# 第二十章 Zabbix二次开发

本节所讲内容：

20.1 基于Zabbix的Python报警

20.2 基于Python的Zabbix开发

## 20.1 基于Zabbix的Python报警

Server 192.168.1.89

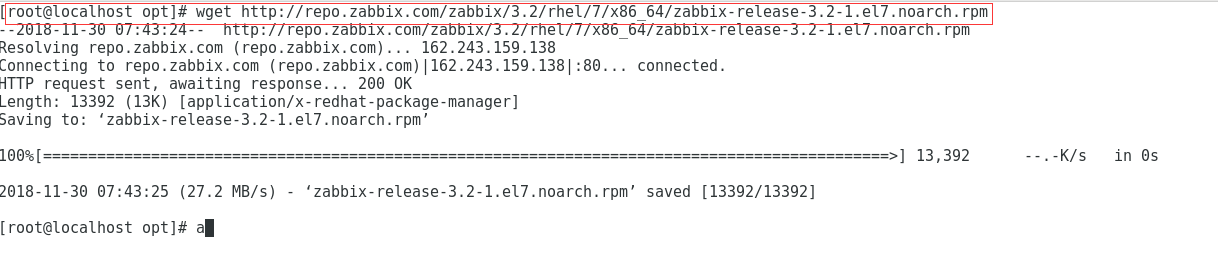
Agent 192.168.1.90

20.1.1、添加一个zabbix的客户端

1、安装

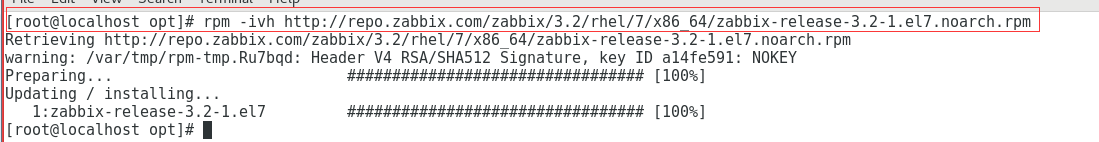
下载rpm包

wget <http://repo.zabbix.com/zabbix/3.2/rhel/7/x86_64/zabbix-release-3.2-1.el7.noarch.rpm>



进行rpm包安装

rpm -ivh http://repo.zabbix.com/zabbix/3.2/rhel/7/x86\_64/zabbix-release-3.2-1.el7.noarch.rpm



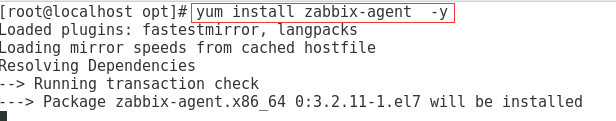
缓存yum信息

yum makecache



发起安装，客户端

yum install zabbix-agent -y



20.1.2配置

[root@xuegod63 ~]# cd /etc/zabbix/

[root@xuegod63 zabbix]# ls

web zabbix\_agentd.conf zabbix\_agentd.d zabbix\_server.conf

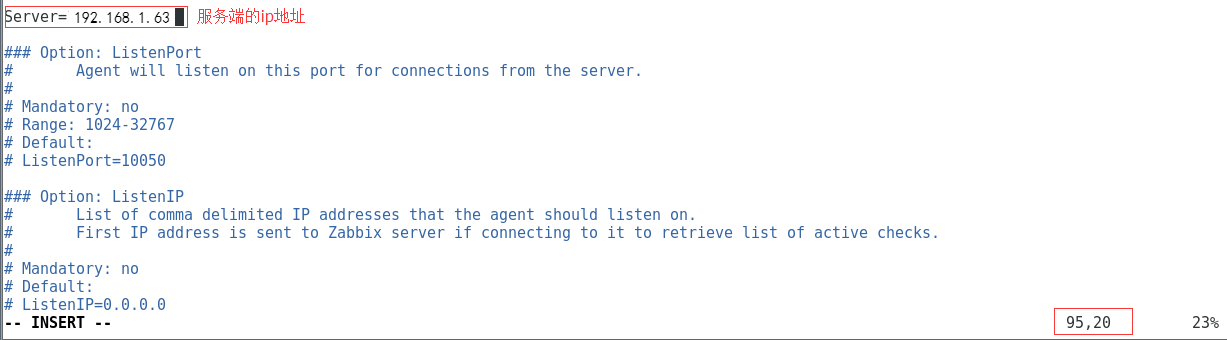
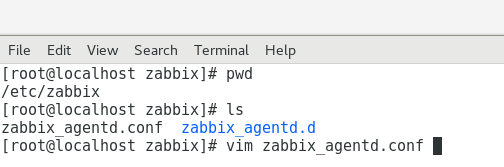
[root@xuegod63 zabbix]# vim zabbix\_agentd.conf

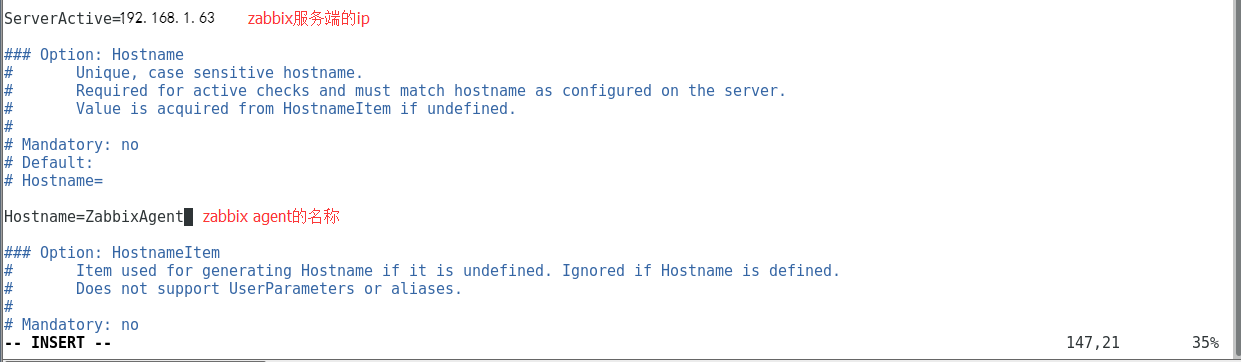
紧接着修改：

95 行 Server=192.168.1.89

136 行ServerActive=192.168.1.89

147行 Hostname=ZabbixAgent



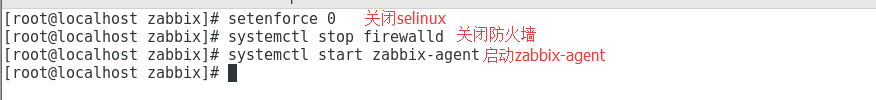


#配置setlinux防火墙配置

[root@localhost zabbix]# setenforce 0

[root@localhost zabbix]# systemctl stop firewalld

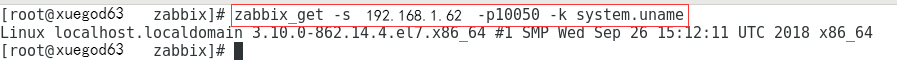
[root@localhost zabbix]# systemctl start zabbix-agent



