**Python模块**

## **目录**

### **JSON模块**

### **PICKLE模块**

### **configparser模块**

### **XML**

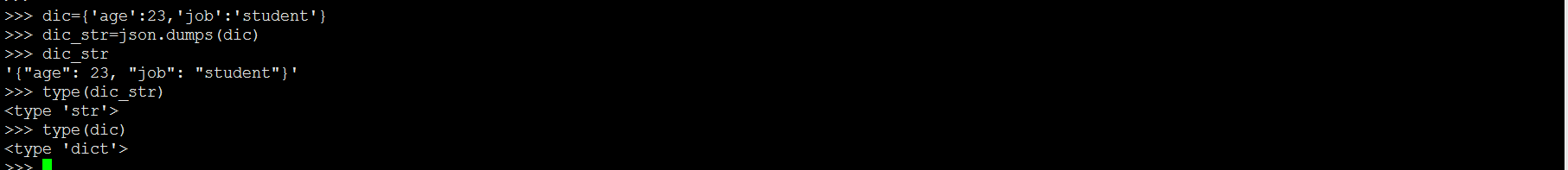
### **Requests模块**

### **6．Request调用gerrit api**

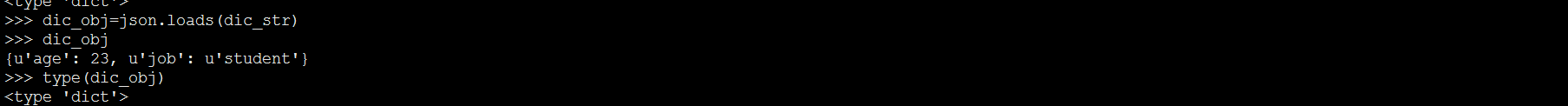
### **Json模块**

**4个功能：dump dumps load loads**

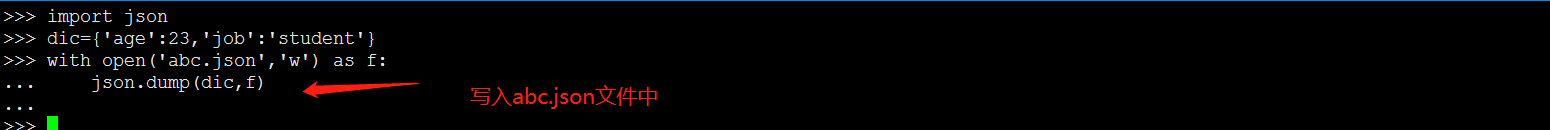
**Dumps:将字典转换成字符串**

****

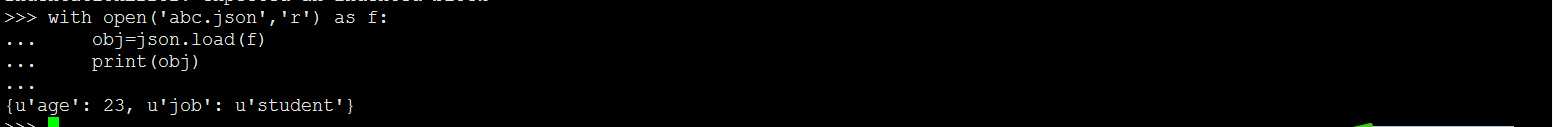
**Loads:将字符串转换成字典**

****

**Dump:字典存储到文件中**

****

**Load:读取文件中的内容**

****

**===============================================================================**

### **pickle模块**

**4个功能：dumps loads dump load**

**dumps:存入数据**

**>>> dic={'age':23,'job':'student'}**

**>>> byte\_data=pickle.dumps(dic) 将字典转换成字符串**

**>>> byte\_data**

**"(dp0\nS'age'\np1\nI23\nsS'job'\np2\nS'student'\np3\ns."**

**loads:读取数据**

**>>> obj=pickle.loads(byte\_data)**

**>>> obj**

**{'age': 23, 'job': 'student'}**

**存储数据到文件中，使用dump和load。由于pickle写入的是二进制数据，所以打开方式需要以wb和rb的模式。**

### **==============================================================configparser模块,只能在python3中使用**

### **Configparser是用来读取配置文件的包,配置文件格式如下:**

**1 [db] ([]为section)**

**2 db\_host = 127.0.0.1 (key=value为option)**

**3 db\_port = 69**

**4 db\_user = root**

**5 db\_pass = root**

**6 host\_port = 69**

**7**

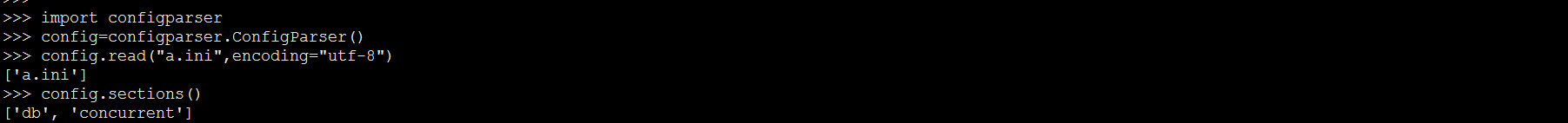
**8 [concurrent]**

**9 thread = 10**

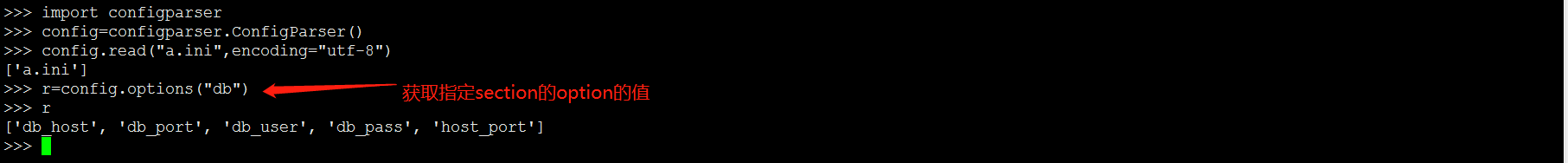
**10 processor = 20**

**案例:**

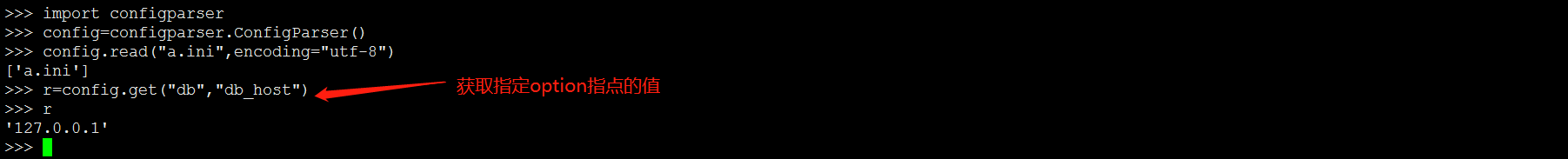
**(1): config.sections()得到所有sections的值**

****

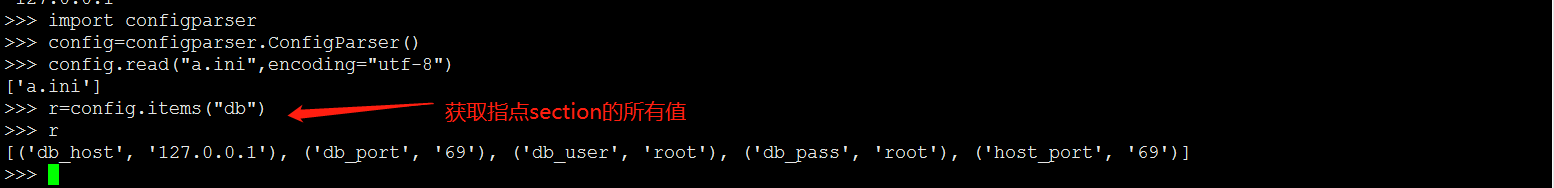
**(2):获取指定section中的option的值**

****

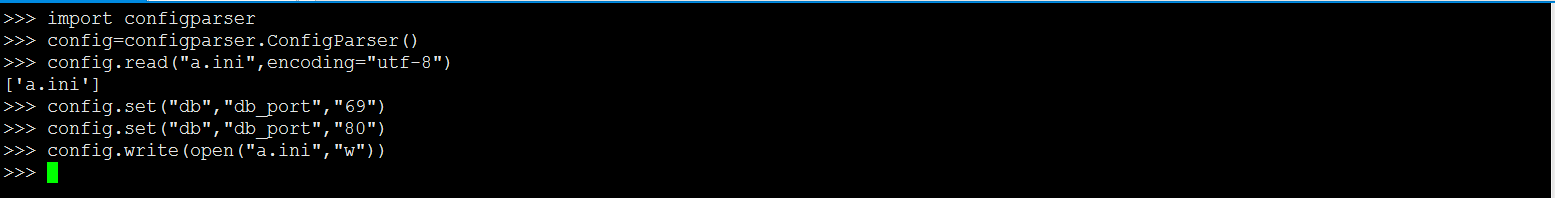
**(3):获取指定option指点的值**

****

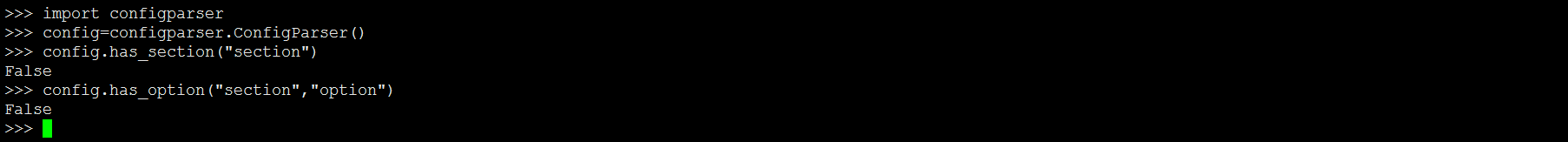
**(4):获取指点section的所有值,会把option的键和值全部输出**

****

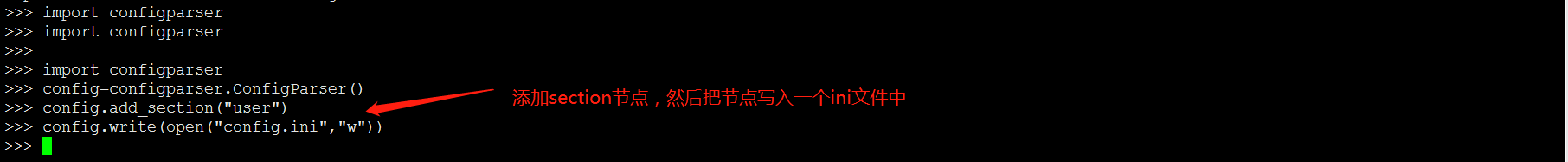
**(5): 修改某个option的值，如果不存在则会出创建**

****

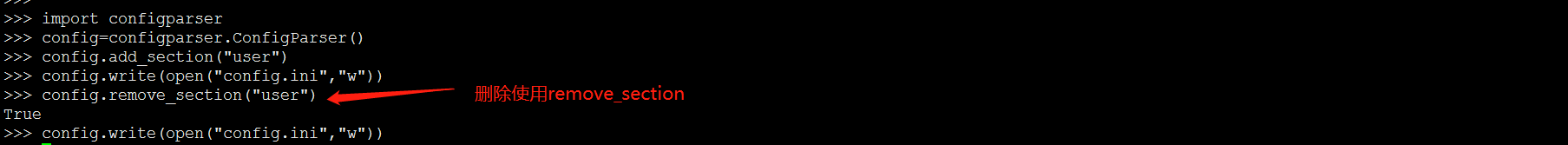
**(6): 检查section或option是否存在，返回bool值**

****

**(7):添加section节点**

****

**（8）：删除section节点**

****

**（9）：删除指定section组内的option**

**>>> import configparser**

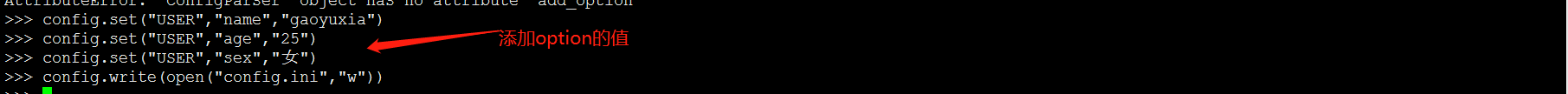
**>>> config=configparser.ConfigParser()**

