

知乎

首发于
乙醚的计算机讲堂

从零开始微信机器人（二）：使用图灵机器人和api.ai相关接口



乙醚

私信不保证回复，有于你重要的问题，值乎见。

15 人赞了该文章

本文最初发布于简书：[从零开始微信机器人（二）：使用图灵机器人和api.ai相关接口](#)本文Github源代码地址：[locoda/connector-wechat-bot](#)（代码适用版本为python 3.6）

图灵机器人相关接口

图灵机器人是一个中文语境下的对话机器人，免费的机器人每天有5000次调用的，如果放在群聊中是完全够用的（如果只有@的消息才使用机器人回复的）。图灵机器人也包括一些简单的能力，比如讲笑话、故事大全、成语接龙、新闻资讯等，我们将介绍如何简单调用图灵机器人接口。

前期准备

1. 前往注册[图灵机器人](#)，增加一个机器人，并记录机器人的APIKey。具体注册方法可以前往[图灵API](#)查看。（如果你觉得很麻烦，也可以暂时使用itchat提供的几个key）
2. 安装[Requests: HTTP for Humans](#)

```
# 从 pip 安装 ()
pip install requests
```

并在机器人中导入Requests包（写在程序最初）：

```
import requests
```

调用接口

wxpy提供了图灵的接口，使用方法：

```
tuling = Tuling
```

▲ 赞同 15 ▼

● 11 条评论

➦ 分享

★ 收藏

...

从零开始微信机器

知乎

首发于
乙醚的计算机讲堂

tuling.do_reply(msg)

为了让大家了解HTTP协议在python中的使用方式，我接下来会介绍如何使用请求获取信息。

本节中内容如果没有特殊提示，都应写在你希望处理的某种注册了的消息方法中以保证它的正常运作。

首先，将图灵API写入程序中：

```
TULING_TOKEN = 'Your API Key'
```

然后，定义接口链接和需要传输的数据：

```
url_api = 'http://www.tuling123.com/openapi/api'
data = {
    'key' : TULING_TOKEN,
    'info' : msg.text, # 收到消息的文字内容
}
```

根据文档，通过HTTP请求，我们将会得到一个json格式的文件。使用Requests包，我们可以简单的获得调用接口所返回的信息：

```
s = requests.post(url_api, data=data).json()
print s # 打印所获得的json查看如何使用
# {u'text': u'回复的内容', u'code': 100000}
```

我们发现经过请求，我们一般会得到一个字典内容，其中包括text和code两项：text是图灵机器人回复的文本，而code是返回的编号。详细的返回数据格式也可以在[图灵API](#)中看到，除了文字类还有新闻类、图片类、链接类等返回类型。在这里我们以文字类为例，介绍如何处理：

```
if s['code'] == 100000:
    print s['text'] # 查看回复消息的内容，可省略
    msg.reply(s['text']) # 回复消息
```

如果需要回复其他类型的消息，也完全可以通过判断code确定消息类型，再决定如何回复。这里给出我的回复方法供大家参考（也可以选择不处理这一类内容）：

```
if s['code'] == 200000: # 链接类：回复文字和链接
    msg.reply(s['text'] + s['url'])
```

至此，我们已经

▲ 赞同 15 ▼

● 11 条评论

➤ 分享

★ 收藏

...

从零开始微信机器

知乎

首发于
乙醚的计算机讲堂

```

from wxpy import *
import requests

TULING_TOKEN = 'Your API Key'
bot = Bot()

@bot.register(Group, TEXT) # 这里注册了群聊中的文字消息，测试时可以设置为自己(上篇中提到过)
def group_msg(msg):
    if msg.is_at:
        url_api = 'http://www.tuling123.com/openapi/api'
        data = {
            'key' : TULING_TOKEN,
            'info' : msg.text, # 收到消息的文字内容
        }

        s = requests.post(url_api, data=data).json()
        print s # 打印所获得的json查看如何使用

        if s['code'] == 100000:
            print s['text'] # 查看回复消息的内容，可省略
            msg.reply(s['text']) # 回复消息

embed()

