respeakerV2 + Tencent 发短信信息

简介.

基于腾讯云消息服务，在国内它比Twilio发送信息速度更快，成功率高达99% .

作为测试用户通过手机免费申请后每个月有100条免费短信，但每一条短信只有

140个字符（包括标点，空格），超出就会被默认为两条信息。

Client须知：

国内短信云平台（腾讯，阿里等）都是基于模板发送信息，所谓模板大意是指在云端建立一个发送模版，用户(client)发送的信息内容必须和你建立的模版完全相同（包括标点符号，空格），如用户(client)发送短信内容不相同你将会得到一个执行错误提示” 模版未审批或内容不匹配”.

当然如果你需要改变的部分内容，可以通过{1},{2},…的占位符方式来获取需要改变的内容，但每一个占位符只能是一个字或者一个不包含空格的字符串，这样才能保证与模版内容完全相同。

腾讯云注册流程：

短信服务：拷贝复制网址到浏览器：<https://cloud.tencent.com/>



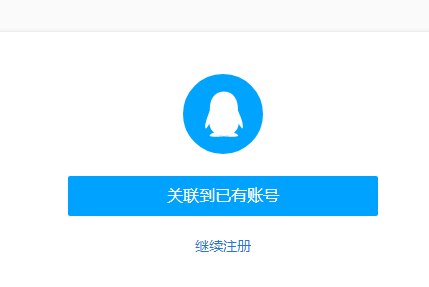
点击 注册:



可以选择以上任一种方式进行注册：



点击 登录：



点击 继续注册：





填写手机号和验证码，并同意协议：



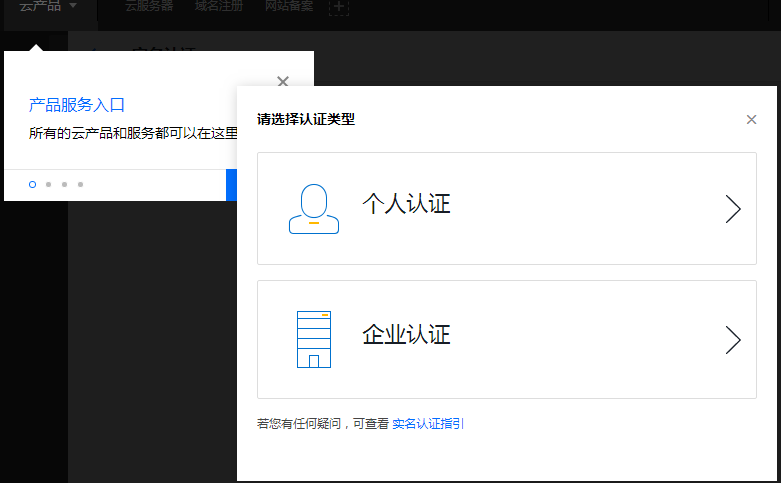
选择 跳过：



选择 跳过：



点击 去实名认证：



选择认证类型：如果有疑问请看实名认证指引我相关详细的介绍：

**个人实名认证**：



绑定.为了认证您的身份，请用绑定了本人银行卡的手Q扫描二维码。本认证方式不扣除任何费用

然后提交审核，审核通过后就可以继续下一步操作，如果没有通过也没关系可以在后续操作中进行验证：

在腾讯首页选择产品-🡪短信。



点击短信;



点击免费试用：

![C:\Users\hong\AppData\Roaming\Tencent\Users\422363669\QQ\WinTemp\RichOle\4_K6`8V](L8SDX}}UDQ%%QP.png](data:image/png;base64,)

