

全部课程 (/courses/) / 利用微信API将你的微信变为聊天机器人 (/courses/684) / 利用微信API将你的微信个人号变为聊天机器人

在线实验，请到PC端体验

一、课程介绍

1. 课程来源

本文为LittleCoder (<https://github.com/littlecodersh>)原创，使用微信个人号接口itchat (<https://github.com/littlecodersh/itchat>)。

关于itchat进一步使用的问题你可以在主页 (<https://github.com/littlecodersh/itchat>)加入官方交流群也可以在课程下面向我提问。

课程在常见的三种系统中都可以进行操作，itchat建议使用可以安装的最新版本。

本课程通过聊天机器人为例，介绍如何使用Python完成微信的点对点信息交互。

2. 内容简介

- 课程实现微信个人号聊天机器人
- 通过自定义消息处理方法加入聊天功能

3. 课程知识点

本课程项目完成过程中将学习：

- 微信消息的基本获取与处理
- 微信消息的指定发送

其中将重点介绍微信消息的获取与处理。

二、实验环境

在终端中输入以下命令，完成微信的API包itchat的安装。

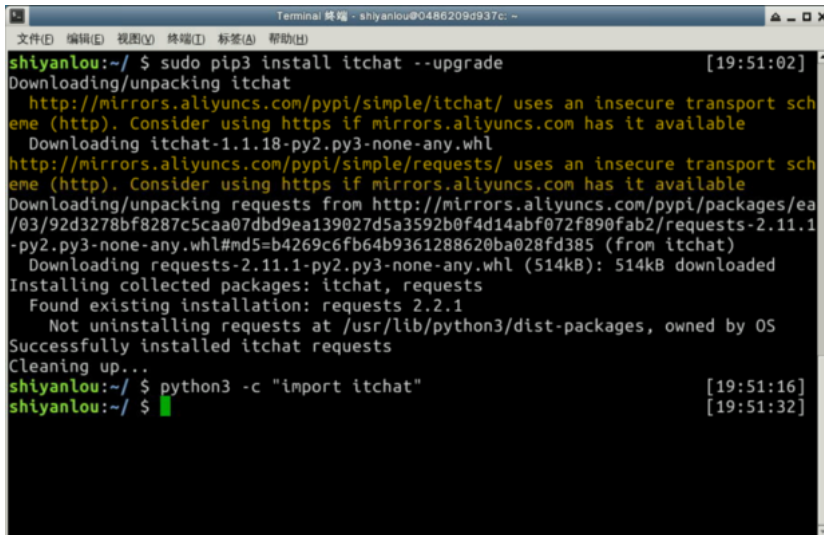
我们这里使用python3的环境（python2也是可行的）：

```
sudo pip3 install itchat --upgrade
```

通过该命令判断是否安装成功：

```
python3 -c "import itchat"
```

如果没有报错信息说明你已经将实验环境安装完成。



```
shiyanolou:~/ $ sudo pip3 install itchat --upgrade [19:51:02]
Downloading/unpacking itchat
  http://mirrors.aliyuncs.com/pypi/simple/itchat/ uses an insecure transport scheme (http). Consider using https if mirrors.aliyuncs.com has it available
  Downloading itchat-1.1.18-py2.py3-none-any.whl
  http://mirrors.aliyuncs.com/pypi/simple/requests/ uses an insecure transport scheme (http). Consider using https if mirrors.aliyuncs.com has it available
  Downloading/unpacking requests from http://mirrors.aliyuncs.com/pypi/packages/ea/03/92d3278bf8287c5caa07dbd9ea139027d5a3592b0f4d14abf072f890fab2/requests-2.11.1-py2.py3-none-any.whl#md5=b4269c6fb64b9361288620ba028fd385 (from itchat)
  Downloading requests-2.11.1-py2.py3-none-any.whl (514kB): 514kB downloaded
Installing collected packages: itchat, requests
  Found existing installation: requests 2.2.1
  Not uninstalling requests at /usr/lib/python3/dist-packages, owned by OS
Successfully installed itchat requests
Cleaning up...
shiyanolou:~/ $ python3 -c "import itchat" [19:51:16]
shiyanolou:~/ $ [19:51:32]
```