3、确认

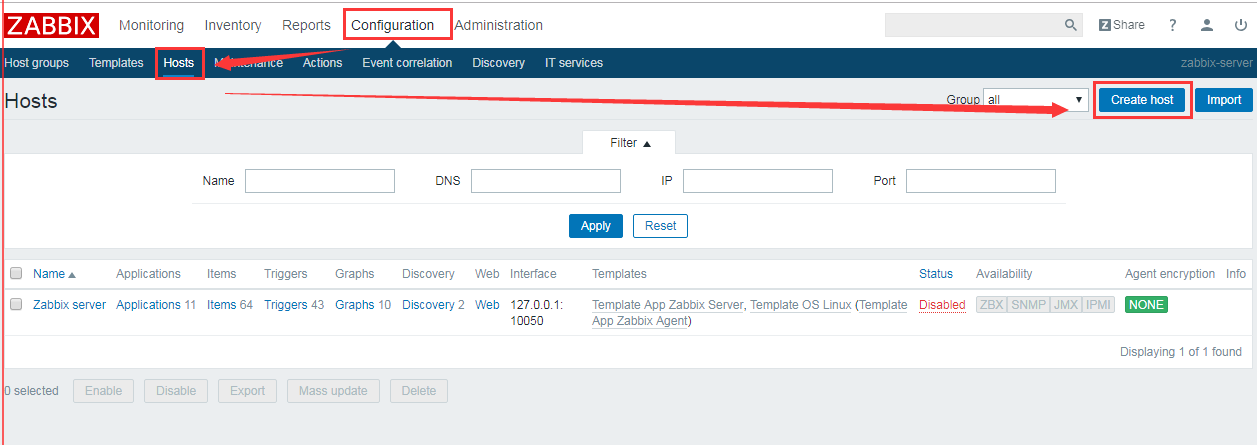
在服务端，在192.168.1.63上尝试连接agent

zabbix\_get -s 192.168.1.62 -p10050 -k system.uname

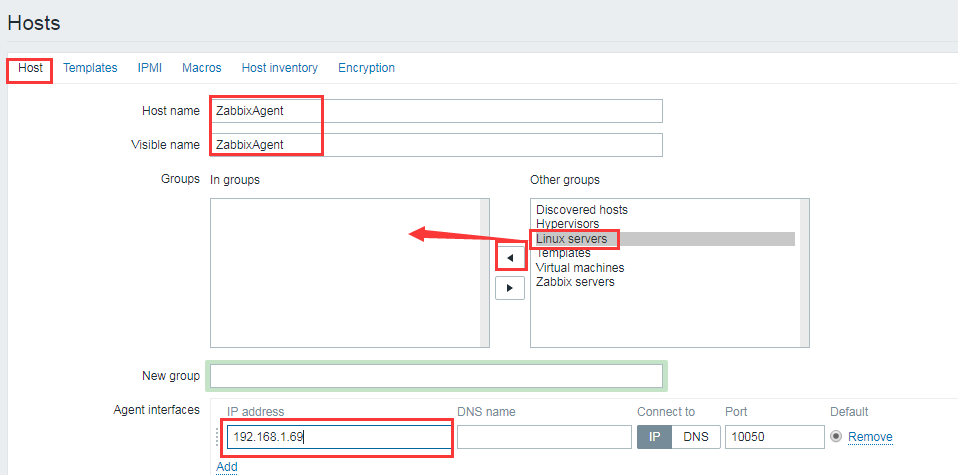


4、zabbix web添加主机

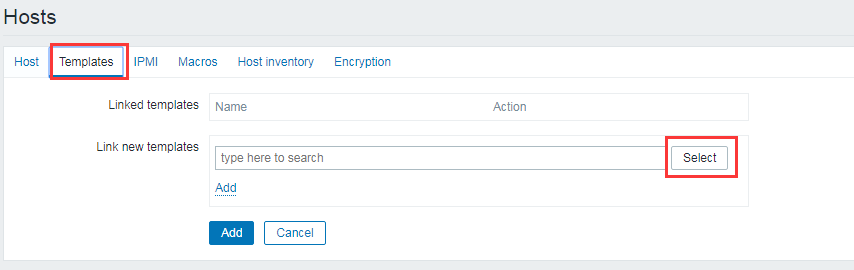
设置 🡪 主机 🡪 创建主机

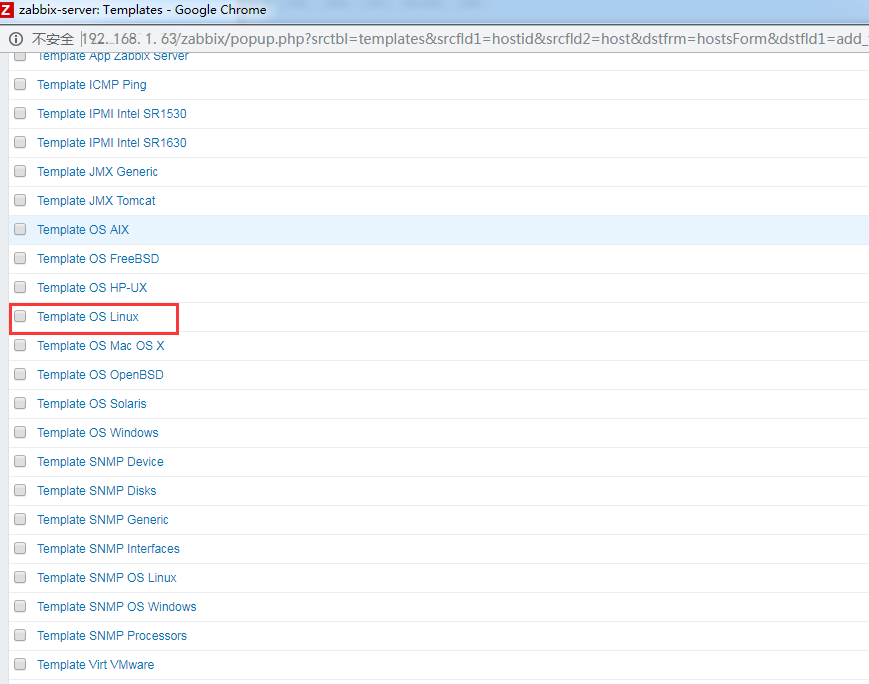


设置host名称、显示名称、组、ip

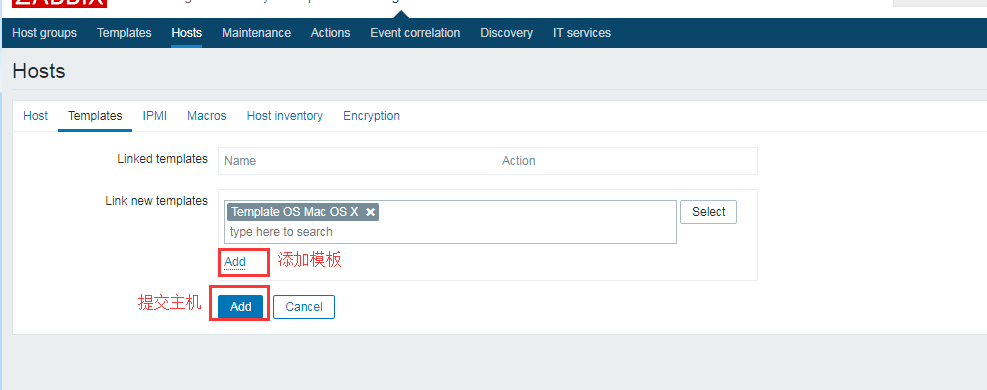


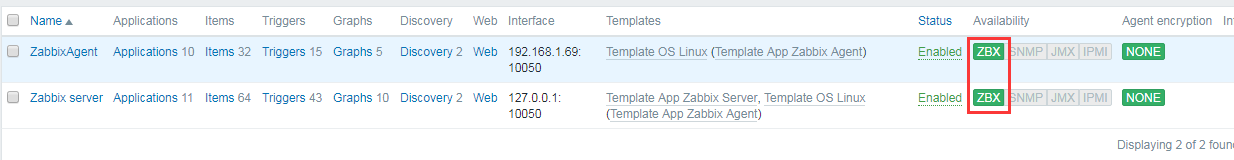
设置linux模板





提交





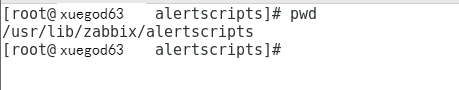
20.1.3进行zabbix报警

进行报警，逻辑大致相同，不同的是报警脚本的编写

1. 设置报警的脚本的目录

首先在服务端切换到报警脚本要存放的目录（这个目录是让zabbix寻找脚本）

[root@xuegod63 alertscripts]# cd /usr/lib/zabbix/alertscripts/

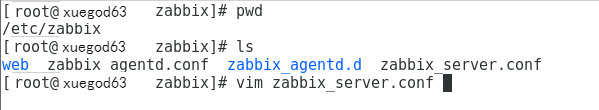


然后修改server服务端配置文件

[root@xuegod63 alertscripts]# cd /etc/zabbix/

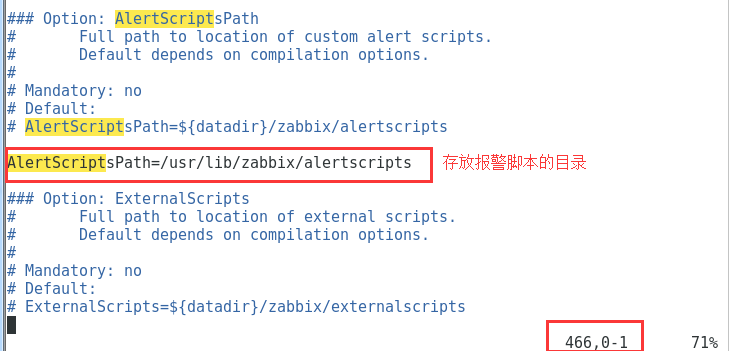
[root@xuegod63 zabbix]# ls

[root@xuegod63 zabbix]# vim zabbix\_server.conf



修改配置

455行 AlertScriptsPath=/usr/lib/zabbix/alertscripts



编写报警脚本

注意：zabbix默认调用的是你linux中的python2版本，我们需要用python2来执行脚本并成功运行。（！！！！非常重要）

1.编写Python sendEmail.py

[root@xuegod63 alertscripts]# vim sendEmail.py

#!/usr/bin/python

#coding:utf-8

import sys

import smtplib

from email.mime.text import MIMEText

mail\_host = "smtp.qq.com"

mail\_user = "3403073998@qq.com"

mail\_password = 'ihenvpgtjinqchbd'

def send\_mail(to,subject,content):