**>>> config.read("a.ini")**

**>>> config.remove\_option("db","host\_port")**

**>>> config.write(open("a.ini","w"))**

**(10)：添加option的键和值：**

****

### **XML**

**解析XML**

**读取xml文件的内容：**

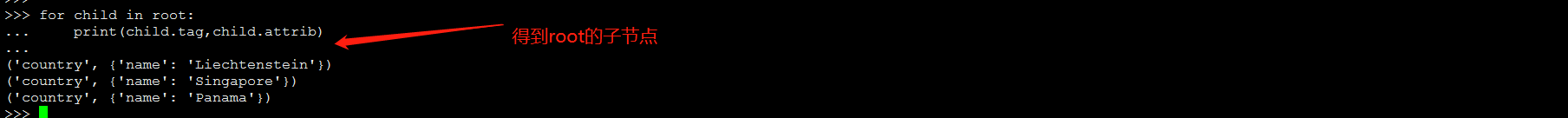
**>>> from xml.etree import ElementTree as ET**

**>>> tree=ET.parse("xo.xml") ====》解析XML**

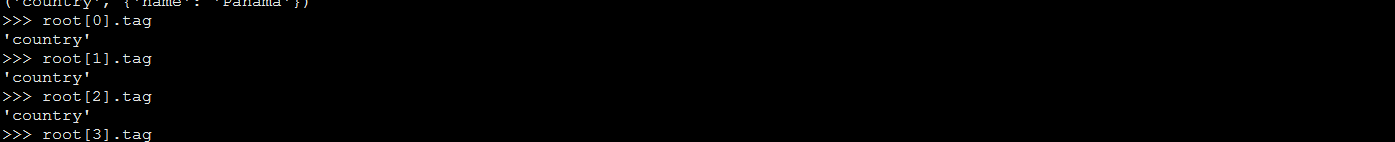
**>>> root=tree.getroot() ==🡺得到Root节点**

**>>> print(root.tag,root.attrib) ==》tag是string,attrib是字典{}**

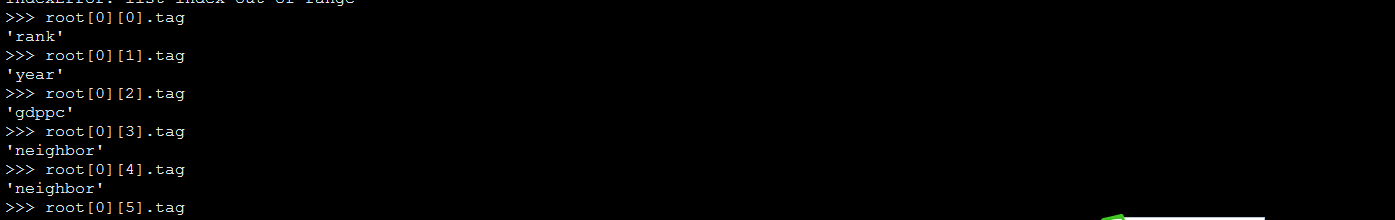
**Root的子节点**

****

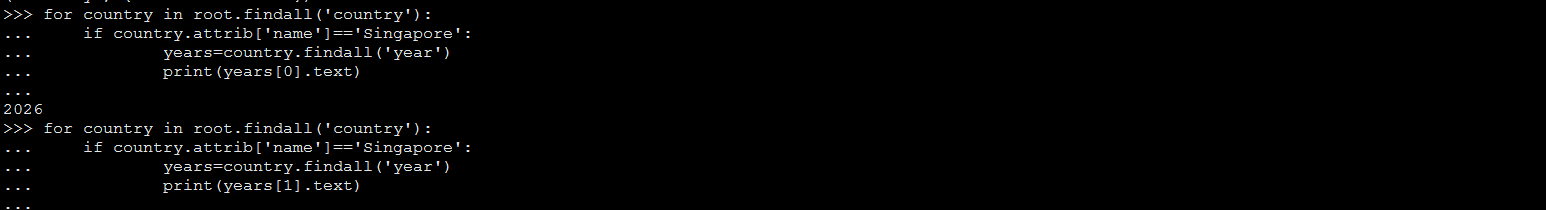
**Root的子节点的key**

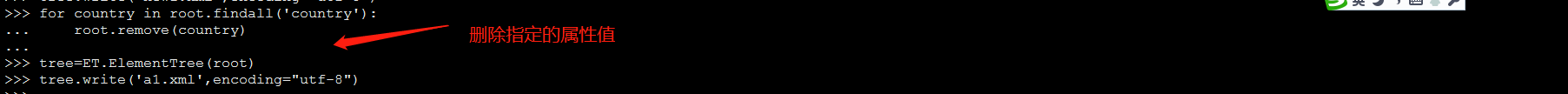
****

**Root的子节点的子节点的key**

****

**Element.findall()只查找直接的孩子，返回所有符合要求的Tag的Element，而Element.find()只返回符合要求的第一个Element。如果查看Singapore的year的值，可以**

****

****

**>>> for country in root.findall("country"):**

**... rank=country.find("rank").text rank的内容**

**... name=country.get("name") country的name属性**

**... print(name,rank)**

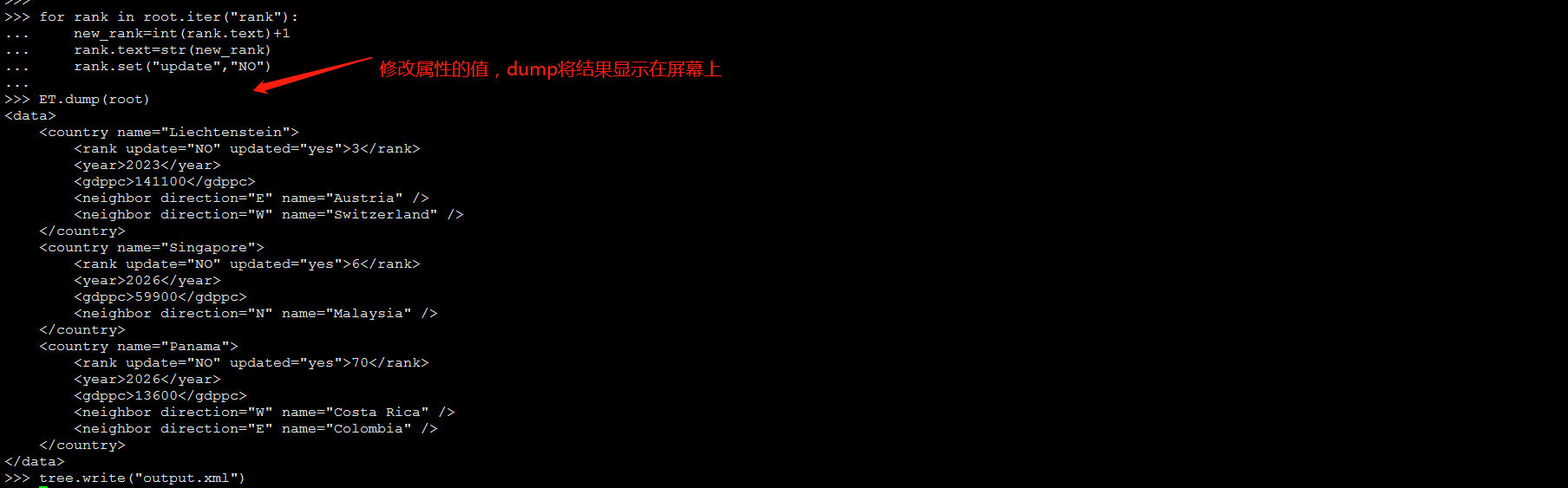
**...**

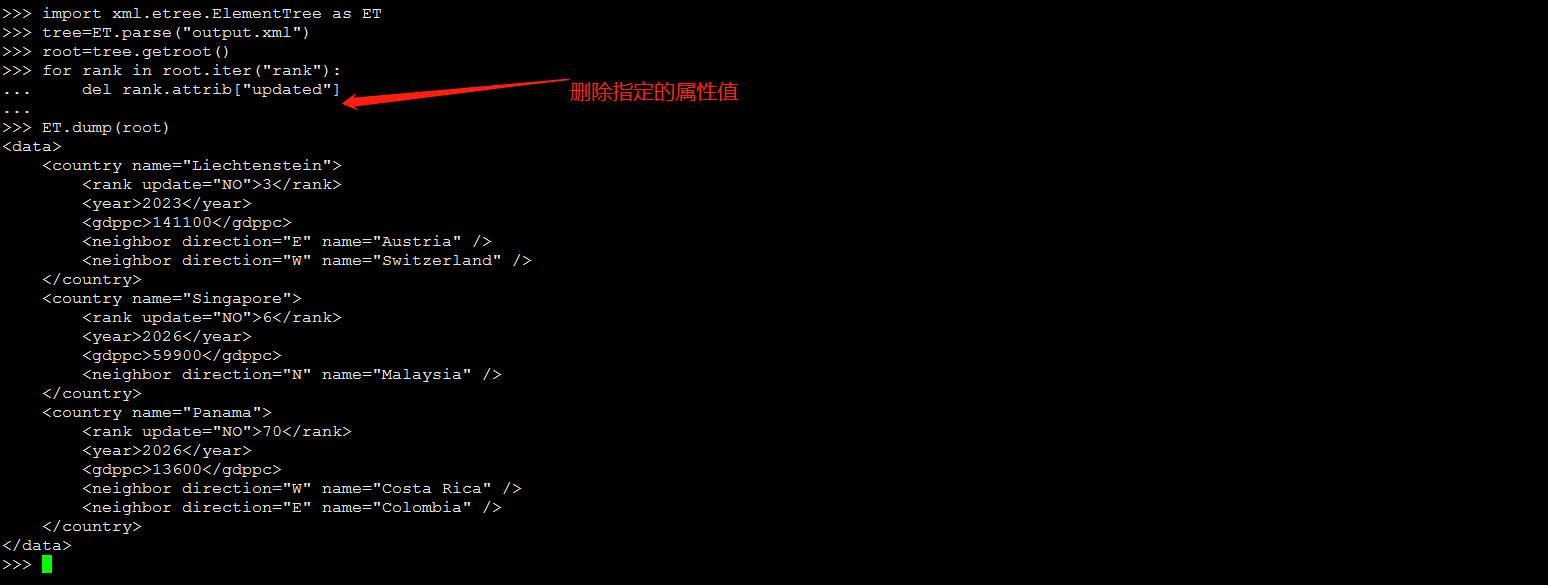
**Liechtenstein 2**

**Singapore 5**

**Panama 69**

**修改XML**

****

****

**小结: 关于*class*xml.etree.ElementTree.Element 属性相关**

* **attrib                 　　              为包含元素属性的字典**
* **keys()             返回元素属性名称列表**
* **items()                                   返回(name,value)列表**
* **get(*key*, *default=None*)         获取属性**
* **set(*key*, *value*)                      # 跟新/添加  属性**
* **del xxx.attrib[key]                  # 删除对应的属性**
* **ET.dump(root) 很有用可以在屏幕上显示输出结果,(括号中输入不同的值，会输出不同的结果)**