点击 开始接入：



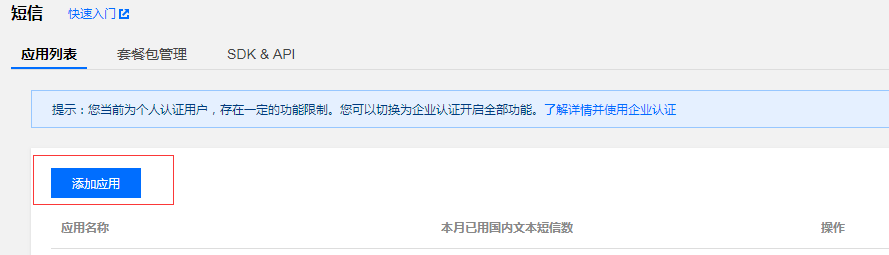
去进行相关认证：点击实名认证：和上面认证过程相同 ：

完成后。



短息设置：

点击添加应用：

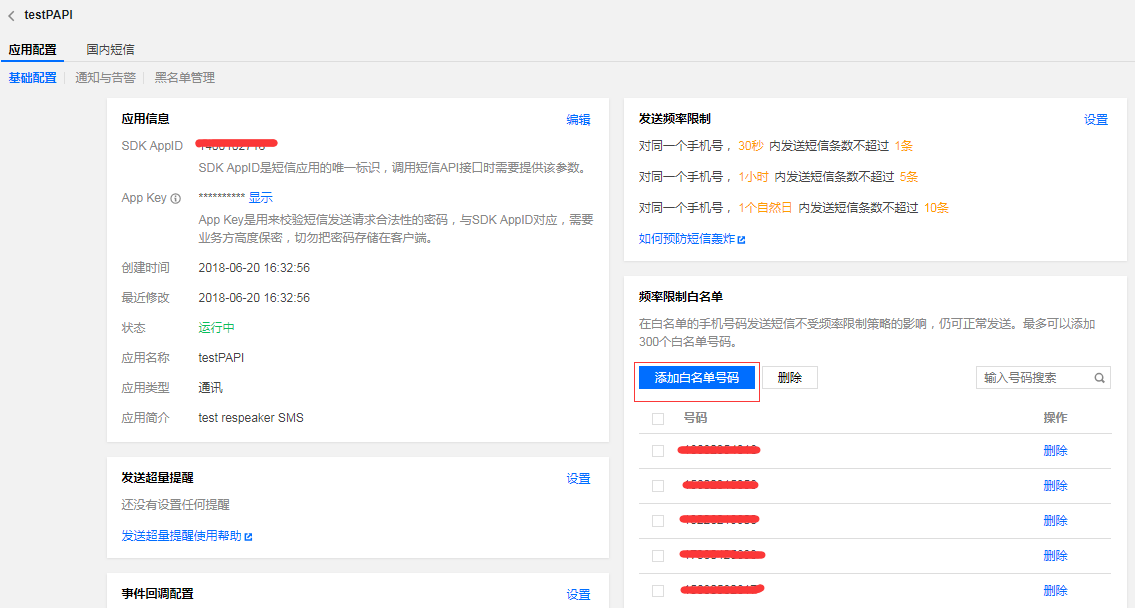




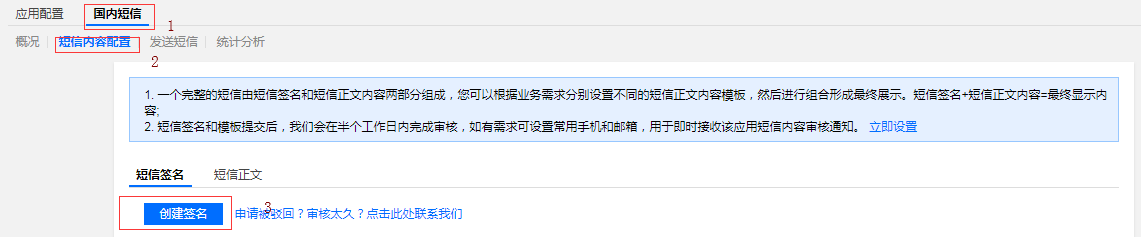
完成后：



点击 你的应用：(testPAPI) ，应用配置 ->基础配置->添加白名单号码：



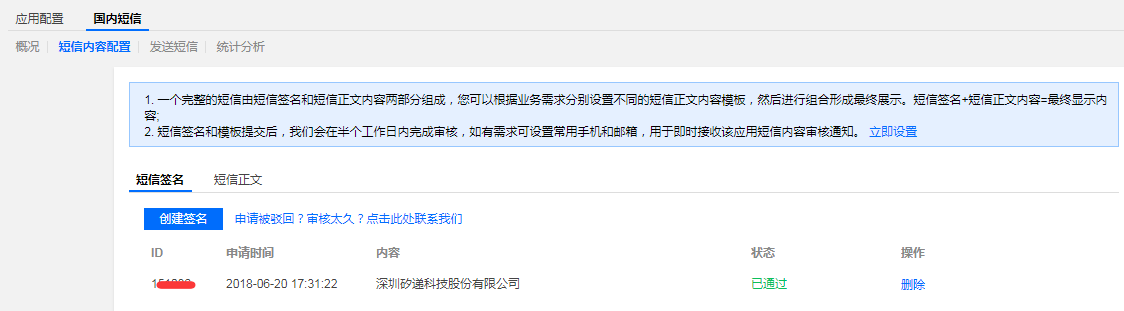
点击 国内短信->短信内容配置->创建签名：



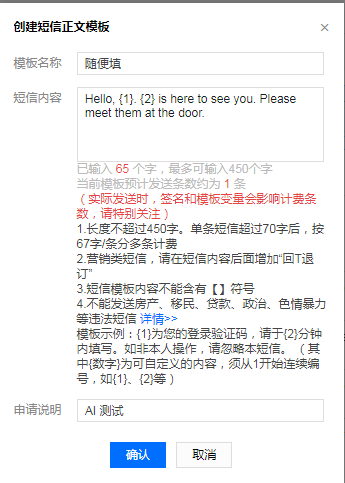
填写相关信息：

![C:\Users\hong\AppData\Roaming\Tencent\Users\422363669\QQ\WinTemp\RichOle\NV9LT[C$UXX}ZYDB2N42](Y.png](data:image/png;base64,)

点击确认：审核通过后：



点击 国内短信->短信内容配置->短信正文->创建正文->创建正文模板



内容：

Hello, {1}. {2} is here to see you. Please meet them at the door.

其{1},{2}是占位符，是需要传入的参数；

如：Hello， search. management is here to see you. Please meet them at the door.

审核通过后如：



模板创建成功后就可以进行发送短信了。

安装准备：

安装qcloudsms\_py

pip install qcloudsms\_py

代码如下：python

#!/usr/bin/python

# coding:utf-8

# import Speech SDK

'''

文件名 : speach.py

支持百度中英语音，中文识别率较高，但英语识别率不理想，

若想用中文请打开宏开关 ，中英文切换，true = zh , false = en

同时改变百度为云端解析

bing /baidu 宏开关 False=baidu ,True=bing

'''

from evdev import InputDevice,categorize,ecodes

from pypinyin import pinyin, lazy\_pinyin

from aip import AipSpeech

from ctypes import \*

from pixel\_ring import pixel\_ring

