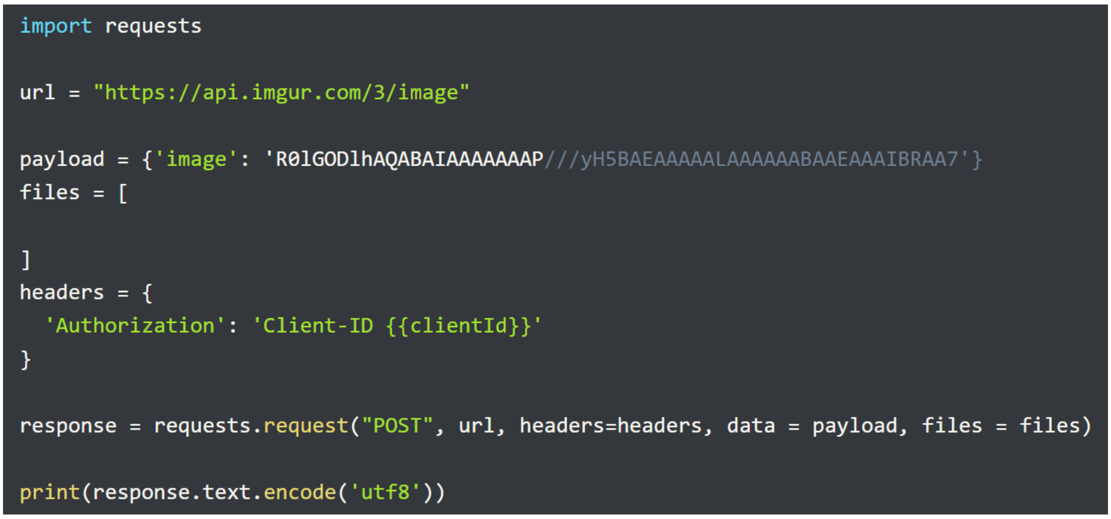
We use imgur as external service. Imgur is an online photo album site, launched in 2009, which offers free storage services for photo and video.

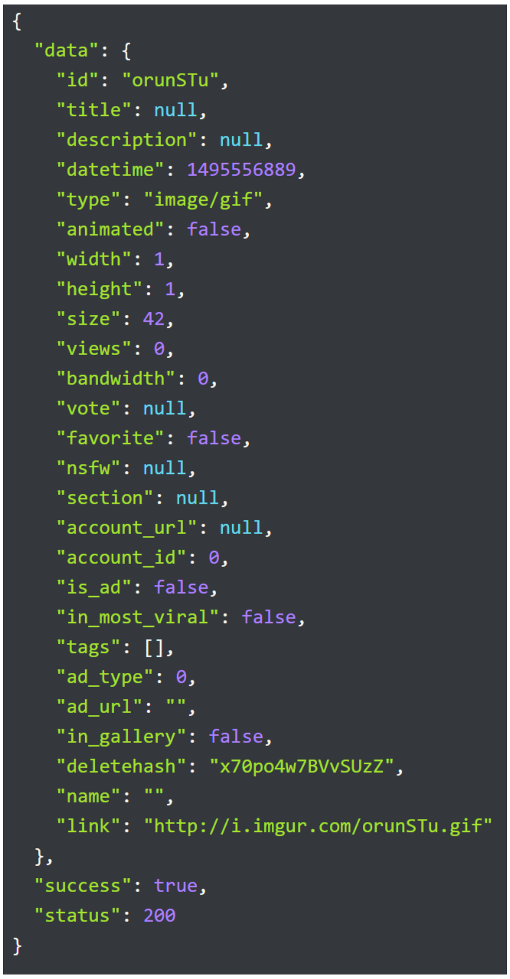
links of tutorial: https://apidocs.imgur.com

Example Request:



```python
import requests

url = "https://api.imgur.com/3/image"

payload = {'image': 'R0lGODlhAQABAIAAAAAAAP///yH5BAEAAAAALAAAAAABAAEAAAIBRAA7'}
files = [

]
headers = {
  'Authorization': 'Client-ID {{clientId}}'
}

response = requests.request("POST", url, headers=headers, data = payload, files = files)

print(response.text.encode('utf8'))
```

Example Response:

