个人号：发送信息

公众号：推广营销，

服务号与好友同级，微信支付功能

# 介入流程

1. 填写信息--测试平台，应先有响应信息 微信默认访问80端口且不能修改

微信公众平台，填写URL，token，接口

1. 服务器响应验证信息(GET)
2. **定义常量**

WX\_TOKEN = “IT”

1. **接收4个值**

def get(self):

signature = get\_argument(“signature”)

timestamp,

nonce,

echostr

1. **字典排序（按字母排序）**

tem = [WX\_TOKEN, timestamp, nonce] #放入一个列表

tem.sort() # 在原字典排序

#sorted(tem) # 返回一个排序的字典

tmp = “”.join(tem) # 拼接字符串

1. **进行hash**

realsignature = hashlib.sha1(tmp).hexdigest() #16进制形式

1. **对比**

if signature == realsignature:

self.write(echostr) # 验证为微信服务器，返回echostr

else:

#退出 self.send\_error(403)

1. **配置nginx**

location / {

proxy\_ pass\_header Server

proxy\_set\_header Host $http\_host

proxy\_redirect off;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Scheme $scheme;

proxy\_pass <http://127.0.0.1:8001;>

}

# 微信类型

有音频文本等类型

# 解析xml

import xmltodict

dict\_data = xmltodict.parse(xml\_data)

msg = dict\_data[‘xml’][‘MsgType’]

if msg == ‘text’:

content = dict\_data[‘xml’][‘content’]

resp\_data = {

“xml”:{

“ToUserName”:dict\_data[‘xml’][‘FromUserName’],

“FromUserName”:dict\_data[‘xml’][‘ToUserName]

“CreateTime”:int(time.time()),

“MsgType”:”Text”, #文本方式

“Content”:Content

}

}

self.write(xmltodict.unparse(resp\_data)

else：

# 因为微信要求必须有回应，如若不回应，5秒后会再次发送

resp\_data = {

“xml”:{

“ToUserName”:dict\_data[‘xml’][‘FromUserName’],

“FromUserName”:dict\_data[‘xml’][‘ToUserName]

“CreateTime”:int(time.time()),

“MsgType”:”Text”, #文本方式

“Content”:Content

}

}

self.write(xmltodict.unparse(resp\_data)

# 关注、取消

# 调用接口凭据

access\_token2个小时有效期

接口地址：微信文档

方式GET

grant\_type,appid,secret

1\新建class

class AccessToken(object):

\_access\_token = None

\_create\_time = None

\_expires\_in = None

@classmethod

def get\_access\_token(cls): # 判断token有效期

nowtime = time.time() - (cls.\_expires\_in - 200)#因为程序有延迟

if nowtime > cls.\_expires\_in:

# 向微信服务器请求的access\_Toke

cls.update\_access\_token()

return cls.\_access\_token

else:

return cls.\_access\_token

2\定义

@classmethod

def update\_access\_token(cls): # token过期，就更新token

client = AsyncHTTPClient() # 导入异步服务

发送请求，换取access\_token,expires\_in,create\_in

更新本来变量

创建二维码

创建二维码的

通过get\_access\_token()获取

req = tornado.HTTPCline.HTTPRequest(

url=url,method=post,body=body

);

clint.fetch(req)

二维码带参数

if “EventKey” in dict\_data[‘xml’]:

event\_key = dict\_data[‘xml’][‘eventkey’]

sense\_id = eventkey[8:] #微信是由前缀的值

# 微信网页授权

1. 用户同意授权，获取code
2. 通过code换取access\_token
3. 拉取用户信息

条件1:配置域名

class ProfileHandler(RequestHandler):

def get(self):