message = MIMEText(content,"plain","utf-8")

message["Subject"] = subject

message["From"] = mail\_user

message["to"] = to

try:

smtp = smtplib.SMTP(mail\_host,25)

smtp.login(mail\_user,mail\_password)

smtp.sendmail(mail\_user,[to],message.as\_string())

smtp.close()

except Exception as e:

with open("mail.log","a") as f:

f.write(str(e)+"\n")

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

send\_mail(

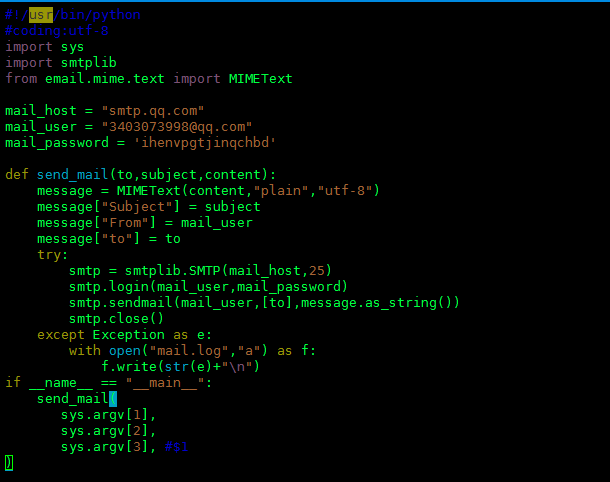
sys.argv[1],

sys.argv[2],

sys.argv[3], #$1

)

注意：在这个地方我们使用的是25端口



运行脚本测试效果如下

测试命令：

[root@xuegod63 alertscripts]# python37 sendEmail.py "1286211699@qq.com" "nice work" "hello I am for"



Python 钉钉报警

#!/usr/bin/python

#coding:utf-8

import sys

import json

import requests

def WriteLogByDing(content):

headers = {

"Content-Type": "application/json",

"Chartset": "utf-8"

}

request\_data = {

"msgtype": "text",

"text": {

"content": content

},

"at": {

"atMobiles": [],

"isAtAll": True

}

}

sendData = json.dumps(request\_data)

response = requests.post(url = url,headers = headers,data = sendData)

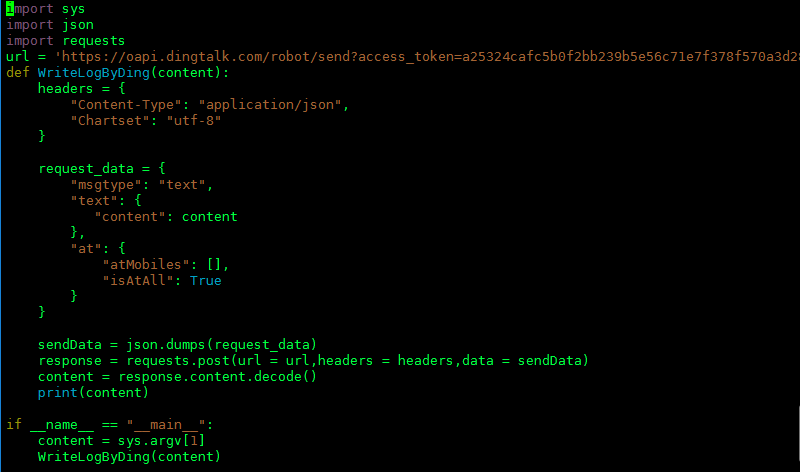
content = response.content.decode()

#print(content)

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

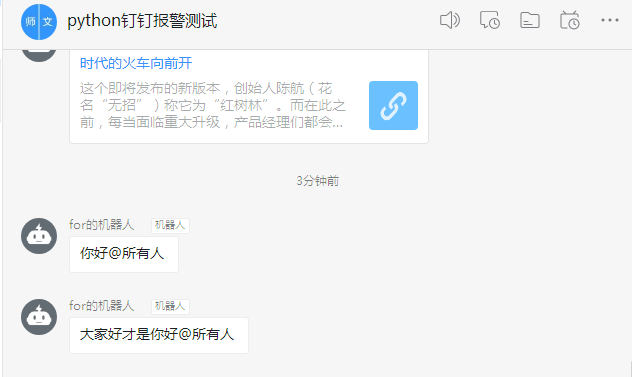
content = sys.argv[1]

WriteLogByDing(content)