三、实验原理

通过微信的Python接口itchat获取微信消息。

将微信消息传输到机器人接口（这里以图灵为例），获取机器人的返回消息。

将返回消息返回给微信消息的发送人。

实现将微信个人号变为聊天机器人的目的。

四、实验步骤

0. 基础知识

为了照顾一些从未使用过Python的新用户与使用其他语言的用户，这里简单的讲一下以下的代码如何使用。

下面的每一段描述都给出了相应的测试代码，如果没有特殊说明这段代码可以这样使用：

打开桌面的Xfce终端，先将目录通过以下命令切到桌面。

```
cd Desktop
```

之后使用gedit编辑器编辑我们的主程序。

你也完全可以使用vim，会使用vim的话想必也知道这里应该输入什么命令了。

```
gedit test.py
```

最后将给出的代码复制进编辑器，保存并退出，使用如下命令就可以使用了。

```
python3 test.py
```

那么，就让我们开始正式进入Python操作微信的探索之旅吧。

1. 实现微信消息的获取

itchat的注册时根据类型注册的。

在获取相应类型的信息时会调用该函数。

我们现在只需要获取最简单的文本消息，那么只需要这样注册：

```
import itchat

@itchat.msg_register(itchat.content.TEXT)
def print_content(msg):
    print(msg['Text'])

itchat.auto_login()
itchat.run()
```

其中第三行即注册的操作，通过装饰符将 `print_content` 注册为处理文本消息的函数。

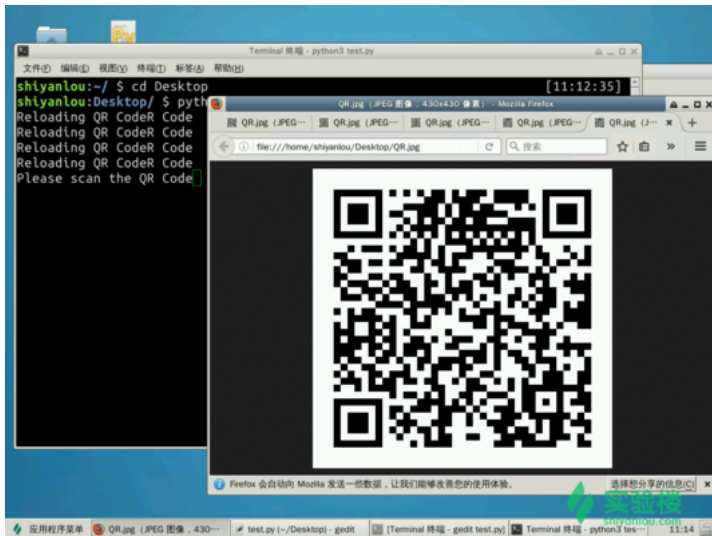
微信有各种类型的数据，例如图片、语音、名片、分享等，也对应不同的注册参数：

- 图片对应 `itchat.content.PICTURE`
- 语音对应 `itchat.content.RECORDING`
- 名片对应 `itchat.content.CARD`
- 其余的这里就不一一列举，更具体的内容可以自行搜索itchat阅读文档 (<http://itchat.readthedocs.io/zh/latest/>)

执行命令

```
python3 test.py
```

就可看到我们开始登陆微信：



扫码完成以后最基础的文本信息的接收就完成了，你可以尝试用他人的微信给自己发一条信息。

如果你不想要每次运行程序都扫码，可以在登陆命令中进行设置：

```
itchat.auto_login(hotReload=True)
```

2. 实现微信消息的发送

微信可以发送各类消息，文本、图片、文件等，不过我们现在只需要使用文本的发送。

其余的消息的发送有兴趣可以自行阅读。

```
itchat.send('Message Content', 'toUserName')
```

该发送消息的函数需要两个参数，消息的内容与接受者的 `UserName`，即标识符。

那么我们试着向文件传输助手发送一条消息：

```
#coding=utf8
import itchat

itchat.auto_login(hotReload=True)

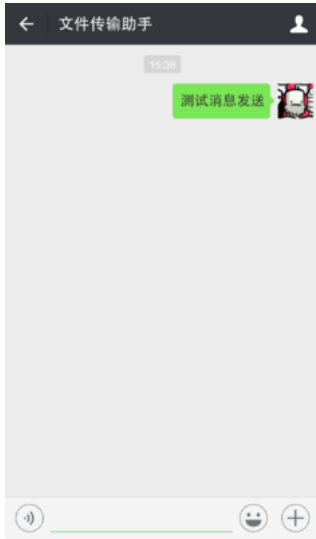
# 注意实验楼环境的中文输入切换
itchat.send(u'测试消息发送', 'filehelper')
```

打开手机看一下是否就完成了消息的发送。

保存代码后，执行命令：

```
python3 test.py
```

扫描登录后的效果如下：



当然，还有一种更加快捷的回复方法就是在注册函数中直接回复。

例如下面的例子将会将文本消息原封不动的返回。

```
import itchat

@itchat.msg_register(itchat.content.TEXT)
def print_content(msg):
    return msg['Text']

itchat.auto_login()
itchat.run()
```