from respeaker.bing\_speech\_api import BingSpeechAPI,RequestError

from twilio.rest import Client

from twilio.twiml.voice\_response import VoiceResponse

from voice\_engine.kws import KWS

from voice\_engine.ns import NS

from voice\_engine.source import Source

from voice\_engine.channel\_picker import ChannelPicker

from qcloudsms\_py import SmsSingleSender

from qcloudsms\_py.httpclient import HTTPError

from voice\_engine.element import Element

import mraa

import logging

import ConfigParser

import threading,signal

import time

import sys,stat

import os

import re

import json

import copy

if sys.version\_info[0] < 3:

import Queue as queue

else:

import queue

is\_pressed = False

is\_record = False

is\_music = False

is\_noteinfo = False

is\_callflags = False

flags = 0

gfd = -1

led\_ring = False

parse\_flags = False

key\_time\_zh = 2

key\_time\_en = 4

starttime = 0

lasttime = 0

recvok = False

stringcopy = ''

'''

Twilio Message

'''

'''

baidu key

'''

BING\_KEY = 'e8816ea657c643ecab1f0a66ddf121ae'

'''

中英文切换，true = zh , false = en

'''

#zh\_en\_on = False

zh\_en\_on = True

'''

宏开关

'''

TREAD\_ON = 1

ON = 0

'''

bing /baidu 宏开关 False=baidu ,True=bing

'''