### **OS模块学习**

**for root,dirs,files in os.walk(".",topdown=False):**

**... print(root,dirs,files)**

**Root: 当前正在遍历的这个文件夹的本身的地址**

**Dirs.: 是一个 list ，内容是该文件夹中所有的目录的名字(不包括子目录)**

**Files: 是一个 list , 内容是该文件夹中所有的文件(不包括子目录)**

**1：使用 python 遍历一个目录 查找文件。(在一套源码中查找Version文件，并返回它所在的目录)**

**1 #!/usr/bin/python**

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

**3**

**4 import os**

**5**

**6 for root,dirs,files in os.walk("wind",topdown=True):**

**7 for file in files:**

**8 if filename=="version":**

**9 print(root,"找到了！")**

**2：使用python给manifest.xml添加path属性值**

**分析：首先使用ElementTree解析XML文件，并读取XML**

**代码如下：**

**#!/usr/bin/python**

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

**#给xml添加path属性**

**from xml.etree import ElementTree as ET**

**#解析XML文件**

**tree=ET.parse("manifest.xml")**

**#得到根节点**

**root=tree.getroot()**

**for node in root.iter("project"):**

**list1=node.attrib**

**if "path" not in list1.keys():**

**name=list1["name"]**

**#node.set("path",list1["name"])**

**node.attrib["path"]=name**

**tree.write("abc.xml")**

### **requests模块**

**安装：sudo apt-get install python-requests**

**请求方式：**

**Get**

**（1）：基本get请求**

**import requests**

**response = requests.get('http://httpbin.org/get')**

**print(response.text)**

**（2）：带参数get请求**

### **方法一：**

**>>> import requests**

**>>> response=requests.get("http://httpbin.org/get?name=germey&age=22")参数的传**

**递方式**

### **方法二：**

**>>> import requests**

**>>> data={**

**... "name":"germey",**

**... "age":23**

**... }**

**>>> response=requests.get("http://httpbin.org/get",params=data)**

**>>> print(response.text)**

**Put**

**Delete**

**Post:**

**>>> import requests**

**>>> data={"name":"tom","age":"22"}**

**>>> response=requests.post("http://httpbin.org/post",data=data)**

**>>> print(response.text)**

### **Gerrit rest api学习**

**>>> import requests**

**>>> from requests.auth import HTTPBasicAuth**

**>>> from requests.auth import HTTPDigestAuth**

**>>> import json**

**>>> auth=HTTPBasicAuth("gerrit","WspgVEG3Bp0XOqqox6km2h2a+H9WZXAQJ3Gc0yGPGA")**

**>>> data=json.dumps({"name"})**

**>>> header={'content-type':'application/json;charset=UTF-8'}**

**上面的几名话，为调用rest api的指定格式**

**1：查询指定用户的ID**

**语法：**

'GET / accounts / [{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id) '

**>>> import requests**

**>>> from requests.auth import HTTPBasicAuth**

**>>> from requests.auth import HTTPDigestAuth**

**>>> auth=HTTPBasicAuth("gerrit","FfP+4QVtxspyEj2+VXb8VUKpQxzLPad+TfffhAGrYw")**

**>>> response=requests.get("http://192.168.56.101:8081/a/accounts/?q=name:gerrit",auth=auth)**

**>>> print(response.text)**

**)]}'**

**[**

**{**

**"\_account\_id": 1**

**}**

**]2：查询指定用户的信息**

**语法：**

'GET / accounts / [{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id) '

**>>> import requests**

**>>> from requests.auth import HTTPBasicAuth**

**>>> from requests.auth import HTTPDigestAuth**

**>>> auth=HTTPBasicAuth("gerrit","FfP+4QVtxspyEj2+VXb8VUKpQxzLPad+TfffhAGrYw")**

**>>> response=requests.get("http://192.168.56.101:8081/a/accounts/gerrit",auth=auth)**

**>>> print(response.text)**

**)]}'**

**{**

**"\_account\_id": 1,**

**"name": "gerrit",**

**"email": "gerrit@byd.com",**

**"username": "gerrit"**

**})]}'**

**{**

**"\_account\_id": 1,**

**"name": "gerrit",**

**"email": "gerrit@byd.com",**

**"username": "gerrit"**

**}**

**注意：默认情况下，所有REST端点都假定匿名访问和过滤结果与匿名用户可以读取的内容相对应（可能根本没有任何内容）。**

**用户（和程序）可以通过为端点URL添加前缀来进行身份验证 /a/。例如，要进行身份验证/projects/，请求URL/a/projects/。**

**Gerrit使用HTTP基本身份验证和用户帐户设置页面中的HTTP密码**

**3：列出All-Projects的访问权限**

**>>> import requests**

**>>> from requests.auth import HTTPBasicAuth**

**>>> from requests.auth import HTTPDigestAuth**

**>>> import json**

**>>> auth=HTTPBasicAuth("gerrit","WspgVEG3Bp0XOqqox6km2h2a+H9WZXAQJ3Gc0yGPGA")**

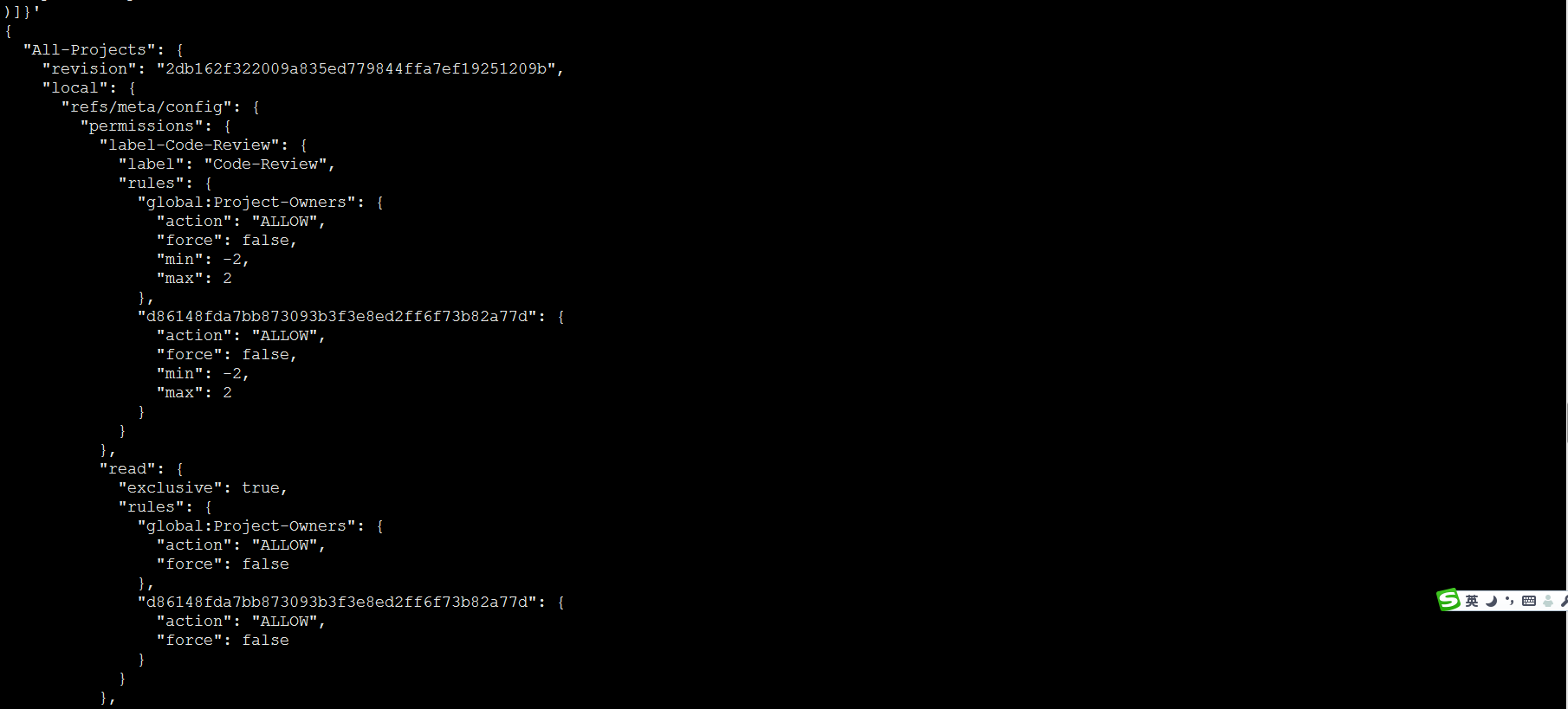
**>>> data=json.dumps({"name"})**

**>>> header={'content-type':'application/json;charset=UTF-8'}**

**>>> response=requests.get("http://192.168.56.101:8081/a/access/?project=All-Projects",auth=auth)**

**>>> print(response.text)、**

**输出结果如下图：**

****

**4：获取指定用户信息**

**语法：**

**方法一：（通过用户名称获取用户信息）**

**>>> response=requests.get("http://192.168.56.101:8081/a/accounts/?suggest&q=gerrit",auth=auth)**

**>>> print(response.text)**

**方法二：（通过用户ID获取用户信息）**

**>>> response=requests.get("http://192.168.56.101:8081/a/accounts/self",auth=auth)**

**>>> print(response.text)**

**5：创建一个新帐户**

**语法：**

'PUT / accounts / [{username}](http://192.168.56.101/Documentation/rest-api-accounts.html#username) '

**>>> response=requests.put("http://192.168.56.101:8081/a/accounts/chuisholing",auth=auth)**

**>>> print(response.text)**

**6：获取用户的详细信息**

**语法：**

'GET / accounts / [{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id) / detail'

**>>> response=requests.get("http://192.168.56.101:8081/a/accounts/chuisholing/detail",auth=auth)**

**>>> print(response.text)**

**获取当前用户的详细信息**

**>>> response=requests.get("http://192.168.56.101:8081/a/accounts/self/detail",auth=auth)**

**>>> print(response.text)**

**7：获取帐户的名称：**

**语法：**

'GET / accounts / [{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id) / name'

**>>> response=requests.get("http://192.168.56.101:8081/a/accounts/2/name",auth=auth)**

**>>> print(response.text)**

**8：删除帐户名称**

**语法：**

'DELETE / accounts / [{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id) / name'

**>>> response=requests.delete("http://192.168.56.101:8081/a/accounts/4/name",auth=auth)**

**>>> print(response.text)**

**这里删除的是FULL NAME**

****

**9：设置帐户名称：（没看明白）**

**语法：**

'PUT / accounts / [{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id) / name'

**>>> data=json.dumps({"name":"admin"})**

**>>> res=requests.put("http://192.168.56.101:8081/a/accounts/5/admin",auth=auth,data=data,headers=header)**

**>>> print(res.text)**

**9：获取帐户状态**

**语法：**

'GET /accounts/[{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id)/status'

**>>> response=requests.get("http://192.168.56.101:8081/a/accounts/1/status",auth=auth)**

**>>> print(response.text)**

**10：获取用户名**

**语法：**

'GET /accounts/[{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id)/username'

**>>> res=requests.get("http://192.168.56.101:8081/accounts/1/username",auth=auth)**

**>>> print(res.text)**

**11：检查用户是否有效**

**语法：**

'GET /accounts/[{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id)/active'

**#!/usr/bin/python**

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

**import requests**

**from requests.auth import HTTPBasicAuth**

**from requests.auth import HTTPDigestAuth**

**import json**

**auth=HTTPBasicAuth("gaoyuxia","d2Tz5eu2D9bcsdSpywD/RAe4nFBCUrx7uJhPY/cjWQ")**

**data=json.dumps({"username":"gaoyuxia"})**

**header={'content-type': 'application/json;charset=UTF-8'}**

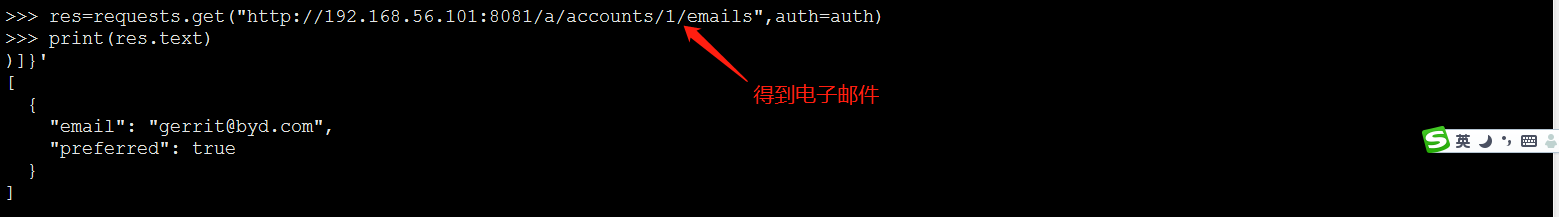
**response=requests.get("http://192.168.56.101:8081/a/accounts/2/active",auth=auth)**