运行脚本测试效果如下

测试命令：python sendDing.py "你好"



3、授权脚本（重要）

[root@xuegod63 alertscripts]# chown zabbix.zabbix sendEmail.py

[root@xuegod63 alertscripts]# chown zabbix.zabbix sendDing.py

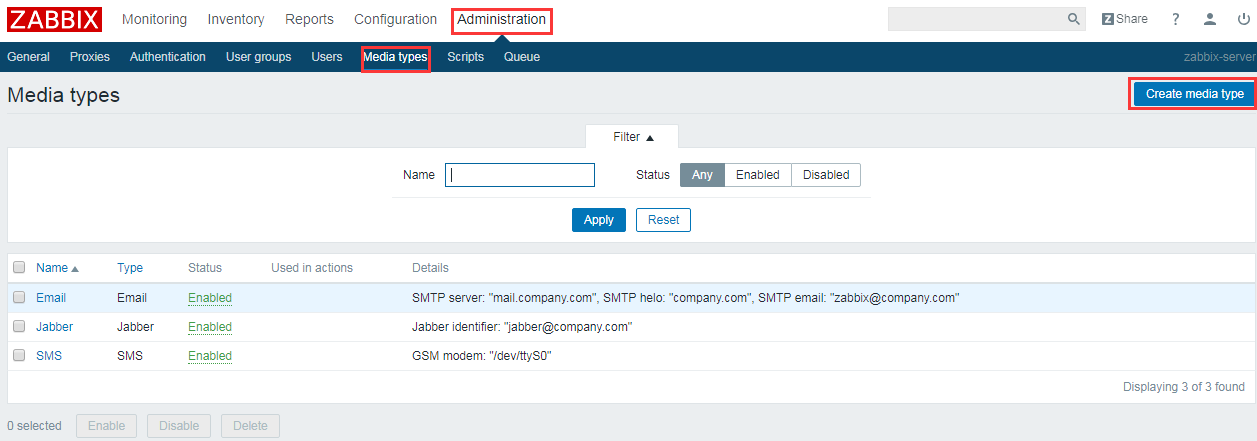
[root@xuegod63 alertscripts]# chmod +x sendDing.py

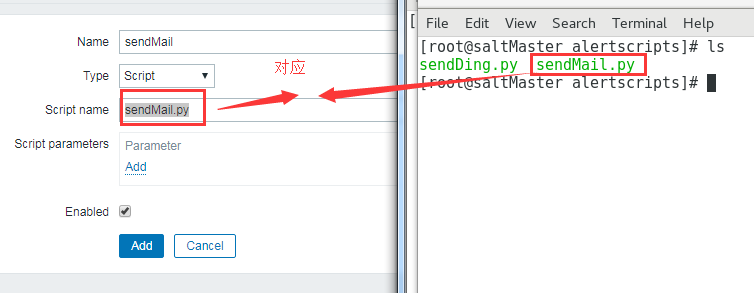
[root@xuegod63 alertscripts]# chmod +x sendEmail.py

