

Application Programming Interface (API) with Python Code

Track International Space Station(ISS) Location



Python ဖြင့် API အသုံးပြုပုံကို အဆင့်ဆင့် ရှင်းပြထားသည်။ data science project များအတွက် လိုအပ်သည့် data များ၊ live data များကို ဖတ်ယူ တောင်းယူသည့်နည်း ဖြစ်သည်။

API ကိုသုံးရန်အတွက် remote web server ထံမှ ဒေတာများ ရယူနိုင်ရန်အတွက် request လုပ်ရသည်။ မိမိက request သည့်အတိုင်း remote web server က ဒေတာများကို လှမ်းယူခွင့် (retrieve လုပ်ခွင့်) ပြုသည်။

အချိန်နှင့်အမျှ အလွန်လျှင်မြန်စွာ ပြောင်းလဲနေသည့် ဒေတာများကို စက္ကန့်တိုင်း ရယူနိုင်ရန်အတွက် API သည် အလွန်အသုံးဝင်သည်။ ဥပမာ stock price data, weather data

ထိုကဲ့သို့သော stock price data, weather data များကို မိနစ်တိုင်း download လုပ်ယူနေခြင်းသည် သိပ်အဆင်မပြေသည့်နည်း ဖြစ်သည်။ bandwidth များစွာ အသုံးပြုရသည်။ နှေးသည်။

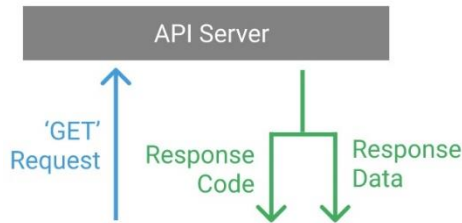
လက်ရှိ ရာသီဥတု အခြေအနေ အပူချိန်၊ လေတိုက်နှုန်းတို့ကို သိရန် dataset တစ်ခုလုံးကို download လုပ်ရန် မလိုပါ။ လက်ရှိ စတော့ဈေးနှုန်းကို သိရန်အတွက် dataset တစ်ခုလုံးကို download လုပ်ရန် မလိုပါ။ API ဖြင့် ဖတ်ယူခြင်း(data retrieve လုပ်ခြင်း)သည် အကောင်းဆုံး ဖြစ်သည်။

တခြားသော(external) system များ ဖြင့် interact လုပ်ရန်အတွက် API ကို အသုံးပြုသည်။ stock price data, weather data စသည့် live data များ ရယူရန်အတွက် API ကို အသုံးပြုသည်။

API ဆိုတာဘာလဲ?

API ဆိုသည်မှာ Application Programming Interface ဖြစ်သည်။ ဒေတာများ တောင်းခံရန် (request) သို့မဟုတ် ရယူရန်(retrieve)နှင့် ဒေတာများ ပေးပို့(send)ရန်အတွက် API ကို အသုံးပြုသည်။ ဒေတာများ တောင်းခံရန်(request) သို့မဟုတ် ရယူရန်(retrieve) အတွက် အလွန် အသုံးပြုကြသည်။

API ဖြင့် ဒေတာများ ရယူ(retrieve) ရန်အတွက် web server မှ request လုပ်ရသည်။



API ဖြင့် remote web server က ဒေတာများကို လှမ်းတောင်း(request လုပ်)သည့်အခါ API မှ သတ်မှတ်ထားသည့် စည်းမျဉ်း(rule)များအတိုင်း request လုပ် ရသည်။ သို့မှသာ external system သို့မဟုတ် remote web server က မိမိအလို ရှိသည့် ဒေတာများ ပြန်ပေးလိမ့်မည်။ စည်းမျဉ်း(rule)များနှင့် မကိုက်ညီသည့် တောင်းခံမှု(data request)များကို ငြင်းဆိုလိမ့်မည်။ မိမိအလိုရှိသည့် ဒေတာများ ရလိမ့်မည် မဟုတ်ပါ။