这种方式显然更加直观也更加简单（不需要输入接受者的UserName）

我们本次实践将会采用这种方式。

3. 实现最简单的与图灵机器人的交互

要做一个能够与人交流的机器人有很多种方法，最简单的莫过于使用他人提供的接口。

我们这里以图灵机器人为例，演示这一功能。

图灵机器人简单而言就是以一定的规则给图灵的服务器发送数据包（包含你对他说的话）

图灵的服务器会以一定的规则给你返回数据包（包含他回复你的话）

你需要一个Tuling Key来告诉图灵服务器你有权和他对话，我这里免费提供一些：

```
8edce3ce905a4c1dbb965e6b35c3834d
eb720a8970964f3f855d863d24406576
1107d5601866433dba9599fac1bc0083
71f28bf79c820df10d39b4074345ef8c
```

下面我做一个配置图灵机器人的简单介绍，你想要自行了解或者申请Tuling Key可以看这里 (http://tuling123.com/help/h_cent_webapi.jhtml)

发送的规则简而言之是这样的：

```
{
  'key'      : 'TULING_KEY',
  'info'     : 'YOUR_MSG',
  'userid'   : 'USERID',
}
```

其中userId是用户的标志，让机器人知道你是你。（也就是一个Tuling Key可以有多个用户）

而返回的内容基本是这样的：

```
{
  'code': 0,
  'text': 'RETURN_MSG',
}
```

我们需要的内容就在text键里面。

这里我们使用requests包完成整个操作（已经包含在itchat包的安装中了）。

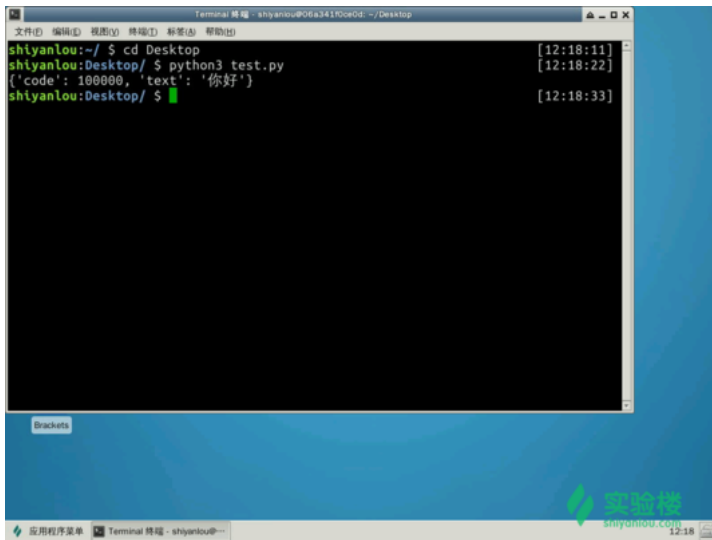
最后值得一提的就是这是一个post请求，那么直接上代码应该比我絮絮叨叨的说要直观很多。

```
#coding=utf8
import requests

apiUrl = 'http://www.tuling123.com/openapi/api'
data = {
  'key' : '8edce3ce905a4c1dbb965e6b35c3834d', # 如果这个Tuling Key不能用，那就换一个
  'info' : 'hello', # 这是我们发出去的消息
  'userid' : 'wechat-robot', # 这里你想改什么都可以
}
# 我们通过如下命令发送一个post请求
r = requests.post(apiUrl, data=data).json()

# 让我们打印一下返回的值，看一下我们拿到了什么
print(r)
```

我们可以看到他回复了你好。



```
shiyanolou:~/ $ cd Desktop
shiyanolou:Desktop/ $ python3 test.py
{'code': 100000, 'text': '你好'}
```