**print(response.text)**

**12：列出帐户电子邮件**

**语法：**

'GET /accounts/[{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id)/emails'

****

**13：创建帐户电子邮件（试验未成功）**

**语法：**

'PUT /accounts/[{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id)/emails/[{email-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#email-id)'

**>>> res=requests.put("http://192.168.56.101:8081/a/accounts/self/emails/yaoyuanchun@wind-mobi.com",auth=auth)**

**>>> print(res.text)**

**14：删除用户的电子邮件（可删除）**

**语法：**

'DELETE /accounts/[{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id)/emails/[{email-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#email-id)'

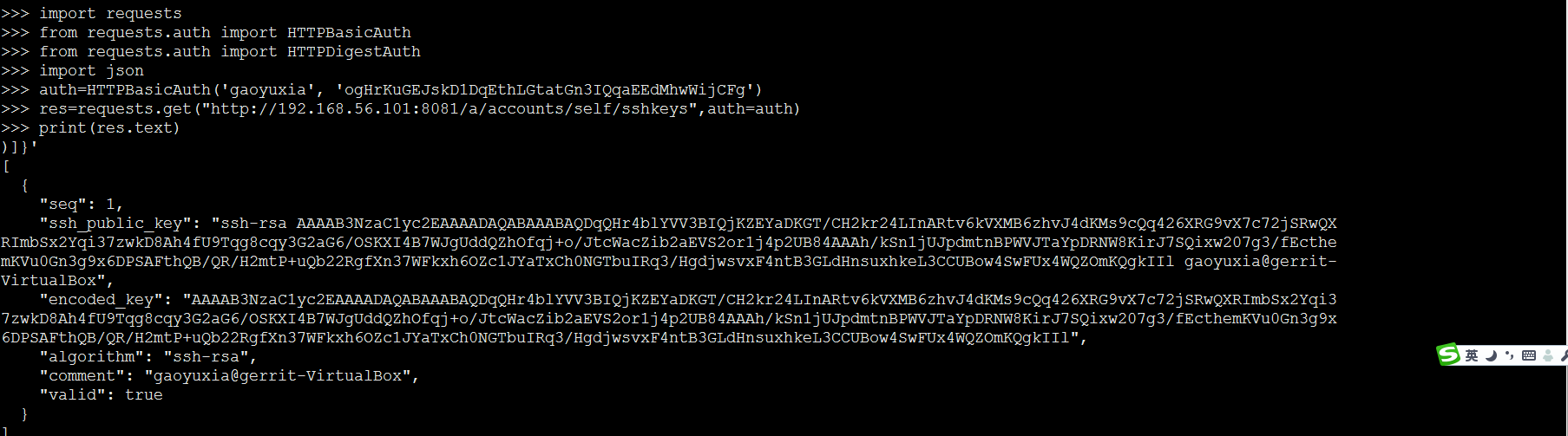
**>>> res=requests.delete("http://192.168.56.101:8081/a/accounts/self/emails/gaoyuxia@wind-mobi.com",auth=auth)**

**>>> print(res.text)**

**15：列出用户的SSH密钥**

**语法：**

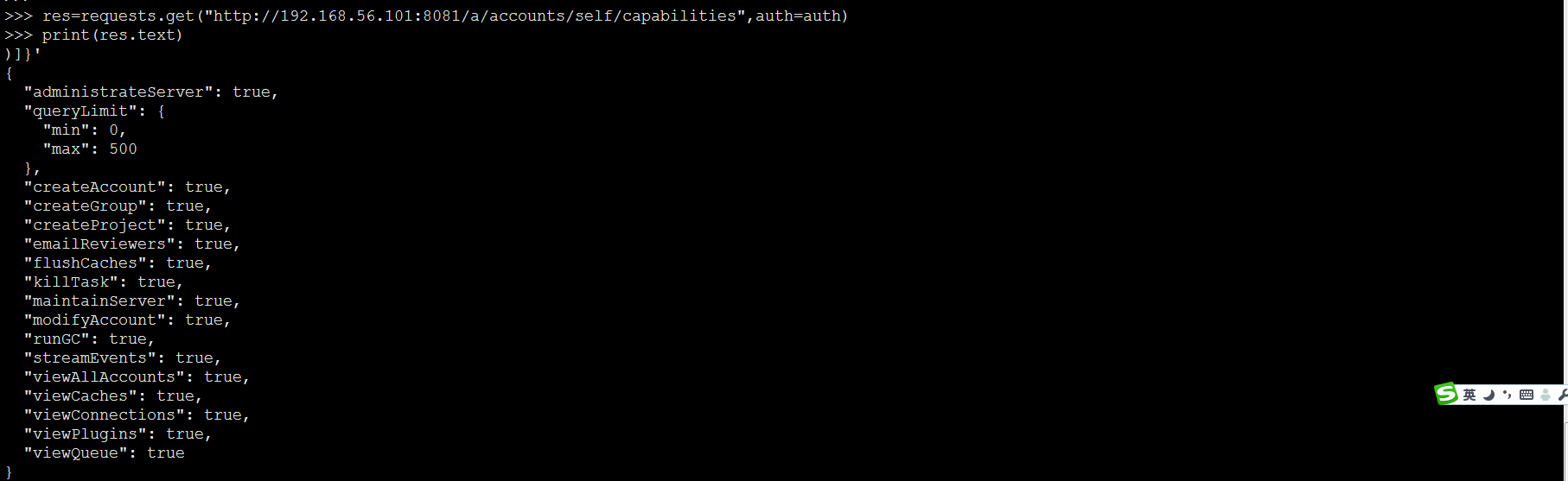
'GET /accounts/[{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id)/sshkeys'

****

**16：列出帐户的功能**

**语法：**

'GET /accounts/[{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id)/capabilities'

****

**17：获取用户指定的功能：**

**语法：**

'GET /accounts/[{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id)/capabilities/[{capability-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#capability-id)'

**>>> res=requests.get("http://192.168.56.101:8081/a/accounts/self/capabilities?q=createAccount&q=createGroup",auth=auth)**

**>>> print(res.text)**

**只获取createAccount和createGroup功能，这里的q用来传递参数**

**18：检查指定用户是否有某项功能：**

**语法：**

**res=requests.get("http://192.168.56.101:8081/a/accounts/self/capabilities?q=createGroup",auth=auth)**

**>>> print(res.text)**

**)]}'**

**{**

**"createGroup": true**

**}**

**返回true有此功能，返回false没有此功能**

**19：列出指定用户所在的所有组**

**语法：**

**>>> res=requests.get("http://192.168.56.101:8081/a/accounts/self/groups/",auth=auth)**

**>>> print(res.text)**

### **CHANGES**

**1：查询已经merged的修改**

**语法：**

'GET /changes/q=status:merge

**>>> res=requests.get("http://192.168.56.101:8081/a/changes/?q=status:merged",auth=auth)**

**>>> print(res.text)**

**2：查询open状态的修改**

**语法：**

'GET /changes/q=status:merge

**>>> res=requests.get("http://192.168.56.101:8081/a/changes/?q=status:open",auth=auth)**

**>>> print(res.text)**

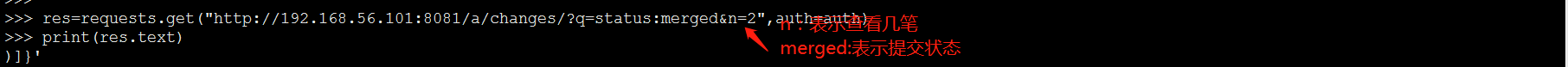
**3：查看最新的公开提交**

**语法：**

'GET /changes/'

**>>> res=requests.get("http://192.168.56.101:8081/a/changes/?q=status:merged&n=2",auth=auth)**

**>>> print(res.text)**

****

**可以通过添加o参数来获取其他字段，**

**参数如下：**

****

****

**4：获取指定change-Id的信息**

**语法：**

'GET /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)'

**>>> res=requests.get("http://192.168.56.101:8081/a/changes/I12d48c0758cce5092167aa8162968bf8c1b20a12",auth=auth)**

**>>> print(res.text)**

### **5：创建合并补丁集以进行更改**

**语法：**

'POST /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/merge'

**>>> res=requests.post("http://192.168.56.101:8081/a/changes/I12d48c0758cce5092167aa8162968bf8c1b20a12/merge",auth=auth)**

**>>> print(res.text)**

**6：获取change-ID的更改细节**

**语法：**

'GET /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/detail'

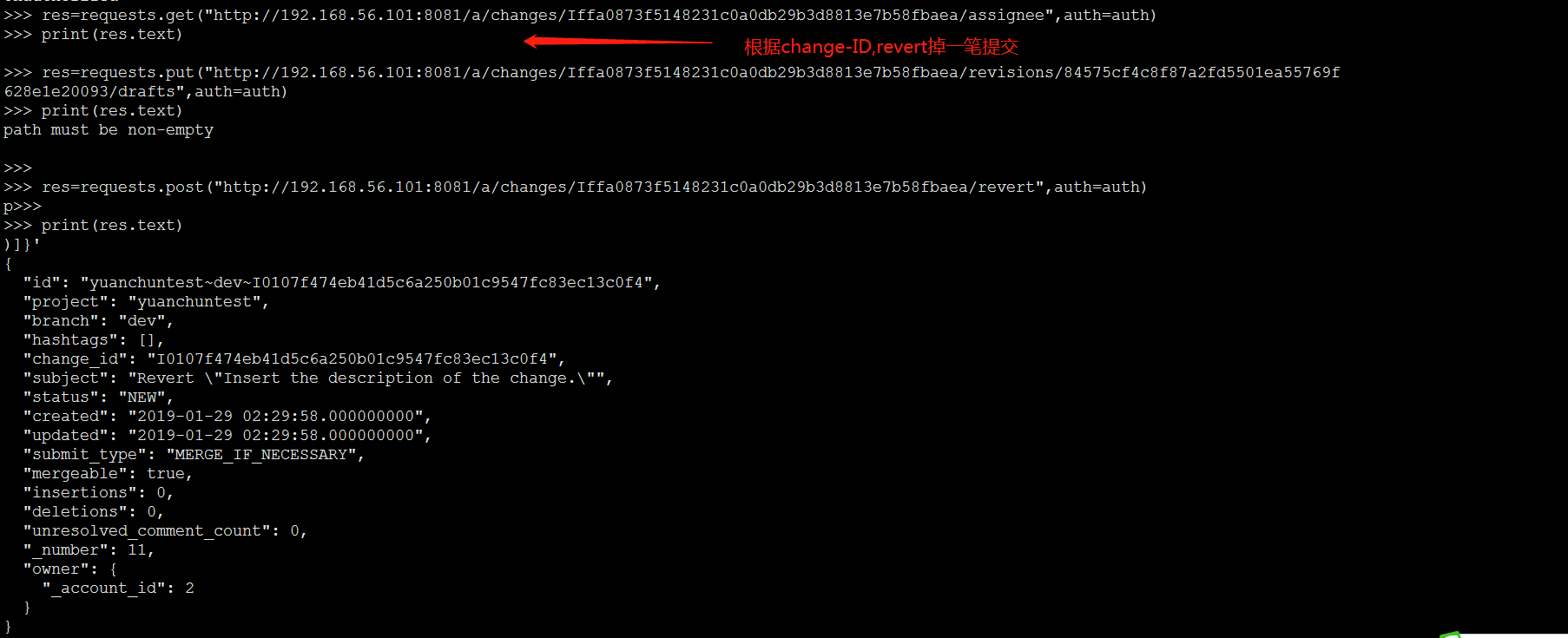
**>>> res=requests.get("http://192.168.56.101:8081/a/changes/I12d48c0758cce5092167aa8162968bf8c1b20a12/detail",auth=auth)**

