

一、机器人设置

设置流程: [如何设置群机器人 -帮助中心-企业微信 \(qq.com\)](#)

记得将Webhook复制保存下来

二、机器人定时推送功能代码实现

推送消息模块代码

```
clock.py
```

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-
import requests
import json
import pandas as pd
import os
# 企业微信机器人 Webhook
webhook_url = 'https://qyapi.weixin.qq.com/cgi-bin/webhook/send?key=你的机器人KEY'

# 发送文本消息
def send_text_message(content):
    headers = {'Content-Type': 'application/json'}
    data = {
        "msgtype": "text",
        "text": {
            "content": content
        }
    }
    response = requests.post(webhook_url, headers=headers, data=json.dumps(data))
    return response.json()

# 发送文件
def send_file_message(file_path):
    # 上传文件接口, 需要先把文件上传, 获取media_id
    upload_url = f'https://qyapi.weixin.qq.com/cgi-bin/webhook/upload_media?key=你的机器人KEY&type=file'
    with open(file_path, 'rb') as f:
        files = {'file': f}
        response = requests.post(upload_url, files=files)
        result = response.json()
        if result.get('media_id'):
            # 使用media_id发送文件消息
            media_id = result['media_id']
            data = {
                "msgtype": "file",
                "file": {
                    "media_id": media_id
                }
            }
            headers = {'Content-Type': 'application/json'}
            send_response = requests.post(webhook_url, headers=headers,
data=json.dumps(data))
            return send_response.json()
        else:
            print("文件上传失败:", result)
            return result
```

```

# 读取并处理 Excel 文件（假设文件名为 '白云指标.xlsx'）
def process_excel_file(file_path):
    df = pd.read_excel(file_path)
    # 你可以在这里进行数据处理，例如读取特定数据
    summary = df.describe() # 简单获取文件的统计信息
    return summary

def main():
    file_path = '白云指标.xlsx'
    if os.path.exists(file_path):
        # 处理 Excel 文件
        summary = process_excel_file(file_path)
        # 发送 Excel 文件
        send_file_message(file_path)
        # 发送文本消息，附上文件的摘要信息
        send_text_message(f"已发送文件: {file_path}\n文件摘要:\n{summary}")
    else:
        send_text_message(f"文件 {file_path} 不存在")

if __name__ == "__main__":
    main()

```

```

#!/usr/bin/python3 --python包所在路径，这里使用应用镜像是CentOs8.x，环境自带python3
--如果是python以下版本，请改成以下
#!/usr/bin/python

```

定时设置代码

使用crontab表达式实现

```

* * * * * python3 /文件路径
#* * * * * crontab表达式
#如果你的环境是python3以下版本，请把python3改成python
#/文件路径 需要执行的py文件路径
#示例：
0 16 * * * python3 /root/clock.py
#上述代码表示在每天下午四点钟运行执行/root目录下的clock.py文件以此实现定时推送功能

```

crontab表达式说明

```

* * * * *
- - - - -
| | | | |
| | | | | +----- 星期中星期几 (0 - 6) (星期天为0)
| | | | | +----- 月份 (1 - 12)
| | | | | +----- 一个月中的第几天 (1 - 31)
| | | | | +----- 小时 (0 - 23)
+----- 分钟 (0 - 59)

```

crontab验证工具

在线常用crontab表达式大全验证解析 - ToolTT [在线工具箱](#)

命令行

```

vi clock.py
进入文件编辑界面，点击键盘i键进行文件内容写入，写入完毕后点击esc键退出，然后键盘输入 :wq保存退出
./clock.py #运行测试看是否能运行clock.py
然后执行
crontab -e

```

进入crontab定时编辑界面，写入表达式 例如 `0 16 * * * python3 /root/clock.py` 然后按esc键退出编辑，
键盘输入 `:wq` 保存退出

你可将此代码文件和命令行在本地或者云服务器上执行。

Dengke Chen