```

以下内容更加进阶，而文末有一些简单问题的解答。如果遇到其他问题，我也会在之后更新。

番外：使用上下文

wxpy给每个用户定义了一个相对稳定的对象/用户id，为puid，可以始终被获取到并有唯一的稳定性（根据文档），我们可以使用这个id来作为userid传给图灵机器人，以方便识别机器人或航班/列车信息的上下文。

```
bot.enable_puid() # puid 需要手动开启，请将这句话写在登陆登录之后
```

这样传送给接口的数据也要同时修改为：

```

data = {
    'key' : TULING_TOKEN,
    'info' : msg.text, # 收到消息的文字内容
}

```

▲ 赞同 15 ▼

● 11 条评论

➤ 分享

★ 收藏

...

从零开始微信机器

知乎

首发于
乙醚的计算机讲堂

这样做的好处是，图灵机器人可以根据得userid来获取上下文信息。例如你询问『天气』，它会回复『亲爱的，悄悄地告诉我你在哪个城市？』。在这种情况下，如果你不使用userid参数，你再次回复城市，图灵机器人也无法正确找到天气；如果你使用了这一参数，且两次回复使用的userid相同，图灵机器人会为你回复你回复的城市的天气情况，完成这一对话。

使用api.ai

api.ai是一家被谷歌收购的人机交互系统，主要着重于对话机器人的开发。图灵机器人虽然包括一个知识库，但其语义识别的能力较差。我所需要的机器人主要用于新生群，很多问题人与人之间会有相当多不同的表达，图灵机器人无法满足我对于群聊机器人的要求。因此，我尝试使用api.ai进行回复有针对性的一些问题。

如果你的英语相对糟糕，我不建议使用api.ai。api.ai的配置大多需要使用英语，虽然接口简单，但是后台设置相对复杂，如果没有英文背景不推荐使用。

这一部分内容相对进阶，如果没有特殊需要，完全可以跳过不看。这里只作一个对api.ai使用方式上大体的介绍，希望能帮助大家了解这一网站。

前期准备

1. 前往[api.ai](#)注册，创建机器人，并获得APIkey。虽然被谷歌收购，但是这个网站是不需要翻墙的呢！
2. 安装api.ai官方提供的Python SDK

```
pip install apiai
```

3. 在文件头部加入（处理返回的消息时使用）

```
import json
```

调用接口

首先，我们需要设置api.ai的Token，

```
APIAI_TOKEN =
```

▲ 赞同 15 ▼

● 11 条评论

➤ 分享

★ 收藏

...

从零开始微信机器



```
ai = apiai.ApiAI(APIAI_TOKEN)
request = ai.text_request()
request.lang = 'zh-CN' # 使用中文
request.session_id = msg.member.puid # api.ai 中用 session id 来区分对话对象，必须
request.query = msg.text # 消息文字内容
```

然后通过接口得到传回的json：

```
response = request.getresponse()
s = json.loads(response.read(), encoding='UTF-8') # 讲传回的json转换为python字典
print s
# {'lang': u'zh-cn', u'status': {'errorType': u'success', u'code': 200}, u'timestamp': ...}
```

我们发现，api.ai传回的json相对于图灵机器人更加复杂。参考[api.ai的query文档](#)，我对这部分回复进行了如下处理：

```
if s['result']['action'] == 'input.unknown': #
    raise Exception('api.ai cannot reply this message') # 抛出异常：使用 try 语句捕捉后
if s['status']['code'] == 200:
    msg.reply(s['result']['fulfillment']['speech']) # 回复 api.ai 返回的内容
```

api.ai的设置和调试

在进入api.ai的机器人后，你将会看到左边的多个菜单。

如果你只是简单的需要特定语句回复的功能，只需要创建并设置Intents就可以实现。在User Says一栏中填写消息可能是什么，在下方Response处填写可能回复的内容，然后保存即可。

api.ai自带机器学习功能，它的参数可以在机器人设置中的ML settings里找到。可以通过调整参数和方式让你的机器人回复更加准确。

在进入机器人后，api.ai的右侧会出现一个对话框。你可以使用它进行一些基础调试。在上方输入你的消息后，下方会给出机器人的回复，你可以通过这个对话框来了解是否正确设置了机器人。（需要翻墙）