API Endpoint

API Endpoint ဆိုသည်မှာ ဒေတာရရှိနိုင်သည့် နေရာ သို့မဟုတ် ဒေတာသိမ်းဆည်းထားသည့် address ဖြစ်သည်။ တစ်နည်းအားဖြင့် ဒေတာပေးနိုင်သည့် url ဖြစ်သည်။ အချိန်တိုင်း စက္ကန့်မလပ် ဒေတာများ ပေးနိုင်တဲ့ web address, API Endpoint ဟု ဆိုနိုင်သည်။

[https:// api.coinbase.com](https://api.coinbase.com)

<https://www.astroviewer.net/iss/en/>

<http://open-notify.org/Open-Notify-API/ISS-Location-Now/>

https://spotthestation.nasa.gov/tracking_map.cfm

ရိုးရှင်းလွယ်ကူသည့် api တစ်ခုကို နမူနာ အဖြစ် စမ်းကြည့်ရအောင်။ International Space Station(ISS) ၏ လက်ရှိ တည်ရှိရာနေရာ(current location)ကို ပေးသည့် API ကို ဥပမာအဖြစ် စမ်းကြည့်မည်။ space station ၏ (current latitude and longitude) လောင်ဂျီဒ် နှင့် လတ္တီဒ်ကို return လုပ်မည်။ ထိုသို့ တည်ရှိနေရာကို ရယူရန် အတွက် မည်သည့် input မှ ထည့်ပေးရန်မလိုပါ။ လက်ရှိ အချိန် (now) အတွက်သာ ဖြစ်သောကြောင့် မည်သည့် input မှ ထည့်ပေးရန် မလိုပါ။

မြို့တစ်မြို့၏ အပူချိန်ဒေတာကို ရယူရန်အတွက် city name ကို input အဖြစ် ထည့်ပေးရသည်။

Output သည် JSON format ဖြစ်သည်။

<http://api.open-notify.org/iss-now.json>

အပေါ်မှ API endpoint ကို browser address bar တွင် ထည့်လျှင်အောက်ပါ ဒေတာများကို ပြန်ပေးလိမ့်မည်။ အချိန်နှင့် တည်ရှိရာနေရာ ကို ပြန်ပေးသည်။

```
{
  "timestamp": 1611379778,
  "message": "success",
  "iss_position": {
    "longitude": "50.3409",
    "latitude": "-51.4813"
  }
}
```

Making API Requests in Python

Python code များ ဖြင့် API ကို အသုံးပြုရန်အတွက် requests လုပ်နိုင်သည့် requests library ကို install လုပ်ထားရန် လိုအပ်သည်။ requests library သည် standard Python library တစ်ခု မဟုတ်ပါ။ ထို့ကြောင့် install လုပ်ပေးရန်လိုအပ်သည်။

<https://pypi.org/project/requests/> တွင် နှင့် ပတ်သက်သည့်အချက်အလက်များကို ဖတ်နိုင်သည်။

Command လိုင်းတွင်

```
pip install requests
```

Conda အတွက်

```
conda install requests
```

library ကို instal လုပ်ပြီးပါက import လုပ်ပြီး အသုံးပြုနိုင်ပြီ ဖြစ်သည်။

```
import requests
```

ISS ကျွန်ုပ်တို့ရဲ့ကဦးခေါင်းပေါ်ကနေ ကမ္ဘာကို ပတ်နေသည်။ ISS location အချိန်တိုင်း စက္ကန့်မလပ်ပေးနိုင်သည့် web address သို့မဟုတ် API endpoint မှ ရယူနိုင်သည်။ ထို API endpoint မှ တစ်ဆင့် တည်ရှိနေသည့် တည်နေရာ လောင်ဂျီဒ် နှင့် လတ္တီဒ်ကို ဖတ်ယူသည်။ ထိုသို့ ဖတ်ယူနိုင်ရန်အတွက် requests module ကို အသုံးပြုသည်။