至此我们已经理解并掌握了所有需要的内容，下面将其组装起来即可。

五、实验程序

我先从概念上说一下组装是一个怎么样的过程。

当然，如果你觉得代码更直观，我也在代码中为你写好了注释。

这里我们首先将与图灵服务器的交互定义为一个函数。

我们需要这个函数接收我们要发送给图灵的消息，返回图灵返回给我们的消息。

再将与图灵交互并返回图灵返回结果的操作写成函数并在itchat中注册。

最后启动itchat，我们的程序就完成了。

```

#coding=utf8
import requests
import itchat

KEY = '8edce3ce905a4c1dbb965e6b35c3834d'

def get_response(msg):
    # 这里我们就像在“3. 实现最简单的与图灵机器人的交互”中做的一样
    # 构造了要发送给服务器的数据
    apiUrl = 'http://www.tuling123.com/openapi/api'
    data = {
        'key' : KEY,
        'info' : msg,
        'userid' : 'wechat-robot',
    }
    try:
        r = requests.post(apiUrl, data=data).json()
        # 字典的get方法在字典没有'text'值的时候会返回None而不会抛出异常
        return r.get('text')
    # 为了防止服务器没有正常响应导致程序异常退出，这里用try-except捕获了异常
    # 如果服务器没能正常交互（返回非json或无法连接），那么就会进入下面的return
    except:
        # 将会返回一个None
        return

# 这里是在“1. 实现微信消息的获取”中已经用过的同样的注册方法
@itchat.msg_register(itchat.content.TEXT)
def tuling_reply(msg):
    # 为了保证在图灵Key出现问题的时候仍旧可以回复，这里设置一个默认回复
    defaultReply = 'I received: ' + msg['Text']
    # 如果图灵Key出现问题，那么reply将会是None
    reply = get_response(msg['Text'])
    # a or b的意思是，如果a有内容，那么返回a，否则返回b
    # 有内容一般就是指非空或者非None，你可以用`if a: print('True')`来测试
    return reply or defaultReply

# 为了让实验过程更加方便（修改程序不用多次扫码），我们使用热启动
itchat.auto_login(hotReload=True)
itchat.run()

```

六、实验结果

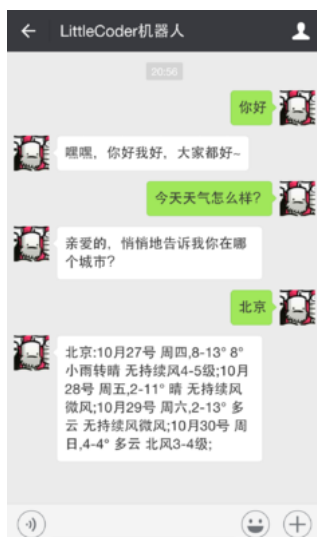
在本机上通过如下命令可以运行该程序

```
python3 main.py
```

扫码登陆后程序就成功运行了。

之后在手机上使用别的账号给自己的微信号发送消息即可获得机器人的回复。

这里给出使用的效果图：



如果你想要通过与其他用户的交互完成该操作，自行在注册的函数中进行修改即可。

如果你想要进一步了解使用Python控制微信的细节，你也可以去到项目主页itchat (<https://github.com/littlecodersh/itchat>)。

或者直接阅读文档 (<http://itchat.readthedocs.io/zh/latest/>)也是不错的选择。

如果你的本地环境并非Python3也没有关系，itchat同样完美支持Python2。

七、代码获取

动手实践是学习 IT 技术最有效的方式!
我将整个项目目录做了一个打包，你可以直接下载后运行。

[开始实验](#)

```
wget https://labfile.oss.aliyuncs.com/courses/684/main.py
```

如果有什么问题，欢迎在我的主页 (<https://github.com/littlecodersh>)留言或者邮件 (<mailto:i7meavnktqegm1b@qq.com>)联系我。

课程教师



littlecodersh
共发布过1门课程

[查看老师的所有课程 > \(/teacher/290787\)](/teacher/290787)



动手做实验，轻松学IT



<http://weibo.com/shiyanlou2013>



合作

[我要投稿 \(/contribute\)](#)

[教师合作 \(/labs\)](#)

[高校合作 \(/edu/\)](#)

[友情链接 \(/friends\)](#)

[开发者 \(/developer\)](#)

学习路径

[Python学习路径 \(/paths/python\)](#)

[Linux学习路径 \(/paths/linuxdev\)](#)

[大数据学习路径 \(/paths/bigdata\)](#)

[Java学习路径 \(/paths/java\)](#)

[PHP学习路径 \(/paths/php\)](#)

[全部 \(/paths/\)](#)

公司

[关于我们 \(/aboutus\)](#)

[联系我们 \(/contact\)](#)

[加入我们 \(http://www.simplecloud.cn/jobs.html\)](http://www.simplecloud.cn/jobs.html)

[技术博客 \(https://blog.shiyanlou.com\)](https://blog.shiyanlou.com)

服务

[企业版 \(/saas\)](#)

[实战训练营 \(/bootcamp/\)](#)

[会员服务 \(/vip\)](#)

[实验报告 \(/courses/reports\)](#)

[常见问题 \(/questions/?](#)

[tag=%E5%B8%B8%E8%A7%81%E9%97%AE%E9%A2%98\)](#)

[隐私条款 \(/privacy\)](#)