**>>> print(res.text)**

### **8：revert change**

**语法：**

'POST /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/revert'

****

**此执行结果可以在如下图中获取**

****

### **9：abandon change**

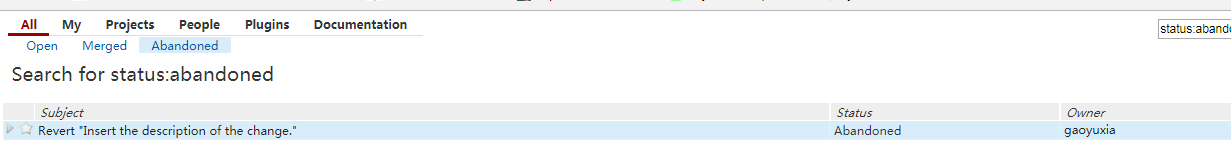
**语法：**

'POST /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/abandon'

**>>> res=requests.post("http://192.168.56.101:8081/a/changes/I0107f474eb41d5c6a250b01c9547fc83ec13c0f4/abandon",auth=auth)**

**>>> print(res.text)**

**执行结果如下图：**

****

### **10：restore change**

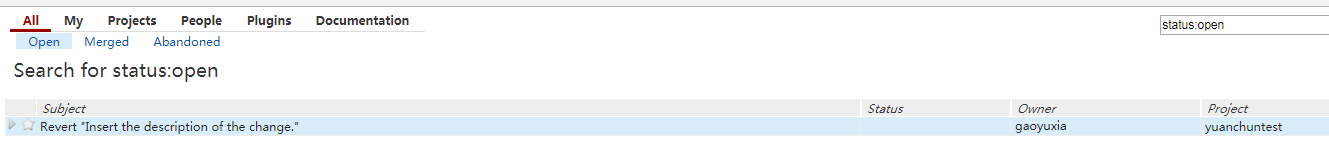
**语法：**

'POST /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/restore'

**>>> res=requests.post("http://192.168.56.101:8081/a/changes/I0107f474eb41d5c6a250b01c9547fc83ec13c0f4/restore",auth=auth)**

**>>> print(res.text)**

**输出结果如下图：**

****

### **11：rebase change**

**语法：**

'POST /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/rebase'

**>>> res=requests.post("http://192.168.56.101:8081/a/changes/I0107f474eb41d5c6a250b01c9547fc83ec13c0f4/rebase",auth=auth)**

**>>> print(res.text)**

### **12:get change**

**语法：**

**>>> res=requests.get("http://192.168.56.101:8081/a/changes/I0107f474eb41d5c6a250b01c9547fc83ec13c0f4",auth=auth)**

**>>> print(res.text)**

**13：get commit**

**语法：**

**>>> res=requests.get("http://192.168.56.101:8081/a/changes/I0107f474eb41d5c6a250b01c9547fc83ec13c0f4/revisions/37d35774f4c1049abc8e0c34d6bd7fb4d23cc7cf/commit",auth=auth)**

**>>> print(res.text)**

**14：get review信息**

**语法：**

**>>> res=requests.get("http://192.168.56.101:8081/a/changes/I0107f474eb41d5c6a250b01c9547fc83ec13c0f4/revisions/37d35774f4c1049abc8e0c34d6bd7fb4d23cc7cf/review",auth=auth)**

**>>> print(res.text)**

**15：提交修改(根据change-ID和revision-ID,提交前需要先code-review和verified)**

**语法：**

**>>> res=requests.post("http://192.168.56.101:8081/a/changes/I0107f474eb41d5c6a250b01c9547fc83ec13c0f4/revisions/37d35774f4c1049abc8e0c34d6bd7fb4d23cc7cf/submit",auth=auth)**

**>>> print(res.text)**

**16:get patch**

**语法：**

**>>> res=requests.get("http://192.168.56.101:8081/a/changes/I0107f474eb41d5c6a250b01c9547fc83ec13c0f4/revisions/37d35774f4c1049abc8e0c34d6bd7fb4d23cc7cf/patch",auth=auth)**

**>>> print(res.text)**

**17:获取提交类型**

**语法：**

**>>> res=requests.get("http://192.168.56.101:8081/a/changes/Ia3f03b1050419e91143e9d4a5f507ed891e63431/revisions/f7948cde5c39730e1ac0be66c8ed86809b6ac04e/submit\_type",auth=auth)**

**>>> print(res.text)**

**)]}'**

**"MERGE\_IF\_NECESSARY"**

**7：获取topic**

**语法：**

'GET /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/topic'

**>>> res=requests.get("http://192.168.56.101:8081/a/changes/I12d48c0758cce5092167aa8162968bf8c1b20a12/topic",auth=auth)**

**>>> print(res.text)**

**8:设置topic**

**语法：**

'PUT /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/topic'

**>>> header={'content-type': 'application/json;charset=UTF-8'}**

**>>> data=json.dumps({"topic": "Documentation"})**

**>>> res=requests.put("http://192.168.56.101:8081/a/changes/I0107f474eb41d5c6a250b01c9547fc83ec13c0f4/topic",auth=auth,data=data,headers=header)**

**>>> print(res.text)**

**9:删除topic(表示没有看懂)**

**语法：**

'DELETE /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/topic'

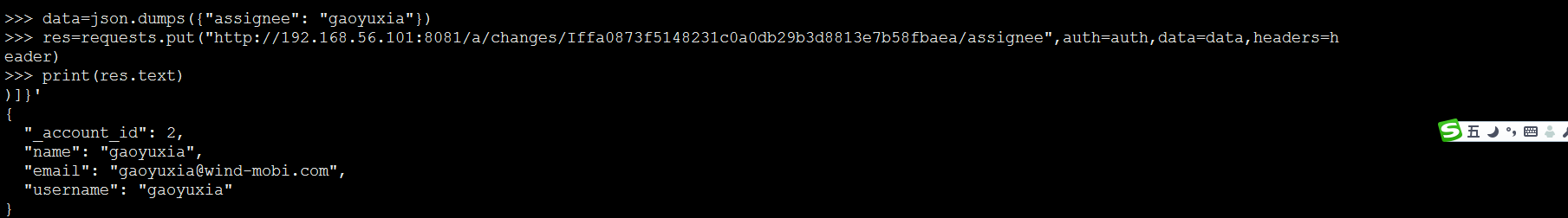
**>>> res=requests.delete("http://192.168.56.101:8081/a/changes/I0107f474eb41d5c6a250b01c9547fc83ec13c0f4/topic",auth=auth)**

**>>> print(res.text)**

### **10: Set Assignee**

**语法：**

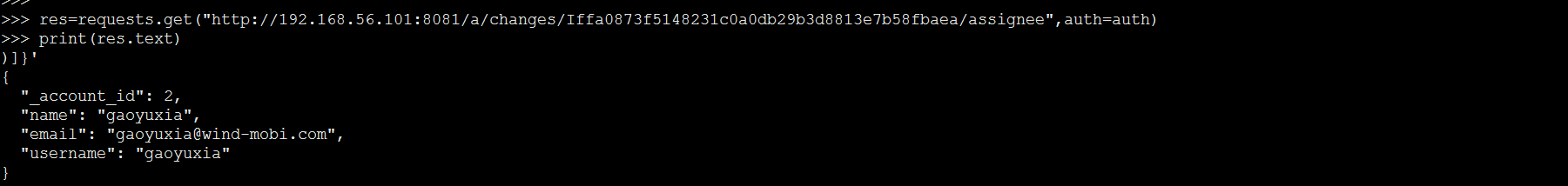
'PUT /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/assignee'

****

### **11:get Assignee**

**语法：**

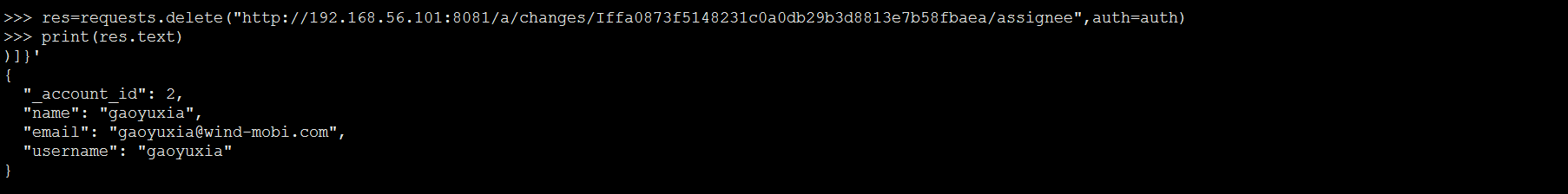
'GET /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/assignee'

****

### **12:delete Assignee**

**语法：**

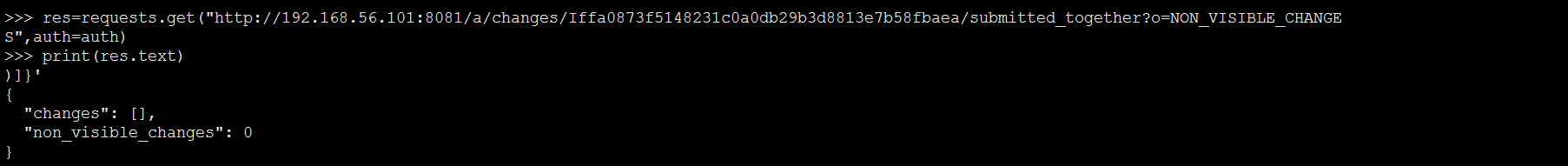
'DELETE /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/assignee'

****

### **13: Changes Submitted Together**

**语法：**

'GET /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/submitted\_together?o=NON\_VISIBLE\_CHANGES'

****

**(没看明白楼上是什么意思)**

### **14: Publish Draft Change(发布一个草稿)**

**语法：**

'POST /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/publish'

**>>> res=requests.post("http://192.168.56.101:8081/a/changes/Idc21039d8474216139c48b19783af240367a89fc/publish",auth=auth)**

**>>> print(res.text)**

### **15:delete change(只有管理员才可以删除)**

**语法：**

'DELETE /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)'