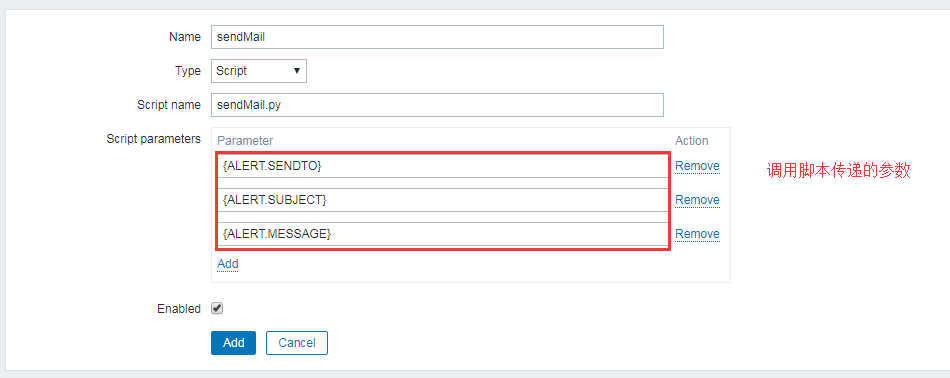
4、设置报警媒介、用户、触发器

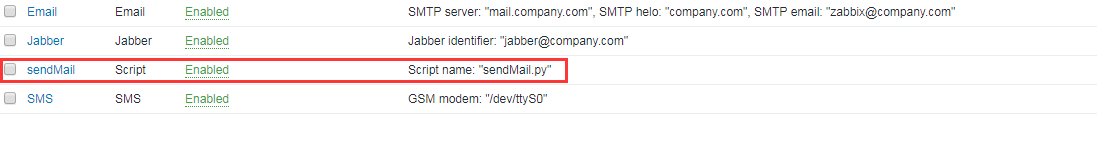
设置报警媒介

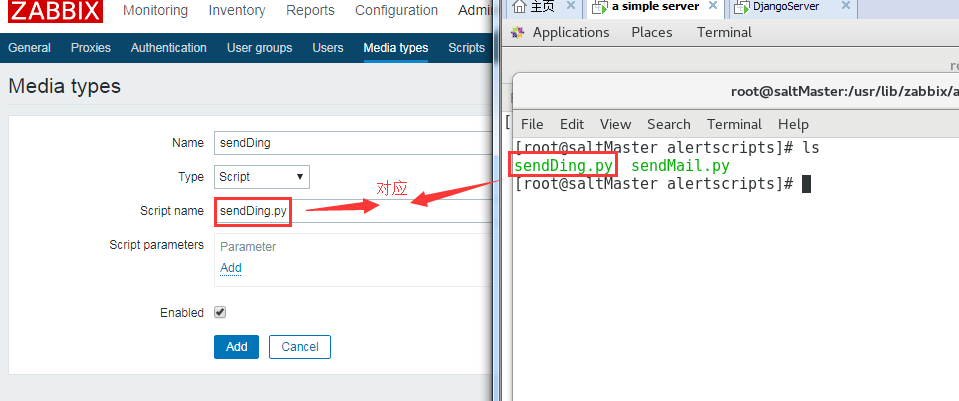
发送邮件

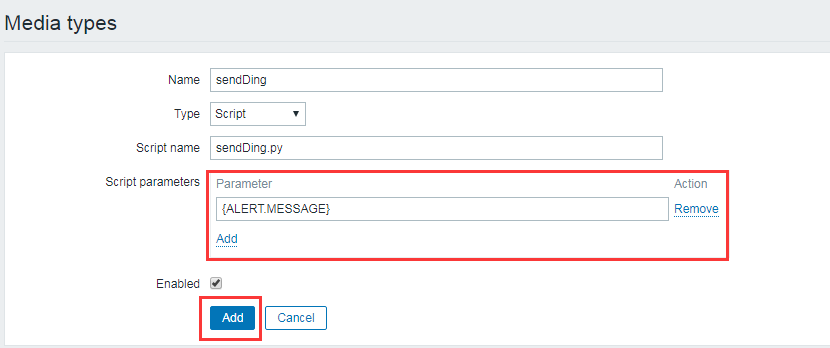




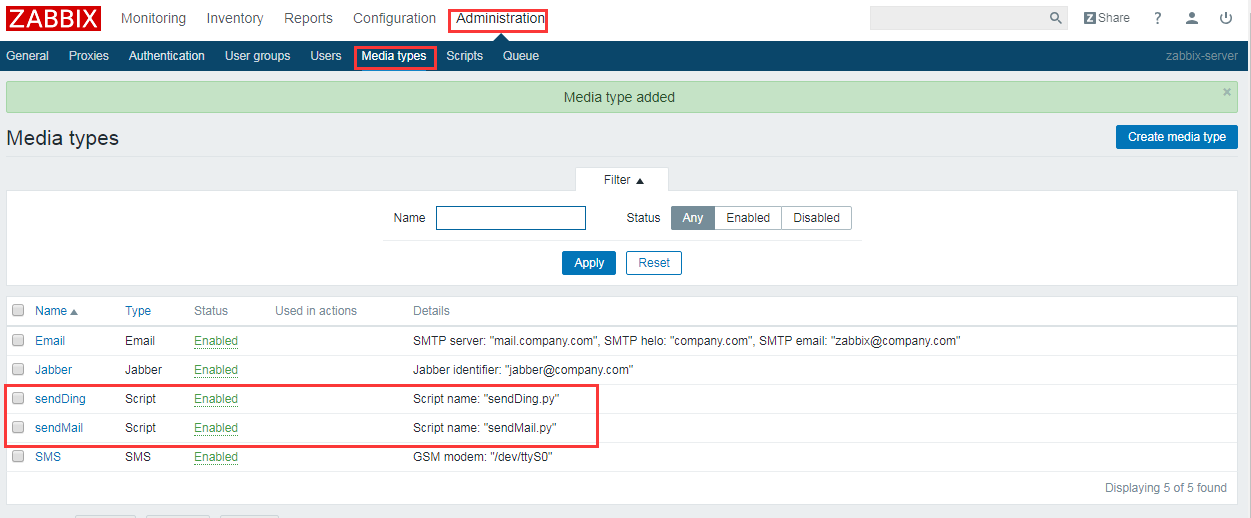


发送钉钉，步骤大同小异

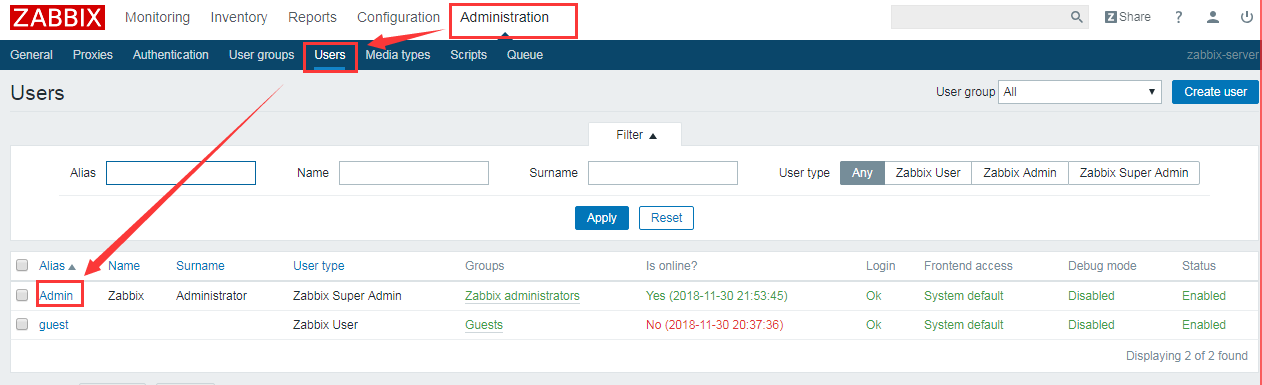




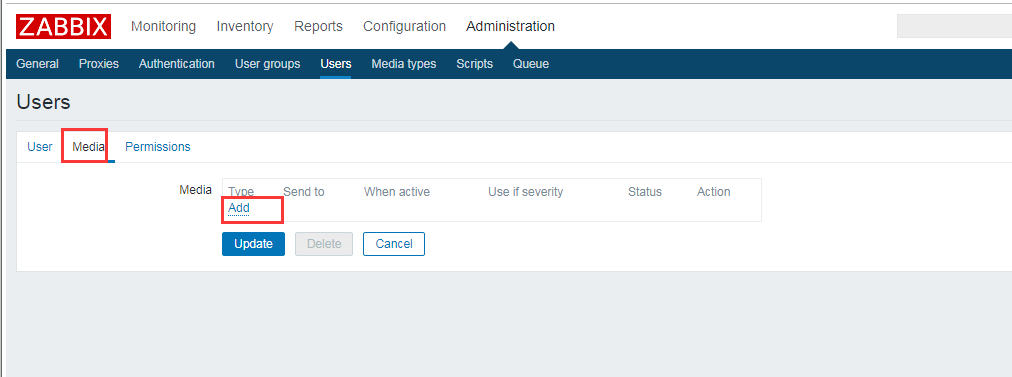
