Jira 用户组自动授权申请

脚本：

#!/usr/bin/env python

# -\*- coding: utf-8 -\*-

import requests

import json

from flask import Flask

from flask import request

import logging

from requests.auth import HTTPBasicAuth

server = Flask(\_\_name\_\_)

def log\_settings():

log\_path = "/opt/logs/jira\_authorize.log"

logger = logging.getLogger(\_\_name\_\_)

logger.setLevel(level=logging.INFO)

handler = logging.FileHandler(log\_path)

handler.setLevel(logging.INFO)

formatter = logging.Formatter(

'%(asctime)s - %(name)s - %(levelname)s - %(message)s')

handler.setFormatter(formatter)

if not logger.handlers:

logger.addHandler(handler)

return logger

class jiraAuthorize():

logger = log\_settings()

jira\_api\_url = "http://jira.saicmobility.com:8080/rest/api/2"

jira\_auth = HTTPBasicAuth("scmuser", "scm@2020")

req\_headers = {

"Accept": "application/json",

"Content-Type": "application/json"

}

# 判断jira用户是否存在

def get\_user(self, user\_name):

req\_url = '{}/user'.format(self.jira\_api\_url)

req\_params = {'username': user\_name}

response = requests.get(req\_url, params=req\_params, headers=self.req\_headers, auth=self.jira\_auth)

if response.status\_code == 200:

return 1

else:

return 0

# 将jira用户加入到申请的标的用户组中

def add\_user\_to\_group(self, user\_name, group\_name):

req\_url = '{}/group/user'.format(self.jira\_api\_url)

req\_params = {'groupname': group\_name}

play\_load = {

'name': user\_name

}

response = requests.post(req\_url, params=req\_params, data=json.dumps(play\_load),

headers=self.req\_headers, auth=self.jira\_auth)

if response.status\_code == 201:

self.logger.info("已经将用户：{} 加入到用户组：{} 中".format(user\_name, group\_name))

return 2

elif response.status\_code == 400:

self.logger.info("用户：{} 已经存在于用户组：{} 中，无需再次添加".format(user\_name, group\_name))

return 1

else:

self.logger.info("异常的返回码：{}".format(response.status\_code))

return 0

# 对issue进行评论，告知用户授权结果

def make\_comment(self, issue\_id, content):

comment\_url = "%s/issue/%s/comment" % (self.jira\_api\_url, issue\_id)

payload = json.dumps({

"body": content

})

headers = {

"Accept": "application/json",

"Content-Type": "application/json"

}

requests.request(

"POST",

comment\_url,

data=payload,

headers=headers,

auth=self.jira\_auth

)

# 对issue进行流转

def make\_transition(self, issue\_id, status):

if status == 'succ':

target\_code = '31'

elif status == 'fail':

target\_code = '41'

payload = json.dumps({

"transition": {

"id": target\_code

}

})

headers = {

"Accept": "application/json",

"Content-Type": "application/json"

}

transition\_url = "%s/issue/%s/transitions" % (self.jira\_api\_url, issue\_id)

res = requests.request(

"POST",

transition\_url,

data=payload,

headers=headers,

auth=self.jira\_auth

)

self.logger.info("流转结果：{}".format(res.text))

if res.status\_code == 204:

self.logger.info("issue：{} 状态流转成功".format(issue\_id))

else:

self.logger.info("issue：{} 状态流转失败".format(issue\_id))

def return\_msg(self, msg, msg\_code):

res = {"msg": msg, "msg\_code": msg\_code}

return json.dumps(res)

cmd = jiraAuthorize()

@server.route("/jiraauthorize", methods=["POST"])

def do\_authorize():

res\_data\_origin = request.data

cmd.logger.info("获取到的jira原始请求数据为：{}".format(res\_data\_origin))

res\_data = json.loads(res\_data\_origin.decode())

user\_name\_list = res\_data['issue']['fields']['customfield\_16400'].split(',')

group\_name\_list = res\_data['issue']['fields']['customfield\_20707']

issue\_id = res\_data['issue']['key']

cmd.logger.info("issue: {}\n待授权的用户列表：{}\n希望将用户添加至以下jira\_group：{}"

.format(issue\_id, user\_name\_list, group\_name\_list))

for user in user\_name\_list:

user = user.strip()

user\_exist\_flag = cmd.get\_user(user)

if user\_exist\_flag == 0:

content = ("授权失败！用户: {} 不存在，请检查用户名拼写".format(user))

cmd.make\_comment(issue\_id, content)

cmd.make\_transition(issue\_id, 'fail')

msg = cmd.return\_msg(content, 404)

return msg

else:

pass

for group in group\_name\_list:

cmd.add\_user\_to\_group(user, group)

# 不管用户是否选择了jira-software-users这个组，都给用户加一下这个组的权限

cmd.add\_user\_to\_group(user, 'jira-software-users')

content = "授权成功"

cmd.make\_comment(issue\_id, content)

cmd.make\_transition(issue\_id, 'succ')

msg = cmd.return\_msg(content, 200)

return msg

server.run(host="0.0.0.0", port=7779, debug=True)

工作流：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图示

描述已自动生成