

ChatGPT接入微信

```
from flask import Flask, request, make_response
import requests
import hashlib
import xml.etree.cElementTree as ET
import time
# from WXBizMsgCrypt import WXBizMsgCrypt
import base64
import redis

app = Flask(__name__)

# 见下图
headers={'Authorization': 'Bearer **_*****'}
appid = '*****'
token = '*****'
encodingAESKey = '*****'
r = redis.Redis()

@app.get('/completions')
def sig():
    args = request.args
    print(args)
    signature = args['signature']
    echostr = args['echostr']
    timestamp = args['timestamp']
    nonce = args['nonce']
    arr = [token, timestamp, nonce]
    arr.sort()
    s = ''.join(arr)
    sha1 = hashlib.sha1(s.encode('utf-8')).hexdigest()
    if sha1 == signature:
        return echostr, 200
    return 'error', 500

@app.post('/completions')
def completions():
    t0 = time.time()
    xmlStr = request.stream.read()
    print(f'xmlStr = {xmlStr}')
    xmlData = None
    try:
        xmlData = ET.fromstring(xmlStr)
    except Exception:
        return '', 200
    msg_type = xmlData.find('MsgType').text
    if msg_type == 'text':
        ToUserName = xmlData.find('ToUserName').text
        FromUserName = xmlData.find('FromUserName').text
        Content = xmlData.find('Content').text
        json = {
```

```

        "model": "text-davinci-003",
        "prompt": Content.strip(),
        "max_tokens": 4000,
        "temperature": 1.0
    }
    key = str(FromUserName + ':' + Content.strip())
    print(f'question:{key}')
    reply = '''
        <xml>
        <ToUserName><![CDATA[%s]]></ToUserName>
        <FromUserName><![CDATA[%s]]></FromUserName>
        <CreateTime>%s</CreateTime>
        <MsgType><![CDATA[text]]></MsgType>
        <Content><![CDATA[%s]]></Content>
        </xml>
        '''

    answer = '没听清，再说一遍'
    cache = r.get(key)
    if cache:
        answer = cache.decode('utf-8')
        r.delete(key)
    elif cache is None:
        r.set(key, '', ex=60)
        text = ''
        try:
            res = requests.post('https://api.openai.com/v1/completions',
headers=headers, json=json).json()
            text = res['choices'][0]['text']
            print(f'res = {res}')
        except Exception as e:
            print(f'error:{e}')
            r.set(key, None)
            return '', 555
        t1 = time.time()
        if t1 - t0 > 4.5:
            r.set(key, text, ex=3600)
            return '', 555
        else:
            answer = res['choices'][0]['text']
    elif cache.decode('utf-8') == '':
        print('sleep')
        time.sleep(4.5)
        answer = f'思考中...\n过会儿再对我说一遍[{Content.strip()}]'
    else:
        return 'success'

    response = make_response(reply % (FromUserName, ToUserName,
str(int(time.time()))), answer.strip())
    response.content_type = 'application/xml'
    return response
else:
    return 'success'

if __name__ == '__main__':
    app.run(port = 9999)

```

appid：必填
token：必填
encodingAESKey：选填（当需要加密时必填）

网页分析

广告与服务

广告主

流量主

小程序管理

微信搜一搜

服务市场

设置与开发

公众号设置

人员设置

安全中心

违规记录

基本配置

开发者工具

运维中心

接口权限

公众号开发信息

开发者ID(AppID)

wx...

开发者ID是公众号开发识别码，配合开发者密码可调用公众号的接口能力。

开发者密码(AppSecret)

启用

开发者密码是校验公众号开发者身份的密码，具有极高的安全性。切记勿把密码直接交给第三方开发者或直接存储在代码中。如需第三方代开发公众号，请使用授权方式接入。也可使用微信云托管免服务器免运维，支持免鉴权调用公众号接口能力，无需启用开发者密码及配置IP白名单。

服务器配置(已启用)

服务器地址(URL)

http://...completions

令牌(Token)

h...

消息加解密密钥(EncodingAESKey)

i...

消息加解密方式

明文模式

修改配置

停用

已绑定的微信开放平台帐号

扫码接收重要通知

接收通知

OverviewDocumentationExamplesPlayground

UpgradeHelpPersonal

ORGANIZATION

Personal

Settings

Usage

Members

Billing

USER

API Keys

API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we've found has leaked publicly.

SECRET KEY	CREATED	LAST USED	
sk-...Vh0s	2022年12月9日	Never	
sk-...S61J	2023年2月7日	2023年2月8日	

+ Create new secret key

Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

Personal

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

SECRET KEY：必填 ({'Authorization': 'Bearer空格SECRET KEY'})