效果如下



设置报警用户

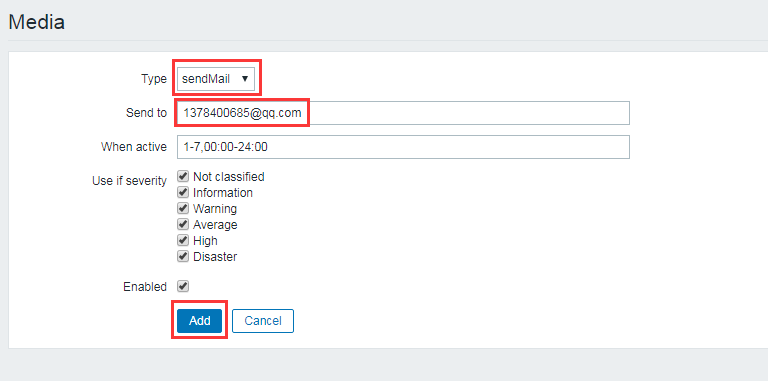


添加报警媒介

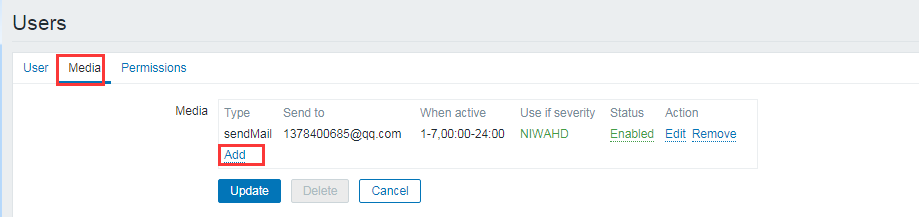


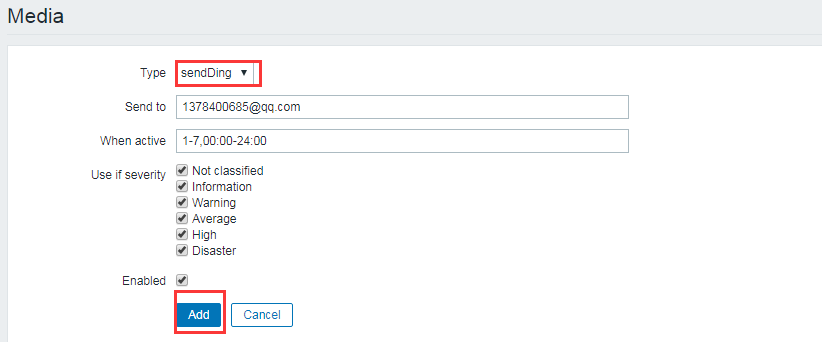
添加报警媒介

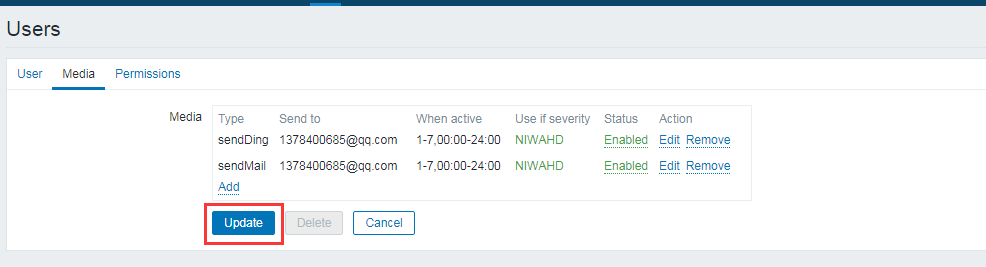
邮件



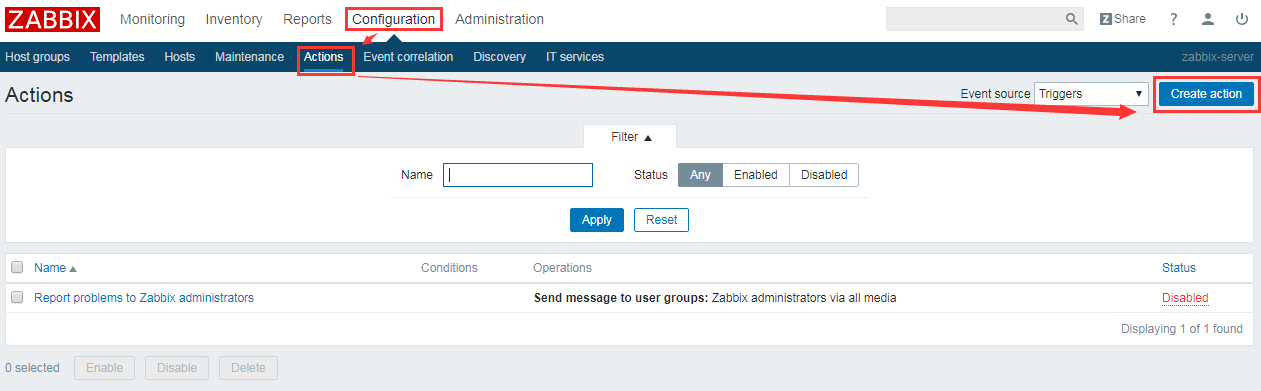
钉钉



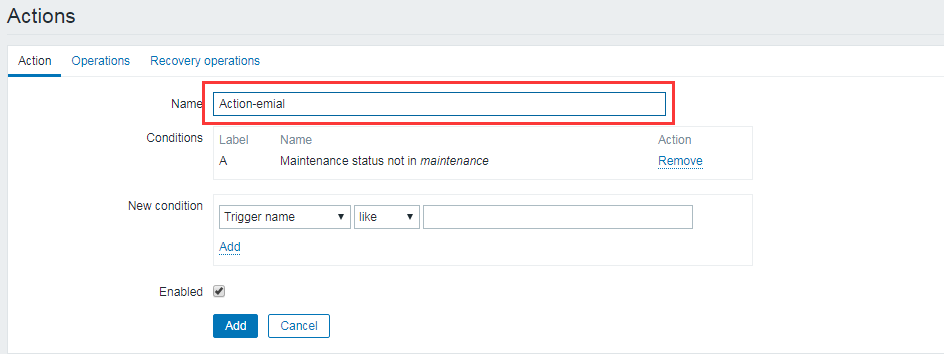




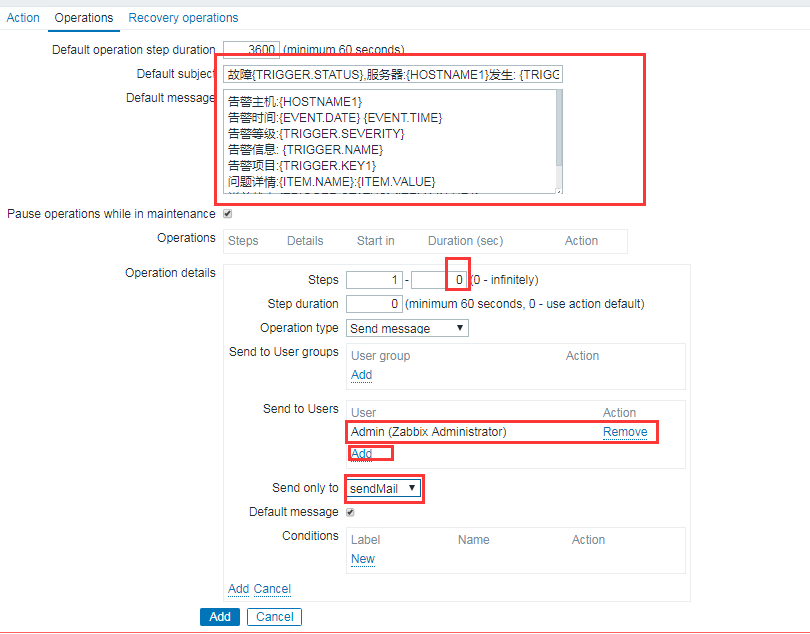