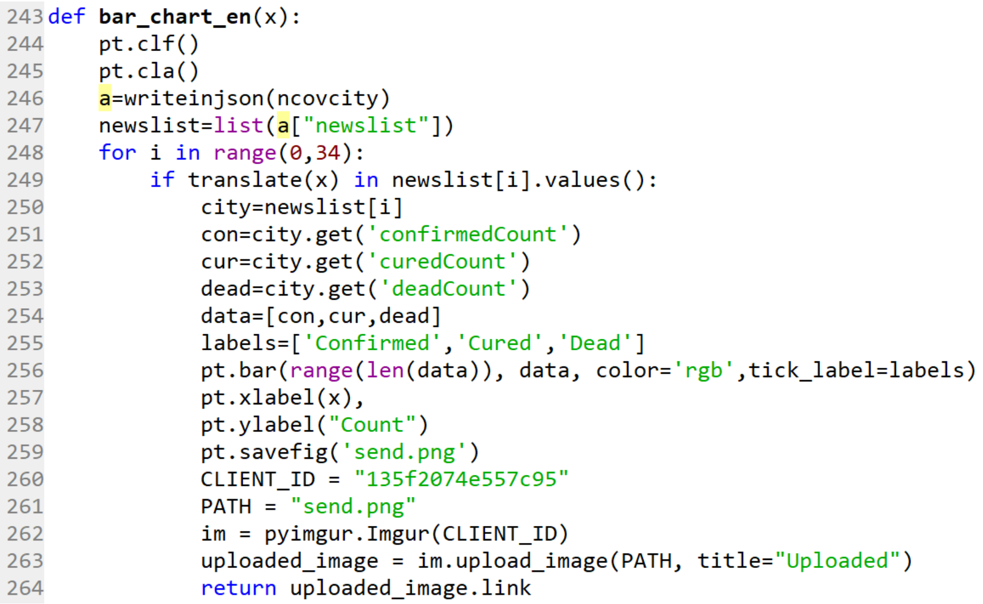


```json
{
  "data": {
    "id": "orunSTu",
    "title": null,
    "description": null,
    "datetime": 1495556889,
    "type": "image/gif",
    "animated": false,
    "width": 1,
    "height": 1,
    "size": 42,
    "views": 0,
    "bandwidth": 0,
    "vote": null,
    "favorite": false,
    "nsfw": null,
    "section": null,
    "account_url": null,
    "account_id": 0,
    "is_ad": false,
    "in_most_viral": false,
    "tags": [],
    "ad_type": 0,
    "ad_url": "",
    "in_gallery": false,
    "deletehash": "x70po4w7BVvSUzZ",
    "name": "",
    "link": "http://i.imgur.com/orunSTu.gif"
  },
  "success": true,
  "status": 200
}
```

From the example request, it contains a parameter called ‘clientId’. Actually, this id does not mean the account of imgur. Instead, it is the id of an application which can be registered from https://api.imgur.com/oauth2/addclient. After the register, we can get the ‘clientId’ and ‘secret’, which can upload pictures and video to personal server via them. We want to use this service to upload some matplotlib pictures and then reply in chart bot.

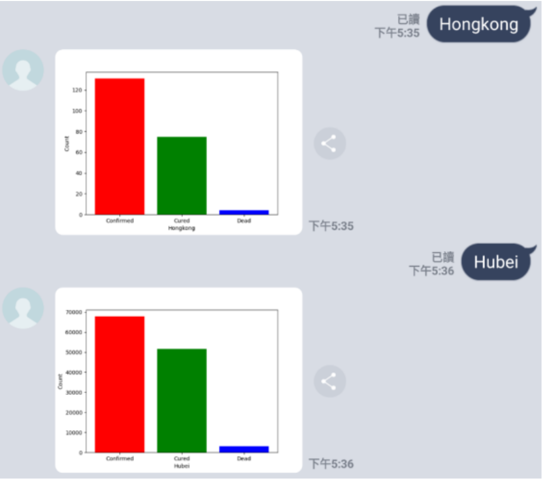
Specifically, we import PyImgur first. Then, we can define a function to get some image from matplotlib.pyplot, and then save them as png format, and then upload the image to imgur server as well as get the ‘link’(url) from the response. Finally, the line chart bot can reply image message. Base on above, we design that the ordinate of the bar chart is the number, and the abscissa has three bars of different colors, which are the number of infections, the number of cures, and the number of deaths. Finally, with the imgur server, our chart bot will reply bar chart image of infections number in an area when it receives message of area name.

Specific function code is as follows:



```
def bar_chart_en(x):
    pt.clf()
    pt.cla()
    a=writeinjson(ncovcity)
    newslist=list(a["newslist"])
    for i in range(0,34):
        if translate(x) in newslist[i].values():
            city=newslist[i]
            con=city.get('confirmedCount')
            cur=city.get('curedCount')
            dead=city.get('deadCount')
            data=[con,cur,dead]
            labels=['Confirmed','Cured','Dead']
            pt.bar(range(len(data)), data, color='rgb',tick_label=labels)
            pt.xlabel(x),
            pt.ylabel("Count")
            pt.savefig('send.png')
            CLIENT_ID = "135f2074e557c95"
            PATH = "send.png"
            im = pyimgur.Imgur(CLIENT_ID)
            uploaded_image = im.upload_image(PATH, title="Uploaded")
            return uploaded_image.link
```

Output:



已讀
下午5:35
Hongkong

下午5:35

已讀
下午5:36
Hubei

下午5:36