BING\_BAIDU\_ON = True

#BING\_BAIDU\_ON = False

'''

led

'''

en = mraa.Gpio(12)

'''

bing api

'''

def bing\_api\_call():

logging.basicConfig(level=logging.DEBUG)

'''

defined

http:

'''

APP\_ID = '11203359'

API\_KEY = 'wG9Iqfg2OplwQmXQaM5fxzRY'

SECRET\_KEY = '2VuEDoSSi7rgFUggYSfzrTeQ4YGLhSlW'

'''

tencent api

'''

appid = 1400102718

appkey = "214b971761479e894e5fd27227d4e936"

'''

init AipSpeech object

'''

aipSpeech = AipSpeech(APP\_ID, API\_KEY, SECRET\_KEY)

'''

led ring

'''

en.dir(mraa.DIR\_OUT)

class Bing(Element):

def \_\_init\_\_(self, key):

super(Bing, self).\_\_init\_\_()

self.key = key

self.queue = queue.Queue()

self.listening = False

self.done = False

self.event = threading.Event()

self.bing = BingSpeechAPI(BING\_KEY)

def put(self, data):

if self.listening:

self.queue.put(data)

def start(self):

self.done = False

thread = threading.Thread(target=self.run)

thread.daemon = True

thread.start()

def stop(self):

self.done = True

def listen(self):

self.listening = True

self.event.set()

def run(self):

while not self.done:

self.event.wait()

def gen():

count = 0

while count < 16000 \* 6:

data = self.queue.get()

if not data:

break

yield data

count += len(data) / 2

#recognize speech using Microsoft Bing Voice Recognition

try:

# text = bing.recognize(gen(), language='zh-CN')

text = self.bing.recognize(gen())

global stringcopy

print('Bing:{}'.format(text).encode('utf-8'))

stringcopy = format(text).encode('utf-8')

except ValueError:

print('Not recognized')

global is\_music

is\_music = False

time.sleep(1)

#play\_other('mpg123 /home/respeaker/twilio/notfind.mp3')

play\_other('mpg123 /home/respeaker/twilio/not\_clear.mp3')

except RequestError as e:

print('Network error {}'.format(e))

time.sleep(1)

play\_other('mpg123 /home/respeaker/twilio/net\_error.mp3')

self.listening = False

self.event.clear()

self.queue.queue.clear()

global recvok,is\_record

recvok = True

is\_record = True

def recv\_string():

global recvok

while recvok != True:

time.sleep(0.1)

recvok = False

print(stringcopy)

return stringcopy

'''

Twilio,no used

'''

def sendmessage(employee,gestName,employeeNumber):

"""

Some example usage of different twilio resources.

"""

client = Client('AC126c5beb95795762e545c22121d61d24', '4d036c85d72c9de3005fd757d1d8663a')

# Get all messages

all\_messages = client.messages.list()

print('There are {} messages in your account.'.format(len(all\_messages)))

# Get only last 10 messages...

some\_messages = client.messages.list(limit=10)

print('Here are the last 10 messages in your account:')

for m in some\_messages:

print(m)

# Get messages in smaller pages...

all\_messages = client.messages.list(page\_size=10)

print('There are {} messages in your account.'.format(len(all\_messages)))

print('Sending a message...')

print('+86'+employeeNumber)

print('Hello,'+employee+'.'+gestName+' is here to see you. Please meet them at the door.')

#Hello, <employee>. <guest> is here to see you. Please meet them at the door.

new\_message = client.messages.create(to='+86'+employeeNumber, from\_='+15109014046', body='Hello,'+employee+'.'+gestName+' is here to see you. Please meet them at the door.')

def tencentSendMesaage(employee,gestName,employeeNumber):

## Enum{0: 普通短信, 1: 营销短信}

sms\_type = 0

template\_id = 151036

params = []

params.append(employee)

params.append(employeeNumber)

ssender = SmsSingleSender(appid, appkey)

try:

sendStr = "Hello,"+employee+"."+gestName+" is here to see you. Please meet them at the door."

print(sendStr)

result = ssender.send(sms\_type, 86, employeeNumber,sendStr)

#result = ssender.send(sms\_type, 86, employeeNumber,"Hello,search.jcr is here to see you. Please meet them at the door.")