**>>> res=requests.delete("http://192.168.56.101:8081/a/changes/Idc21039d8474216139c48b19783af240367a89fc",auth=auth)**

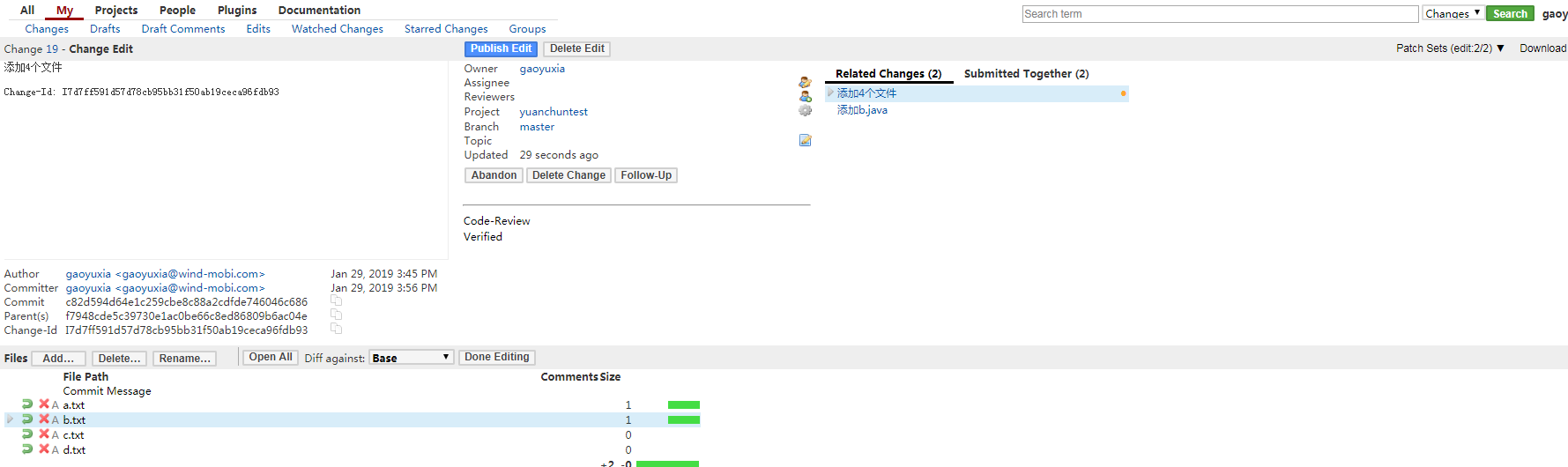
**>>> print(res.text)**

### **16:发布更改的edit**

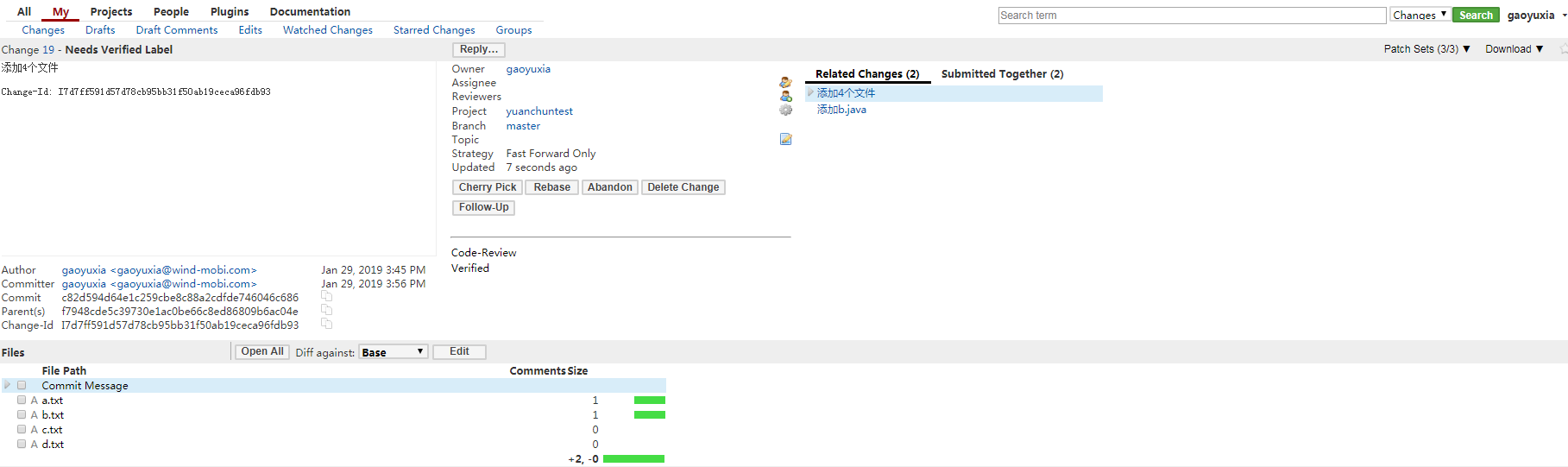
**语法：**

'POST /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/edit:publish

**未发布前的状态显示:**

****

**发布后的状态显示:**

****

**发布指令:**

**>>> res=requests.post("http://192.168.56.101:8081/a/changes/I7d7ff591d57d78cb95bb31f50ab19ceca96fdb93/edit:publish",auth=auth)**

**>>> print(res.text)**

**注意:如果要传递参数,需要加入headers=header这个指令**

**data=json.dumps({'name':'gaoyuxia'})**

**header={'content-type': 'application/json;charset=UTF-8'}**

**r = requests.put('http://192.168.56.102:8083/a/accounts/self/name',auth=auth,data=data,headers=header)**

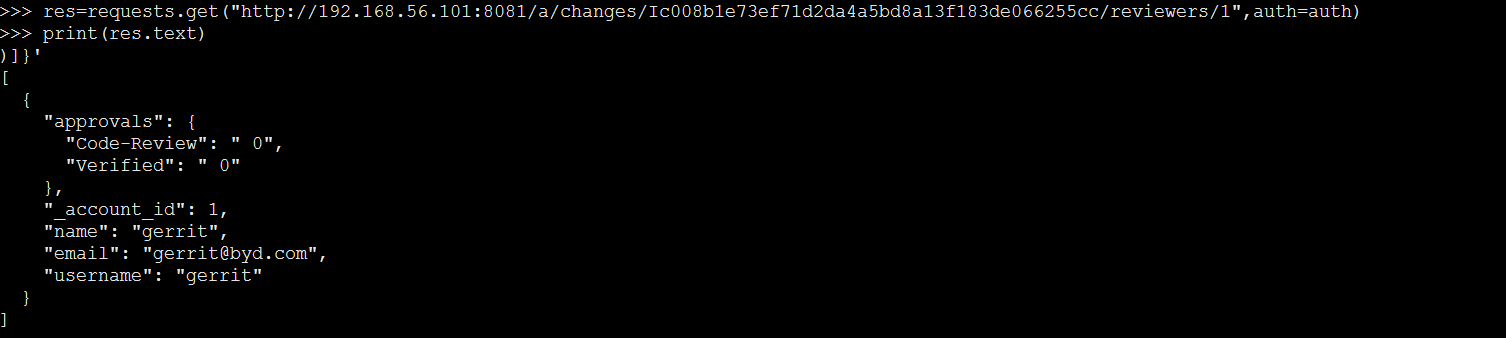
### **17:list reviewers**

语法：

'GET /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/reviewers/'****

### **18:get reviewer**

语法：

'POST /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/reviewers'****

### **19:add reviewer**

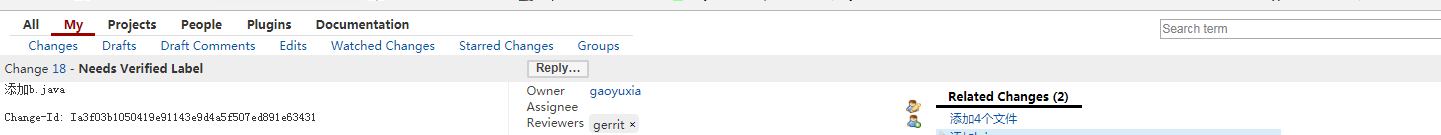
语法：

'POST /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/reviewers'

**>>> user1=json.dumps({"reviewer": "gerrit"})**

**>>> res=requests.post("http://192.168.56.101:8081/a/changes/Ia3f03b1050419e91143e9d4a5f507ed891e63431/reviewers",auth=auth,data=user1,headers=header)**

**>>> print(res.text)**

****

### **20:delete reviewer**

语法：

'DELETE /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/reviewers/[{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id)'  
'POST /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/reviewers/[{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id)/delete'

**>>> res=requests.delete("http://192.168.56.101:8081/a/changes/Ia3f03b1050419e91143e9d4a5f507ed891e63431/reviewers/1",auth=auth,data=user1,headers=header)**

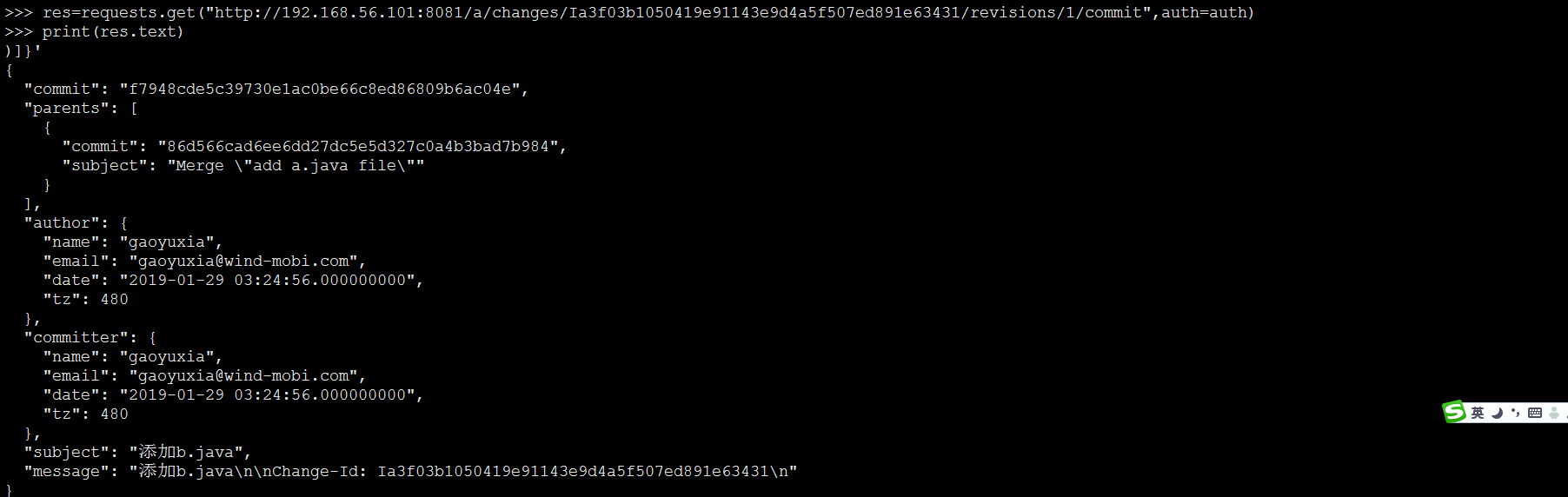
**>>> res=requests.delete("http://192.168.56.101:8081/a/changes/Ia3f03b1050419e91143e9d4a5f507ed891e63431/reviewers/1",auth=auth)**

**>>> print(res.text)**

**21:get commit**

语法：

'GET /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/revisions/[{revision-id}](http://192.168.56.101/Documentation/rest-api-changes.html#revision-id)/commit'

****

**22:get patch**

语法：

'GET /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/revisions/[{revision-id}](http://192.168.56.101/Documentation/rest-api-changes.html#revision-id)/patch'

**>>> res=requests.get("http://192.168.56.101:8081/a/changes/Ic008b1e73ef71d2da4a5bd8a13f183de066255cc/revisions/c772c741fc80fe64f0bb47ebe6149444278ce8ba/patch",auth=auth)**

**>>> print(res.text)**

**23:get submit type**

语法：

'GET /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/revisions/[{revision-id}](http://192.168.56.101/Documentation/rest-api-changes.html#revision-id)/submit\_type'

**>>> res=requests.get("http://192.168.56.101:8081/a/changes/Ic008b1e73ef71d2da4a5bd8a13f183de066255cc/revisions/c772c741fc80fe64f0bb47ebe6149444278ce8ba/submit\_type",auth=auth)**

**>>> print(res.text)**

**)]}'**

**"FAST\_FORWARD\_ONLY"**

**24:list files（得到此笔提交的修改的文件）**

语法：

'GET /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/revisions/[{revision-id}](http://192.168.56.101/Documentation/rest-api-changes.html#revision-id)/files/'

**>>> res=requests.get("http://192.168.56.101:8081/a/changes/I7d7ff591d57d78cb95bb31f50ab19ceca96fdb93/revisions/c82d594d64e1c259cbe8c88a2cdfde746046c686/files",auth=auth)**

**>>> print(res.text)**

**25:get content(得到修改文件的内容，内容以base64编码的字符串形式返回)**

语法：

'GET /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/revisions/[{revision-id}](http://192.168.56.101/Documentation/rest-api-changes.html#revision-id)/files/[{file-id}](http://192.168.56.101/Documentation/rest-api-changes.html#file-id)/content'

**>>> res=requests.get("http://192.168.56.101:8081/a/changes/Id2a64084f1f811b307c01b5b81cc0dcf6784a352/revisions/2ff57de3e9bc192a6940df8157815e94dd8954fb/files/a.java/content",auth=auth)**