本文中我们介绍了接口的不同调用方式（HTTP POST和使用封装好的包），下一篇中，我们将介绍如何构建一个表情搜索机器人。

知乎

首发于
乙醚的计算机讲堂

报错：No handlers could be found for logger "wxpy.api.bot"

有报错但是无法显示，可以选择在代码头部加入：

```
import logging
logging.basicConfig()
```

消息处理：删除@内容

如果不删除消息中@部分的内容，图灵机器人的回复可能会受到昵称内容的影响，导致回复不准确，或是识别不出一些应当识别出的内容。我们可以用一段简单的代码删除@到空格之间的内容并去除首尾多余的空格。

```
msg_content = re.sub('@[^\s]*', '', unicodedata.normalize('NFKC', msg.text)).strip()
```

这里使用了正则表达式，匹配@以及它之后所有不为空的字符。如果你的微信昵称中没有空白字符，这条代码是可行的。（需要在代码开头添加import re）

编辑于 2017-06-26

[Python](#) [聊天机器人](#) [微信](#)

文章被以下专栏收录

**乙醚的计算机讲堂**

写了那么多程序，却仍然过不好一生

[进入专栏](#)

推荐阅读

从零开始微信机器人（一）：
wxpy简介（登录 消息发送

[▲ 赞同 15 ▼](#)[● 11 条评论](#)[➤ 分享](#)[★ 收藏](#)[...](#)[从零开始微信机器](#)

知乎



首发于
乙醚的计算机讲堂



locoda/connector-wechat-bot

乙醚


发表于乙醚的计算...



11 条评论

切换为时间排序

写下你的评论...



- 

千夫指

1 年前

占领首赞，过几天闲了做一个

👍 赞
- 

风冷冷的吹

1 年前

key是啥？指点指点我

👍 赞
- 

Yan Zou

1 年前

膜拜大神

👍 赞
- 

知乎用户

1 年前

msg.member一直显示为None是什么情况？

👍 赞
- 

乙醚 (作者) 回复 知乎用户

1 年前

你是不是在群聊中？不在群聊中的没有这个member属性，在私聊中你可以选择使用msg.chat

👍 赞
- 

知乎用户 回复 乙醚 (作者)

1 年前

恩，已经好了，多谢啦~

👍 赞
- 

王涛

1 年前

api.ai已经改名字叫dialogflow了，而且网站翻墙了都打不开

知乎

首发于
乙醚的计算机讲堂

输入完tuling = Tuling(api_key='密钥')

之后, 报错说, Tuling没有被定义

请问这个怎么搞? 谢谢!

👍 赞



知乎用户

8 个月前

能更新到v2版本吗, 我的一直报4001错误

👍 赞



黄金猫

4 个月前

不知道你还会不会看到...

```
msg_content = re.sub('@[^\s]*', '', unicodedata.normalize('NFKC', msg.text)).strip
```



这一句, 我加在代码里, 发现回复都没了, 但是print在后台的内容可见。部分代码如下

@bot.register(Group, TEXT) # 这里注册了群聊中的文字消息, 测试时可以设置为自己(上篇中提到过)

```
def group_msg(msg):
```

```
    if '大猫' in msg.text:
```

```
        msg_content = re.sub('大猫', '', unicodedata.normalize('NFKC', msg.text)).strip()#去掉问话里的大猫
```

👍 赞



薛定谔的猫DoA

3 个月前

ha ?我以为wxpy会自己调用图灵。还要自己写啊、、

👍 赞

▲ 赞同 15 ▼

💬 11 条评论

➦ 分享

★ 收藏

...

从零开始微信机器