设置动作

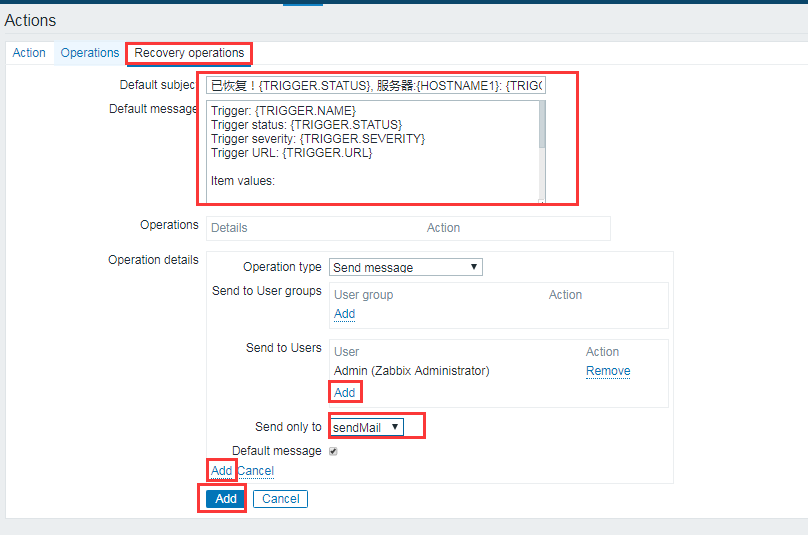


邮件

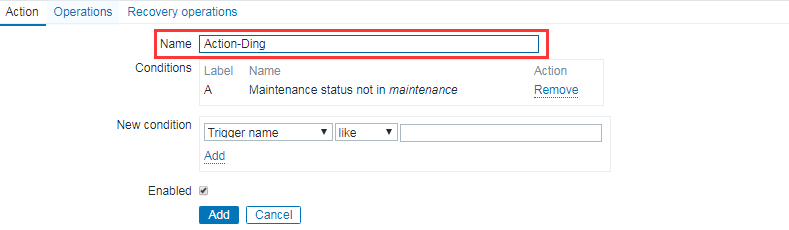


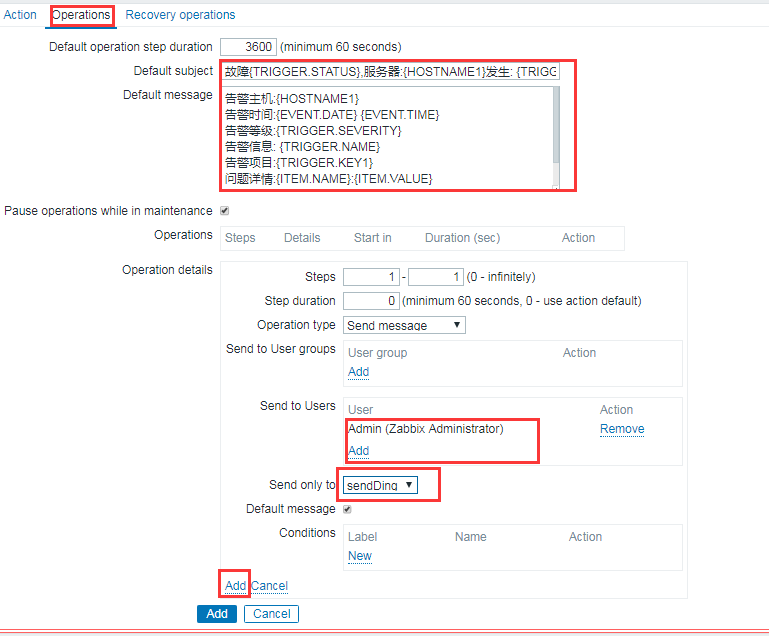
添加options信息

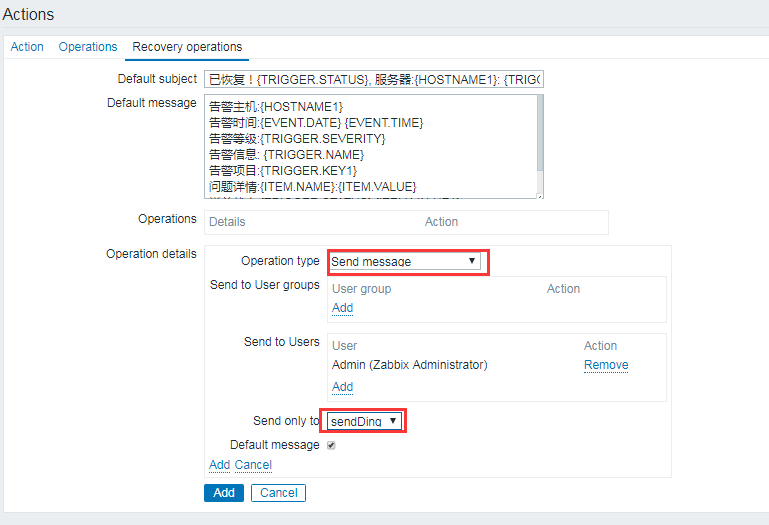


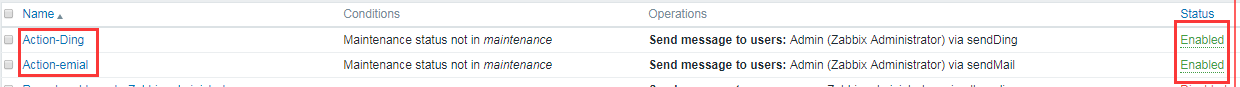


钉钉









默认接收人：故障{TRIGGER.STATUS},服务器:{HOSTNAME1}发生: {TRIGGER.NAME}故障!