# 签名参数未提供或者为空时，会使用默认签名发送短信

#result = ssender.send\_with\_param(86, employeeNumber,template\_id, params)

except HTTPError as e:

print(e)

except Exception as e:

print(e)

print('tencnet:{}'.format(result).decode('unicode\_escape'))

def led\_pixel\_ring():

en.write(0)

pixel\_ring.set\_brightness(20)

while led\_ring :

try:

pixel\_ring.wakeup()

time.sleep(0.1)

pixel\_ring.off()

time.sleep(0.1)

pixel\_ring.off()

except KeyboardInterrupt:

break

pixel\_ring.off()

en.write(1)

'''

list

'''

def convert\_to\_list():

result=[]

fd = file("/home/respeaker/twilio/allname.txt", "r")

for line in fd.readlines():

result.append(''.join(list(line.lower().rstrip().split(','))))

'''

for item in result:

for it in item :

print(it)

'''

fd.close()

return result

def fuzzyfinder(parse\_name,result):

suggestions = []

pattern = '.\*'.join(parse\_name) # Converts 'tes' to 't.\*e.\*s'

regex = re.compile(pattern) # Compiles a regex.

for item in result:

for it in item:

match = regex.search(it) # Checks if the current item matches the regex.

if match:

suggestions.append(it)

print(suggestions)

return suggestions

'''

parse local file

'''

def search\_user(user):

print(user[0:len(user)-3])

cf = ConfigParser.ConfigParser()

cf.read("/home/respeaker/twilio/username.txt")

is\_ = cf.has\_option("db", user[0:len(user)-3])

if is\_ :

db\_phone = cf.get("db", user[0:len(user)-3])

if db\_phone.isdigit() :

return db\_phone

else:

return 0;

else:

return 0;

'''

parse local file

'''

def search\_user\_en(user):

print(len(user))

print(user)

cf = ConfigParser.ConfigParser()

cf.read("/home/respeaker/twilio/username\_en.txt")

is\_ = cf.has\_option("db", user)

if is\_ :

db\_phone = cf.get("db", user)

if db\_phone.isdigit() :

return db\_phone

else:

return 0;

else:

return 0;

def bing\_hande(self):

text = ""

bing = BingSpeechAPI(key=BING\_KEY)

try:

fd = open(self.path)

conect = fd.read(-1)

text = bing.recognize(conect)

fd.close()

return text

except Exception as e:

print(e.message)

return text

def parse\_array(text):

array =[]

array = text.lower().split(' ')

return array

def parse\_array\_noneuplow(text):

array =[]

array = text.split(' ')

return array

def filter\_re(string):

names = re.search(r'.\*?[^A-Z]\.', string).group(0)

return map(lambda x: x.replace('and', '').replace('et al','').strip(), names.rstrip('.').split(','))

'''

百度API云端处理

'''

class baidu\_bing\_aip\_speech(object):

def \_\_init\_\_(self,\_path,\_format):

self.path = \_path

self.format = \_format

'''

read file

'''

def get\_file\_content(self):

with open(self.path, 'rb') as fp:

return fp.read()

def bing\_\_parse\_speech(self):

global is\_music,is\_find,count,stringcopy

try:

bing.listen()

text = recv\_string()

stringcopy = ''

print('Recognized text %s' % text)

if text:

print('Recognized %s' % text)

is\_find = False

array = parse\_array(text)

print("---------------------------\n")

convert\_lis = convert\_to\_list()

'''

for strd in convert\_lis:

print(strd)

print("\n")

'''

# start parse voice

for i in array:

for employee in convert\_lis:

if i == employee :

#do what you want to do

print(employee)

print("\n")

is\_find = True

is\_music = False

play\_request\_thanks()

count=2

while(count):

bing.listen()

time.sleep(0.1)

gestName = recv\_string()

stringcopy = ''

is\_music = True

is\_music = False

getGestName = ''

if gestName:

result = []

str\_list = parse\_array\_noneuplow(gestName)

print(len(str\_list))

list\_len = len(str\_list)

if list\_len == 1:

getGestName = str\_list[list\_len-1]

elif list\_len == 2:

getGestName = str\_list[0]+'.'+str\_list[1]

elif list\_len >= 3:

getGestName = str\_list[list\_len-2]+'.'+str\_list[list\_len-1]

print(getGestName)

time.sleep(0.1)

play\_request\_sendMessage()

time.sleep(0.1)

employeeNumber = search\_user\_en(''.join(employee).lower())

print(employeeNumber)