Making Our First API Request

Data တောင်းယူရန်(retrieve လုပ်)ရန် requests လုပ်နိုင်သည့် နည်းများစွာရှိသည့် အနက် အသုံးအများဆုံး နည်းမှာ GET request ဖြစ်သည်။

GET request လုပ်လိုက်သည့်အခါ ထိုတောင်းခံမှုအောင်မြင်သည် မအောင်မြင်သည်ကို ဖော်ပြထားသည့် response code ကို ပြန်ပေးသည်။ ချက်ချင်း သိနိုင်သောကြောင့် response code များသည့် အဆင်ပြေသည့်နည်းလမ်းတစ်ခု ဖြစ်သည်။

requests.get() function ကို အသုံးပြုသည်။

() ထဲတွင် URL ကို argument အဖြစ် ထည့်ပေးရသည်။
ပထမဦးစွာ API endpoint သည် တကယ်ရှိ မရှိ စစ်ဆေးကြည့်မည်။

```
requests.get(url = "http://api.open-notify.org/iss-now.json")
```

<Response [200]>

```
requests.get("http://api.open-notify.org/iss-now.json")
```

<Response [200]>

<Response [200]> ဟုသာ ပြန်ပေးသည်။ ဒေတာများကို မဖော်ပြသေးပါ။ GET request လုပ်ခြင်း
အောင်မြင်မှုရှိ မရှိကို ဖော်ပြသည့် response code ဖြစ်သည်။ <Response [200]> သည် အောင်မြင်သည်။

<Response [404]> သည် ထို url မရှိပါ။ (doesn't exist)

- 1xx:** Hold, something happening. (တစ်ခုခု ဖြစ်နေပြီ ။ ခဏစောင့်ပါ။)
- 2xx:** Here you go. (အဆင်ပြေပါတယ်။ ဒီမှာ သင့်တောင်းဆိုထားတာတွေ ဒီမှာ ရပါပြီ။)
- 3xx:** You don't have permission. Go away. (သင့်တောင်းဆိုမှုကို ခွင့် မပြုနိုင်ပါ။ သွားပါတော့)
- 4xx:** You screwed up. (အမှားတစ်ခုခု ဖြစ်ပေါ်နေပါသည်။)
- 5xx:** I screwed up. (ငါ ပြဿနာတက်နေတယ်။ server down, web site down.)

status_code ဖြင့်လည်း ဖတ်ယူနိုင်သည်။

```
response = requests.get("http://api.open-notify.org/iss-now.json")
print(response.status_code)
```

200

<http://api.open-notify.org/iss-now.json> တွင် အမှားဖြစ်အောင် စာလုံးတစ်လုံးလုံ ဖြုတ်ထားလိုက်လျှင်

<Response [404]> ရရှိလိမ့်မည်။

<Response [200]> ပြန်မရပါက exception ထုတ်ပေးမည့် code ထည့်ရေးနိုင်သည်။

```
if response.status_code != 200:
    raise Exception("Bad response from ISS API")
```

အောက်က link တွင် HTTP Status Codes ကို လေ့လာနိုင်သည်။

<https://httpstatuses.com/>

- **1xx Informational**
- **100** Continue
- **101** Switching Protocols

- [102 Processing](#)
- **2xx Success**
- [200 OK](#)
- [201 Created](#)
- [202 Accepted](#)
- [203 Non-authoritative Information](#)
- [204 No Content](#)
- [205 Reset Content](#)
- [206 Partial Content](#)
- [207 Multi-Status](#)
- [208 Already Reported](#)
- [226 IM Used](#)

`raise_for_status()` function ကို သုံး၍ အထက်တွင် ဖော်ပြထားသည့် HTTP Status Code များကို ရယူနိုင်သည်။

```
response.raise_for_status()
```