默认信息：

告警主机:{HOSTNAME1}

告警时间:{EVENT.DATE} {EVENT.TIME}

告警等级:{TRIGGER.SEVERITY}

告警信息: {TRIGGER.NAME}

告警项目:{TRIGGER.KEY1}

问题详情:{ITEM.NAME}:{ITEM.VALUE}

当前状态:{TRIGGER.STATUS}:{ITEM.VALUE1}

事件 ID:{EVENT.ID}

恢复操作

默认接收人：已恢复！{TRIGGER.STATUS}, 服务器:{HOSTNAME1}: {TRIGGER.NAME}

恢复信息：

告警主机:{HOSTNAME1}

告警时间:{EVENT.DATE} {EVENT.TIME}

告警等级:{TRIGGER.SEVERITY}

告警信息: {TRIGGER.NAME}

告警项目:{TRIGGER.KEY1}

问题详情:{ITEM.NAME}:{ITEM.VALUE}

当前状态:{TRIGGER.STATUS}:{ITEM.VALUE1}

事件 ID:{EVENT.ID}

测试报警

关闭zabbix-agent查看是否报警

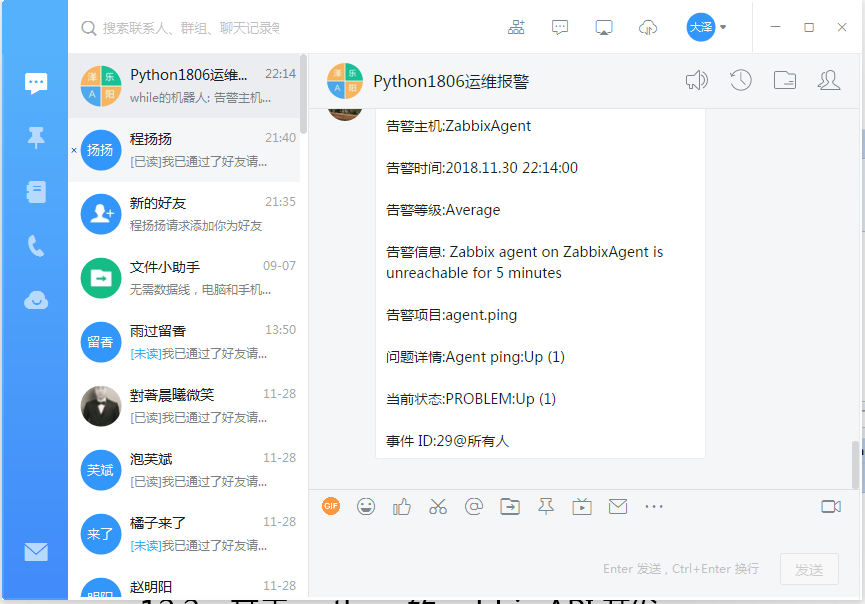


效果如下

mail



Ding



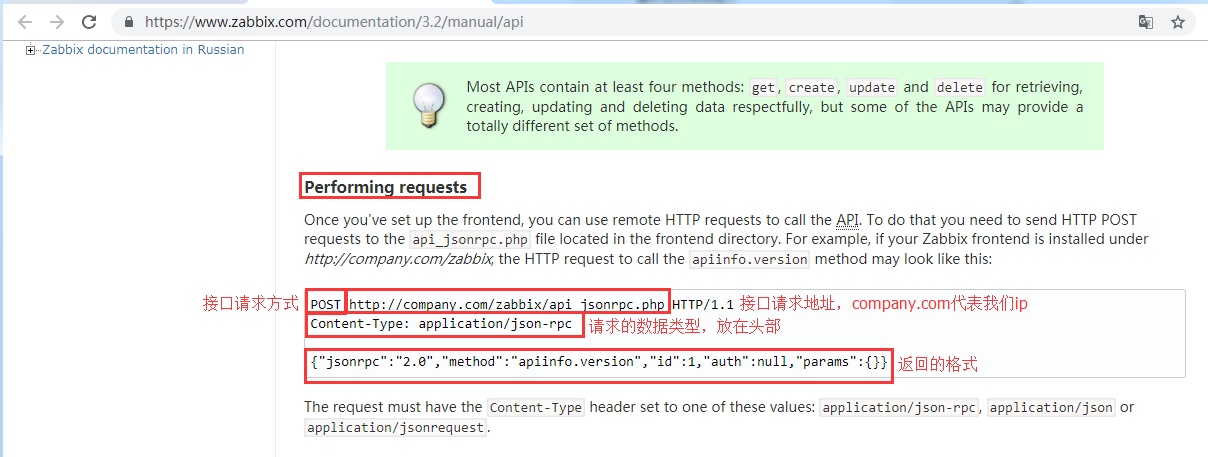
## 20.2 基于Python的Zabbix API开发

Zabbix给我们提供了稳定的api接口，方便大家二次开发，使用脚本对zabbix进行批量控制，

Zabbix给我们提供了健全的api文档。

文档地址在此：<https://www.zabbix.com/documentation/3.2/manual/api>

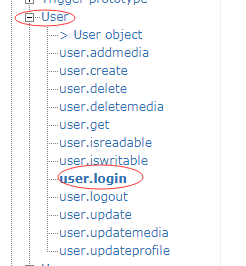
首先我们在文档首页，发现了zabbix接口开发的初步约定



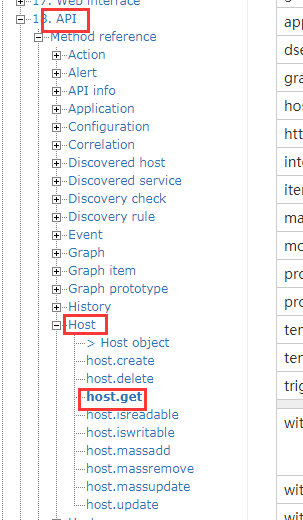
然后查看细节，我们以login和host来作为案例

接口命名禁止有动词

Zabbix user.login

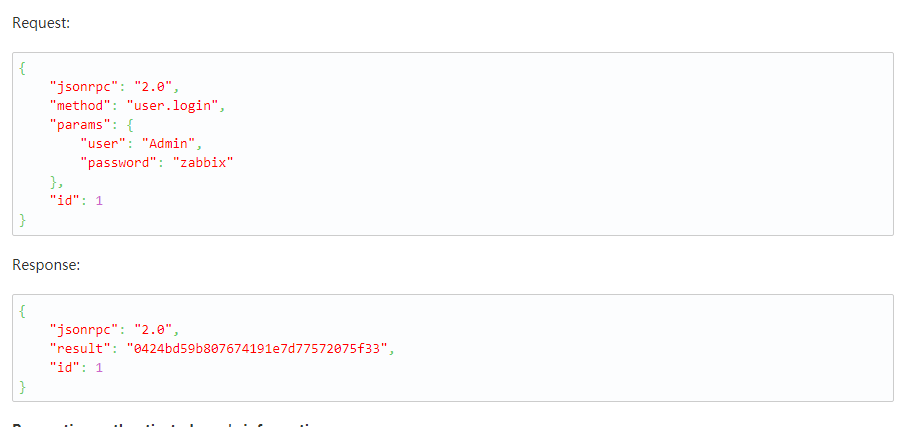


Zabbix host.get

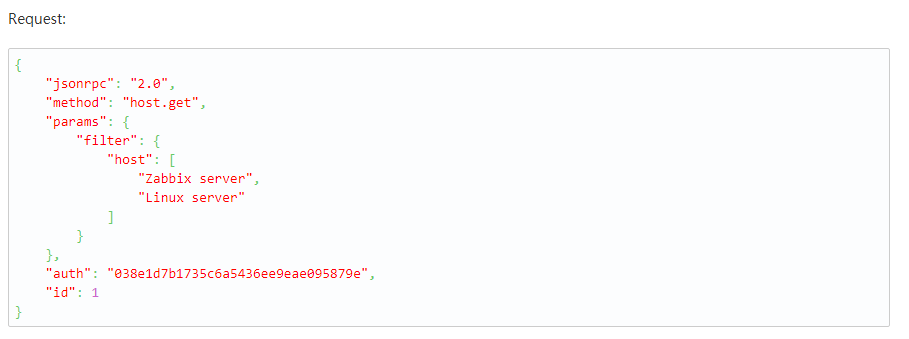


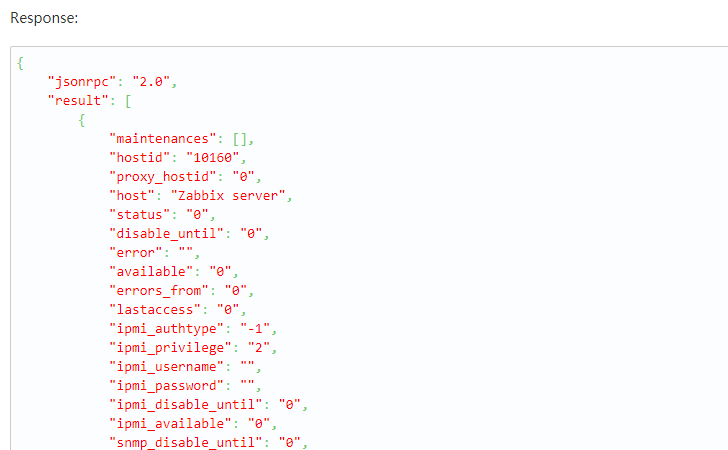
然后我们通过文档来查看请求需要的数据和返回的数据

User.Login



host.get

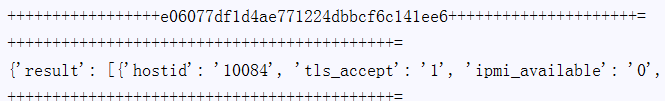




进行api接口请求

***#coding:utf-8*import json  
import requests  
  
api\_url = "http://192.168.1.63/zabbix/api\_jsonrpc.php"  
headers = {  
 "Content-Type": "application/json-rpc"  
}  
*#首先登录获取token*data = {  
 "jsonrpc": "2.0",  
 "method": "user.login",  
 "params": {  
 "user": "Admin",  
 "password": "zabbix"  
 },  
 "id": 1  
 }  
sendData = json.dumps(data)  
response = requests.post(url = api\_url,headers = headers,data = sendData)  
token = response.json().get("result","nothing")  
print("+++++++++++++++++%s+++++++++++++++++++++="%token)  
  
*#然后携带token查询主机*data = {  
 "jsonrpc": "2.0",  
 "method": "host.get",  
 "params": {  
 "filter": {  
 "host": [  
 "Zabbix server",  
 "Linux server"  
 ]  
 }  
 },  
 "auth": token,  
 "id": 1  
 }  
sendData = json.dumps(data)  
response = requests.post(url = api\_url,headers = headers,data = sendData)  
result = response.json()  
print("+++++++++++++++++++++++++++++++++++++++++++=")  
print(result)  
print("+++++++++++++++++++++++++++++++++++++++++++=")**

效果如下



总结：

20.1 基于Zabbix的Python报警

20.2 基于Python的Zabbix开发