#sendmessage(employee,getGestName,employeeNumber)

tencentSendMesaage(employee,getGestName,employeeNumber)

print("\*\*\*\*\*\*\*\*Bing\_English\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

count = 0

return 0

else:

count = count -1

if count == 0 :

return 0

if is\_find == False:

#Oh, sorry, I didn't find it.

is\_music = False

play\_other('mpg123 /home/respeaker/twilio/notfind.mp3')

else:

return -1

except Exception as e:

print(e.message)

'''

call 调用

'''

def pre\_play(v):

global is\_pressed,is\_music,is\_record,is\_noteinfo,led\_ring,parse\_flags

while True:

if is\_pressed == True:

'''

Welcome to x.factory. Who are you here to see?

'''

os.system("mpg123 /home/respeaker/twilio/welcome.mp3")

is\_pressed = False

try:

'''

数据解析及电话连线使能

'''

#global is\_music

is\_music = True

par = baidu\_bing\_aip\_speech('/home/respeaker/twilio/test.wav', 'wav')

'''

baidu api call

'''

if par.bing\_\_parse\_speech() == 0 :

is\_music = False

parse\_flags = False

else:

parse\_flags = True

is\_music = False

except Exception,e:

is\_music = False

is\_noteinfo = False

print e.message

led\_ring = False

time.sleep(0.1)

'''

访客按键处理

'''

def key\_hander(v):

global is\_pressed,is\_noteinfo,led\_ring,starttime,lasttime

key = InputDevice("/dev/input/event0")

for event in key.read\_loop():

if event.type == ecodes.EV\_KEY:

print(categorize(event))

'''

hold time

'''

if event.value == 2 :

lasttime = event.sec

sp\_time = lasttime - starttime

'''

down

'''

elif event.value == 1 :

starttime = event.sec

'''

up

'''

else:

led\_ring = True

is\_pressed = True

is\_noteinfo = True

starttime = 0

lasttime = 0

'''

数据解析及电话连线时提示音

'''

def play\_music(v):

global is\_music

while True:

time.sleep(1)

def play\_request\_thanks():

'''

Thank you. What is your name?

'''

os.system("mpg123 /home/respeaker/twilio/Thanks.mp3")

def play\_other(strInfo):

'''

Thank you. What is your name?

'''

os.system(strInfo)

def play\_request\_sendMessage():

'''

Thank you. I will send them a message

'''

os.system("mpg123 /home/respeaker/twilio/message.mp3")

'''

CTRL + C

'''

def CtrlC(signum, frame):

os.kill(os.getpid(),signal.SIGKILL)

'''

main start ...

'''

if \_\_name\_\_ == "\_\_main\_\_":

try:

src = Source(channels=8)

ch0 = ChannelPicker(channels=src.channels, pick=0)

ns = NS()

bing = Bing(BING\_KEY)

src.pipeline(ch0, ns, bing)

bing\_api\_call()

'''

调试时CTRL + C 处理

'''

signal.signal(signal.SIGINT, CtrlC)

signal.signal(signal.SIGTERM, CtrlC)

thread\_key = threading.Thread(target=key\_hander,args=(1,))

thread\_play = threading.Thread(target=pre\_play, args=(1,))

thread\_music = threading.Thread(target=play\_music, args=(1,))

thread\_key.setDaemon(True)

thread\_play.setDaemon(True)

thread\_music.setDaemon(True)

thread\_key.start()

thread\_play.start()

thread\_music.start()

src.recursive\_start()

if TREAD\_ON == ON :

thread\_key.join()

thread\_play.join()

thread\_music.join()

else :

while True:

led\_pixel\_ring()

time.sleep(1)

pass

'''

异常处理

'''

except Exception,exc:

print exc