4.3.0 certifi-2024.2.2 colorama-0.4.6 distro-1.9.0 exceptiongroup-1.2.1 h11-0.14.0 httpcore-1.0.5 httpx-0.27.0 idna-3.7 openai-1.26.0 pydantic-2.7.1 pydantic-core-2.18.2 sniffio-1.3.1 tqdm-4.66.4 typing_extensions-4.11.0
WARNING: You are using pip version 19.2.3, however version 24.0 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
C:\Users\加菲>
```

4 现在列表为:

```
C:\Users\加菲>
C:\Users\加菲>pip list
Package           Version
-----
annotated-types   0.6.0
anyio              4.3.0
certifi            2024.2.2
colorama           0.4.6
distro             1.9.0
exceptiongroup     1.2.1
h11                0.14.0
httpcore           1.0.5
httpx              0.27.0
idna               3.7
openai             1.26.0
pip                24.0
pydantic           2.7.1
pydantic_core      2.18.2
setuptools         41.2.0
sniffio            1.3.1
tqdm               4.66.4
typing_extensions  4.11.0
```

5 先在 VScode 中调试:

先按照官网

([https://platform.openai.com/docs/quickstart?\\_\\_cf\\_chl\\_rt\\_tk=ceqQyhSAIVLx7doDIkGfCTc\\_ETKxCZBYG2\\_triVQq9o-1715139381-0.0.1.1-1642](https://platform.openai.com/docs/quickstart?__cf_chl_rt_tk=ceqQyhSAIVLx7doDIkGfCTc_ETKxCZBYG2_triVQq9o-1715139381-0.0.1.1-1642)) 的流程来配置,

=====此处是将 openai-api-key 配置到系统的环境变量=====

报 `openai.APIConnectionError: Connection error` 错误

6.又按照 windows 本地调用方式:

1. 生成 key

2. 配置 key

(1) 生成.env 文件: (a) 首先新建一个空的.txt 文档, 可随意命名, 如 temp.txt;

(b) 在当前路径下打开终端（可用 cmd 打开再 cd 到当前路径，或直接在路径下打开），输入如下指令：

```
ren .\temp.txt .env
```

之后会生成.env 文件，然后在文件中写入：

```
OPENAI_API_KEY=your_key
```

保存退出。

## 7.事实证明,

上述方法并不管用。另谋他路，在 youtube 中找到了一个相对来说靠谱的教程来学习 <https://www.youtube.com/watch?v=9971sxBhEyQ>，跟随教程下载了 git-scm 来管理代码(也就是可以直接克隆 github 中的代码命令：git clone URL)

## 8.视频看完，问题依然存在

无梯子

penai.APITimeoutError: Request timed out.

开梯子

OpenAI 的。APIConnectionError：连接错误。本质原因：

因为 openai 不支持中国 ip，所以我们需要购买 vpn 来达到访问的目的，那出现这个问题很大原因就是自己买的 vpn 不对，只要你的 vpn 节点在 openai 支持的区域，就不存在类似的问题。

如果不行怎么处理：

分两种场景

场景一：购买的是 http 代理，那可能是节点区域不支持。

场景二：购买的是 socks 代理，也是不行的，因为 openai 默认发的是 http 请求。

打开自己的当前连接网络，看下网络代理的菜单。

这种就是 socks 代理。↓



场景一：如果是 http 代理且不行的，直接切节点，切到支撑 openai 的区域，比如老美

场景二：socks 代理的那只能改造下源码了，这部分网上方案一堆，去随便整个就行。

9.上述方法依然不可用，现在重新返回，另寻他路。在 openai->lib->site-packages->openai->type->\_base\_client.py->

```

class BaseClient(Generic[_HttpClientT, _DefaultStreamT]):
    _client: _HttpClientT
    _version: str
    _base_url: URL
    max_retries: int
    timeout: Union[float, Timeout, None]
    _limits: httpx.Limits
    _proxies: ProxiesTypes | None
    _transport: Transport | AsyncTransport | None
    _strict_response_validation: bool
    _idempotency_header: str | None
    _default_stream_cls: type[_DefaultStreamT] | None = None

    def __init__(
        self,
        *,
        version: str,
        base_url: str | URL,
        _strict_response_validation: bool,
        max_retries: int = DEFAULT_MAX_RETRIES,
        timeout: float | Timeout | None = DEFAULT_TIMEOUT,
        limits: httpx.Limits,
        transport: Transport | AsyncTransport | None,
        proxies: ProxiesTypes | None,
        custom_headers: Mapping[str, str] | None = None,
        custom_query: Mapping[str, object] | None = None,
    ) -> None:
        self._version = version
        self._base_url = self._enforce_trailing_slash(URL(base_url))
        self.max_retries = max_retries
        self.timeout = timeout
        self._limits = limits
        self._proxies = {'http': 'http://localhost:7890', 'https': 'http://localhost:7890'}
        self._transport = transport
        self._custom_headers = custom_headers or {}

```

中，将 BaseClient 类，把 init 中原本的 self.\_proxies=proxies 修改为图片中的内容，

再将自己的测试文件添加以下代码

```
import os
```

```
os.environ["http_proxy"] = "http://localhost:7890"
```

```
os.environ["https_proxy"] = http://localhost:7890
```

## 9.但是又出现新的问题

(现在不知道到底可不可以引入 API)，现在报错为访问请求限制。但不知道问题是否依然存在

## 1. 问题依旧