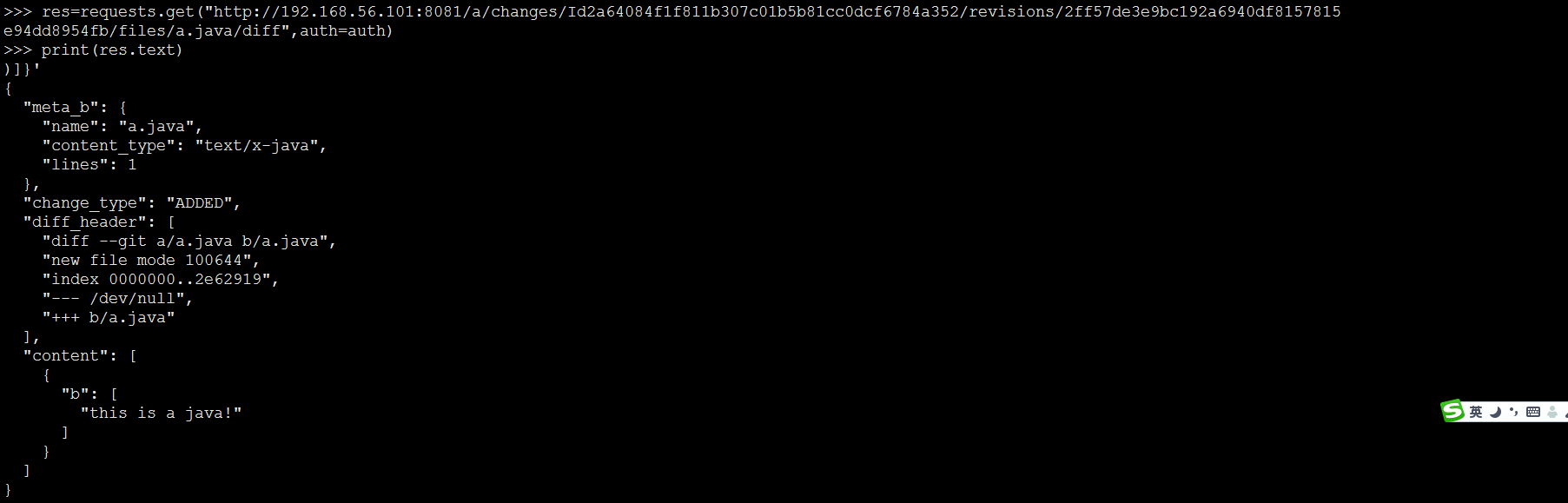
**>>> print(res.text)**

**dGhpcyBpcyBhIGphdmEhCg==**

**26：get diff(此命令可以得到修改文件的内容)**

语法：

'GET /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/revisions/[{revision-id}](http://192.168.56.101/Documentation/rest-api-changes.html#revision-id)/files/[{file-id}](http://192.168.56.101/Documentation/rest-api-changes.html#file-id)/diff'

****

**27:get blame(此命令相当于git blame,可以得到文件每一行的详细修改信息)**

语法：

'GET /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/revisions/[{revision-id}](http://192.168.56.101/Documentation/rest-api-changes.html#revision-id)/files/[{file-id}](http://192.168.56.101/Documentation/rest-api-changes.html#file-id)/blame'

**>>> res=requests.get("http://192.168.56.101:8081/a/changes/Id2a64084f1f811b307c01b5b81cc0dcf6784a352/revisions/2ff57de3e9bc192a6940df8157815e94dd8954fb/files/a.java/blame",auth=auth)**

**>>> print(res.text)**

### **28: Cherry Pick Revision**

语法：

'POST /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/revisions/[{revision-id}](http://192.168.56.101/Documentation/rest-api-changes.html#revision-id)/cherrypick'**#传入参数**

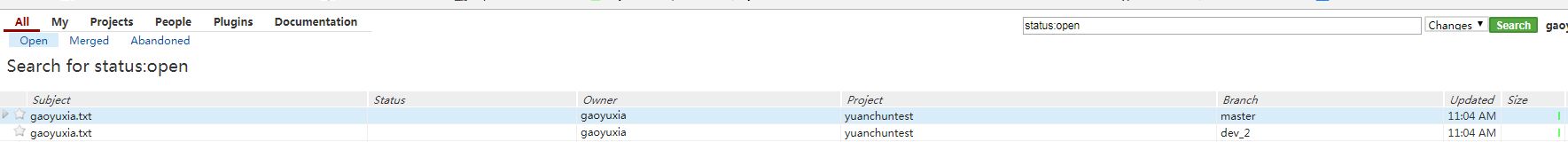
**>>> data=json.dumps({"message":"gaoyuxia.txt","destination":"master"})**

**#输入原PICK的change-id和revision-id**

**>>> res=requests.post("http://192.168.56.101:8081/a/changes/I36c4f6ec7bb69a161a2782321388790efa69fcf2/revisions/9c1c0886aad10b8d8d6e8e99af74931e155c3305/cherrypick",auth=auth,data=data,headers=header)**

**>>> print(res.text)**

**输出结果如下图:**

****

## **29:list revision reviewers**

语法：

'GET /changes/[{change-id}](http://192.168.56.101/Documentation/rest-api-changes.html#change-id)/revisions/[{revision-id}](http://192.168.56.101/Documentation/rest-api-changes.html#revision-id)/reviewers/'

**>>> res=requests.get("http://192.168.56.101:8081/a/changes/I36c4f6ec7bb69a161a2782321388790efa69fcf2/revisions/9c1c0886aad10b8d8d6e8e99af74931e155c3305/reviewers/",auth=auth)**

**>>> print(res.text)**

### **Config篇**

**1:得到当前Gerrit的版本:**

语法：

'GET / config / server / version'

**>>> res=requests.get("http://192.168.56.101:8081/a/config/server/version",auth=auth)**

**>>> print(res.text)**

**)]}'**

**"2.14"**

**2:get server info**

语法：

'GET / config / server / info'

**>>> res=requests.get("http://192.168.56.101:8081/a/config/server/info",auth=auth)**

**>>> print(res.text)**

### **3: List Tasks**

语法：

'GET / config / server / tasks /'

**>>> res=requests.get("http://192.168.56.101:8081/a/config/server/tasks",auth=auth)**

**>>> print(res.text)**

### **Groups篇**

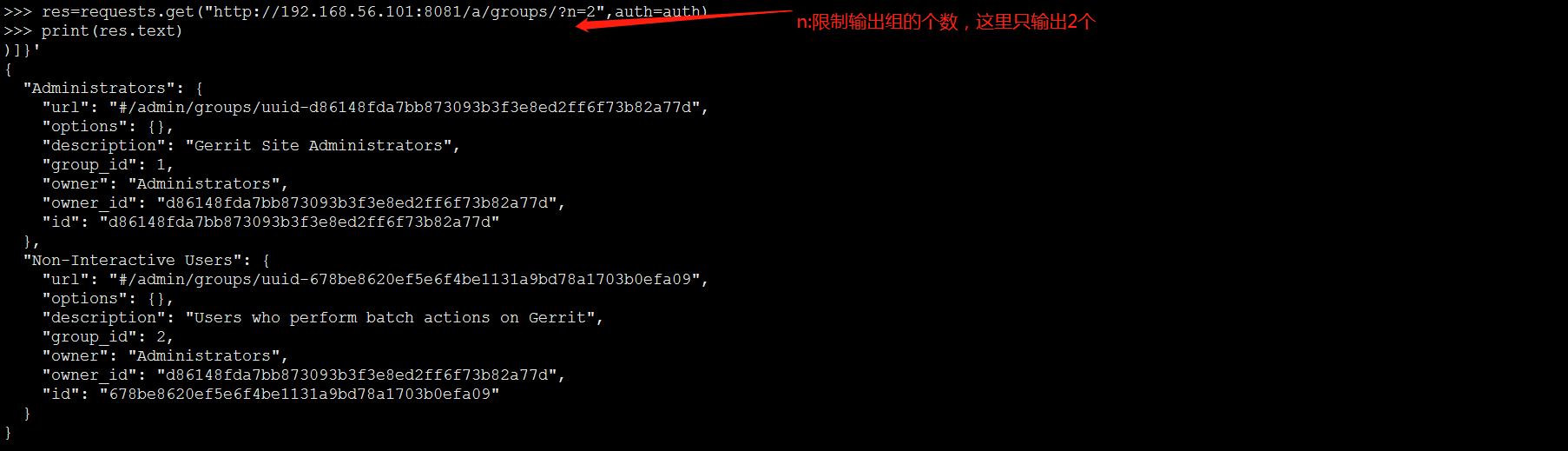
**1:得到所有组(相当于Ssh命令的ls-groups)**

语法：

'GET / groups /'

**>>> res=requests.get("http://192.168.56.101:8081/a/groups/",auth=auth)**

**>>> print(res.text)**

****

**2:创建权限组**

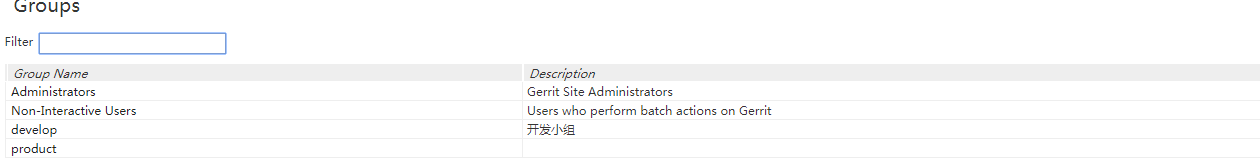
语法：

'PUT / groups / [{group-name}](http://192.168.56.101/Documentation/rest-api-groups.html#group-name) '

**>>> res=requests.put("http://192.168.56.101:8081/a/groups/product",auth=auth)**

**>>> print(res.text)**

**输出结果如下图:**

****

**3:get group detail**

语法：

'GET / groups / [{group-id}](http://192.168.56.101/Documentation/rest-api-groups.html#group-id) / detail'

**>>> res=requests.get("http://192.168.56.101:8081/a/groups/4/detail",auth=auth)**

**>>> print(res.text)**

**4:根据组的ID得到组的名称**

语法：

'GET / groups / [{group-id}](http://192.168.56.101/Documentation/rest-api-groups.html#group-id) / name'

**>>> res=requests.get("http://192.168.56.101:8081/a/groups/4/name",auth=auth)**

**>>> print(res.text)**

**)]}'**

**"product"**

**5:rename group(对组进行重命名)**

语法：

'PUT / groups / [{group-id}](http://192.168.56.101/Documentation/rest-api-groups.html#group-id) / name'

**>>> data=json.dumps({"name":"Review\_Group"})**

**>>> res=requests.put("http://192.168.56.101:8081/a/groups/4/name",auth=auth,data=data,headers=header)**

**>>> print(res.text)**

**)]}'**

**"product"**

**6:得到组的描述信息(get group desripriton)**

语法：

**>>> res=requests.get("http://192.168.56.101:8081/a/groups/3/description",auth=auth)**

**>>> print(res.text)**

**)]}'**

**"开发小组"**

**7:设置组的描述信息**

语法：

'GET / groups / [{group-id}](http://192.168.56.101/Documentation/rest-api-groups.html#group-id) / description'

**>>> data=json.dumps({"description":"此组的成员，拥有读取代码，和下载代码的权限"})**

**>>> res=requests.put("http://192.168.56.101:8081/a/groups/4/description",auth=auth,data=data,headers=header)**

**>>> print(res.text)**

**)]}'**

**"此组的成员，拥有读取代码，和下载代码的权限"**

### **8: Delete Group Description(删除组的描述信息)**

语法：

'DELETE / groups / [{group-id}](http://192.168.56.101/Documentation/rest-api-groups.html#group-id) / description'

**>>> res=requests.delete("http://192.168.56.101:8081/a/groups/3/description",auth=auth)**

**>>> print(res.text)**

**9:设置组选项**

语法：

'PUT / groups / [{group-id}](http://192.168.56.101/Documentation/rest-api-groups.html#group-id) / options'

**>>> data=json.dumps({"visible\_to\_all":"true"})**

**>>> res=requests.put("http://192.168.56.101:8081/a/groups/4/options",auth=auth,data=data,headers=header)**

**>>> print(res.text)**

**)]}'**

**{**

**"visible\_to\_all": true**

**}**

**10:得到组选项(不知道此选项功能)**

语法：

'GET / groups / [{group-id}](http://192.168.56.101/Documentation/rest-api-groups.html#group-id) / options'

**>>> res=requests.get("http://192.168.56.101:8081/a/groups/4/options",auth=auth)**

**>>> print(res.text)**

**)]}'**

**{**

**"visible\_to\_all": true**

**}**

**11:列出组的所有成员**

语法：

'GET / groups / [{group-id}](http://192.168.56.101/Documentation/rest-api-groups.html#group-id) / members /'

**>>> res=requests.get("http://192.168.56.101:8081/a/groups/1/members/",auth=auth)**

**>>> print(res.text)**

**12:添加小组成员**

语法：

'PUT / groups / [{group-id}](http://192.168.56.101/Documentation/rest-api-groups.html#group-id) / members / [{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id) '

'POST /groups/[{group-id}](http://192.168.56.101/Documentation/rest-api-groups.html#group-id)/members'

**>>> data=json.dumps({"username":"gerrit","email":"gerrit@byd.com","name":"gerrit"})**

**#4:group\_id,1: \_account\_id**

**>>res=requests.put("http://192.168.56.101:8081/a/groups/4/members/1/",auth=auth,data=data,headers=header)**

**>>> print(res.text)**

**13：删除组成员**

**语法：**

'DELETE / groups / [{group-id}](http://192.168.56.101/Documentation/rest-api-groups.html#group-id) / members / [{account-id}](http://192.168.56.101/Documentation/rest-api-accounts.html#account-id) '、

'POST / groups / [{group-id}](http://192.168.56.101/Documentation/rest-api-groups.html#group-id) /members.delete'

方法一：

**>>> res=requests.delete("http://192.168.56.101:8081/a/groups/4/members/gerrit",auth=auth)**

**>>> print(res.text)  
方法二：**

**>>> data=json.dumps({"members":["gerrit@byd.com"]})**

**>>> res=requests.post("http://192.168.56.101:8081/a/groups/4/members.delete",auth=auth,data=data,headers=header)**

**>>> print(res.text)**

**13:list include group**

语法：

'GET / groups / [{group-id}](http://192.168.56.101/Documentation/rest-api-groups.html#group-id) / groups /'

**>>> res=requests.get("http://192.168.56.101:8081/a/groups/4/groups",auth=auth)**

**>>> print(res.text)**

### **plugins**

**1:得到已经安装的插件列表**

**语法：**

'GET / plugins /'

**>>> res=requests.get("http://192.168.56.101:8081/a/plugins/",auth=auth)**

**>>> print(res.text)**

**2:获取插件状态**

**语法：**

'GET / plugins / [{plugin-id}](http://192.168.56.101/Documentation/rest-api-plugins.html#plugin-id) / gerrit~status'

**>>> res=requests.get("http://192.168.56.101:8081/a/plugins/reviewnotes/gerrit~status",auth=auth)**

**>>> print(res.text)**

### **Project**

**1:列出所有项目:**

**语法：**

'GET /projects/'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/",auth=auth)**

**>>> print(res.text)**

**2:get project**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest",auth=auth)**

**>>> print(res.text)**

**)]}'**

**{**

**"id": "yuanchuntest",**

**"name": "yuanchuntest",**

**"parent": "All-Projects",**

**"state": "ACTIVE"**

**3:create project**

**语法：**

'PUT /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)'

**>>> data=json.dumps({"description": "This is a demo project","submit\_type":"FAST\_FORWARD\_ONLY"})**

**>>> res=requests.put("http://192.168.56.101:8081/a/projects/MyProject",auth=auth,data=data,headers=header)------------------🡪(这里的MyProject就是要创建的项目名称，可以根据自己的需求填写)**

**>>> print(res.text)**

**创建完成后可在project list中看到，如下图：**

****

**4:get project description**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/description'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuxiaTest/description",auth=auth)**

**>>> print(res.text)**

**5:set project description**

**语法：**

'PUT /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/description'

**>>> data=json.dumps({"description": "This is a manifest repository"})**

**>>> res=requests.put("http://192.168.56.101:8081/a/projects/manifest/description",auth=auth,data=data,headers=header)**

**>>> print(res.text)**

**输出结果如下：**

****

### **6: delete project description**

**语法：**

'DELETE /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/description'

**>>> res=requests.delete("http://192.168.56.101:8081/a/projects/manifest/description",auth=auth)**

**>>> print(res.text)**

### **7: Get Project Parent**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/parent'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/manifest/parent",auth=auth)**

**>>> print(res.text)**

### **8:set project parent**

**语法：**

'PUT /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/parent'

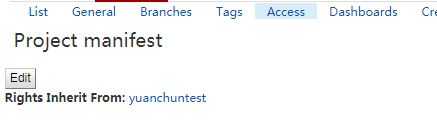
**>>> data=json.dumps({"parent": "yuanchuntest"})**

**>>> res=requests.put("http://192.168.56.101:8081/a/projects/manifest/parent",auth=auth,data=data,headers=header)**

**>>> print(res.text)**

**)]}'**

**"yuanchuntest"**

****

**9:”get HEAD(检索项目的分支名称HEAD)**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/HEAD'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/manifest/HEAD",auth=auth)**

**>>> print(res.text)**

**10:set HEAD**

**语法：**

'PUT /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/HEAD'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/manifest/HEAD",auth=auth)**

**>>> print(res.text)**

### **11: 获取存储库统计信息**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/statistics.git'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/manifest/statistics.git",auth=auth)**

**>>> print(res.text)**

**12:get config**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/config'

**>>> auth=HTTPBasicAuth('gaoyuxia', '+F9MWakj02djuNvPZKdBhuBcY46p50xTJAAeedPCDQ')**

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest/config",auth=auth)**

**>>> print(res.text)**

**13:设置配置**

**语法：**

'PUT /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/config'

**>>> data=json.dumps({"submit\_type":"FAST\_FORWARD\_ONLY"})**

**>>> res=requests.put("http://192.168.56.101:8081/a/projects/yuanchuntest/config",auth=auth,data=data,headers=header)**

**>>> print(res.text)**

**14:运行gc**

**语法：**

'POST /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/gc'

**>>> res=requests.post("http://192.168.56.101:8081/a/projects/yuanchuntest/gc",auth=auth)**

**>>> print(res.text)**

**15:列出项目的访问权限**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/access'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest/access",auth=auth)**

**>>> print(res.text)**

**16:列出项目的所有分支**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/branches/'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest/branches/",auth=auth)**

**>>> print(res.text)**

**17:指定列出分支的数量**

**Limit(n)：传递分支数量**

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest/branches?n=2",auth=auth)**

**>>> print(res.text)**

**18:Skip(s)跳过给定数量的分支**

**s=2：指定跳过前2个分支**

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest/branches?n=2&s=2",auth=auth)**

**>>> print(res.text)**

**19:m参数（没明白有什么用）**

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest/branches?m=master",auth=auth)**

**>>> print(res.text)**

**20:列出所有以dev开头的分支**

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest/branches?r=dev.\*",auth=auth)**

**>>> print(res.text)**

**21：得到分支（可以得到此分支提交的最新的commit-id）**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/branches/[{branch-id}](http://192.168.56.101/Documentation/rest-api-projects.html#branch-id)'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest/branches/master",auth=auth)**

**>>> print(res.text)**

**22：创建分支**

**语法：**

'PUT /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/branches/[{branch-id}](http://192.168.56.101/Documentation/rest-api-projects.html#branch-id)'

**>>> res=requests.put("http://192.168.56.101:8081/a/projects/yuanchuntest/branches/dev\_3",auth=auth)**

**>>> print(res.text)**

**23：删除分支（根据分支名称删除）**

**语法：**

'POST /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/branches:delete'

'DELETE /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/branches/[{branch-id}](http://192.168.56.101/Documentation/rest-api-projects.html#branch-id)'

**>>> res=requests.delete("http://192.168.56.101:8081/a/projects/yuanchuntest/branches/f7948cde5c39730e1ac0be66c8ed86809b6ac04e",auth=auth)**

**>>> print(res.text)**

**24：获取文件内容（返回的文件内容为base64加密内容）**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/branches/[{branch-id}](http://192.168.56.101/Documentation/rest-api-projects.html#branch-id)/files/[{file-id}](http://192.168.56.101/Documentation/rest-api-changes.html#file-id)/content'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest/branches/master/files/a.java/content",auth=auth)**

**>>> print(res.text)**

**25：获取reflog(感觉没什么用)**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/branches/[{branch-id}](http://192.168.56.101/Documentation/rest-api-projects.html#branch-id)/reflog'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest/branches/master/reflog",auth=auth)**

**>>> print(res.text)**

**26：列出子项目(列出继承All-Projects的所有项目)**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/children/'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/All-Projects/children",auth=auth)**

**>>> print(res.text)**

**27：创建tag**

**语法：**

'PUT /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/tags/[{tag-id}](http://192.168.56.101/Documentation/rest-api-projects.html#tag-id)'

**>>> data=json.dumps({"revision":"f7948cde5c39730e1ac0be66c8ed86809b6ac04e"})**

**>>> res=requests.put("http://192.168.56.101:8081/a/projects/yuanchuntest/tags/v1.0",auth=auth,data=data,headers=header)**

**>>> print(res.text)**

**28：列出Tag**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/tags/'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest/tags",auth=auth)**

**>>> print(res.text)**

**29：limit n(输出指定数量的tag)**

**语法：**

GET /projects/work%2Fmy-project/tags?n=2

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest/tags?n=2",auth=auth)**

**>>> print(res.text)**

**30:获取tag**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/tags/[{tag-id}](http://192.168.56.101/Documentation/rest-api-projects.html#tag-id)'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest/tags/v1.0",auth=auth)**

**>>> print(res.text)**

**31:删除tag**

**语法：**

'DELETE /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/tags/[{tag-id}](http://192.168.56.101/Documentation/rest-api-projects.html#tag-id)'

'POST /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/tags:delete'

**>>> import requests**

**>>> from requests.auth import HTTPBasicAuth**

**>>> from requests.auth import HTTPDigestAuth**

**>>> import json**

**>>> auth=HTTPBasicAuth('gaoyuxia', 'iFo5opoSBYTbEkByIR3Qd5mR5yDCANaAdLlcKf08vA')**

**>>> header={'content-type': 'application/json;charset=UTF-8'}**

**>>> res=requests.delete("http://192.168.56.101:8081/a/projects/yuanchuntest/tags/v1.0",auth=auth)**

**>>> print(res.text)**

**32:get commit**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/commits/[{commit-id}](http://192.168.56.101/Documentation/rest-api-projects.html#commit-id)'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest/commits/6ad6ac4b1e90b93e624af4d4308d57fef29189ec",auth=auth)**

**>>> print(res.text)**

**33:get tag in(输出Tag所在分支，以及tag名称等相关信息)**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/commits/[{commit-id}](http://192.168.56.101/Documentation/rest-api-projects.html#commit-id)/in'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest/commits/6ad6ac4b1e90b93e624af4d4308d57fef29189ec/in",auth=auth)**

**>>> print(res.text)**

**34:获取指定文件的内容**

**语法：**

'GET /projects/[{project-name}](http://192.168.56.101/Documentation/rest-api-projects.html#project-name)/commits/[{commit-id}](http://192.168.56.101/Documentation/rest-api-projects.html#commit-id)/files/[{file-id}](http://192.168.56.101/Documentation/rest-api-changes.html#file-id)/content'

**>>> res=requests.get("http://192.168.56.101:8081/a/projects/yuanchuntest/commits/6ad6ac4b1e90b93e624af4d4308d57fef29189ec/files/readme.txt/content",auth=auth)**

**>>> print(res.text)**