Builtin JSON decoder

return ပြန်လာသည့် ဒေတာများသည် JSON data format ဖြစ်သောကြောင့် builtin JSON decoder ကို အသုံးပြု၍ ဒေတာများကို ပြန်ဖြန့်(decode လုပ်)မည်။ တစ်နည်းအားဖြင့် JSON format data များကို ပြန်ဖြည့်မည်။ ထို့နောက် data variable ဖြင့် assign လုပ်မည်။

```
data = response.json()
print(data)
```

```
{'timestamp': 1611383216, 'message': 'success', 'iss_position': {'longitude': '-91.2292', 'latitude': '33.1826'}}
```

ဒေတာများ အားလုံးမဟုတ်ဘဲ မိမိအလိုရှိသည့် တစ်မျိုးတည်းကိုလည်း ထုတ်ယူနိုင်သည်။

```
data = response.json()['iss_position']
print(data)
```

```
{'longitude': '-84.7582', 'latitude': '27.3370'}
```

အောက်ပါအတိုင်း လောင်ဂျီတွဒ် တန်ဖိုး တစ်ခုတည်းကိုသာ ထုတ်ယူနိုင်သည်။

```
data = response.json()['iss_position']['longitude']
print(data)
```

```
-81.2369
```

longitude နှင့် latitude တို့ကို ထုတ်ယူပြီး tuple အဖြစ် ပြုလုပ်သည်။ longitude နှင့် latitude တို့သာ ပါသည့် ဒေတာအဖြစ် ပြင်ယူသည်။

```
data = response.json()
longitude = data ['iss_position']['longitude']
latitude = data ['iss_position']['latitude']
iss_current_position=(longitude,latitude)
print(iss_current_position)
```

```
('-69.8400', '9.6055')
```

ထိုတန်ဖိုးနှစ်ခုသည် ကမ္ဘာမြေပုံပေါ်တွင် မည်သည့်နေရာတွင် တည်ရှိနေသည်ကို ဖော်ပြရန်အတွက် <https://www.latlong.net/Show-Latitude-Longitude.html> တွင် သွားကြည့်နိုင်သည်။

အောက်တွင် ကုဒ်များ အားလုံးပေါင်း၍ ပြန်ဖော်ပြထားသည်။

```
import requests

response = requests.get("http://api.open-notify.org/iss-now.json")

data = response.json()['iss_position']
print(data)

data = response.json()
longitude = data ['iss_position']['longitude']
latitude = data ['iss_position']['latitude']
iss_current_position=(longitude,latitude)
print(iss_current_position)
```

```
{'longitude': '102.1012', 'latitude': '-4.4208'}
('102.1012', '-4.4208')
```

International Space Station (ISS) ၏ တည်နေရာ(location) ကို track လုပ်လိုပါက အောက်ပါအတိုင်း ပြင်ရေးနိုင်သည်။ အချိန် နှင့်တွဲရေးရန် လက်ရှိအချိန်ကို ဖတ်ရန်အတွက်

```
from datetime import datetime
time_now = datetime.now()
```

(၁၀) စက္ကန့်တိုင်း နေရာကို ဖော်ပြရန်အတွက် while loop နှင့် တွဲသုံးမည်။ (၁၀) စက္ကန့် sleep လုပ်မည်။

```
import time
time.sleep(10)
```

```
import requests
from datetime import datetime
import time
```

```

while True:
    response = requests.get("http://api.open-notify.org/iss-
                             now.json")
    time.sleep(10)
    data = response.json()
    longitude = data['iss_position']['longitude']
    latitude = data['iss_position']['latitude']
    iss_current_position = (longitude, latitude)
    # print(iss_current_position)
    time_now = datetime.now()
    print(time_now, iss_current_position)

```

```

2021-01-23 17:02:34.391739 ('91.8689', '14.0375')
2021-01-23 17:02:44.937203 ('92.2739', '14.5595')
2021-01-23 17:02:55.375256 ('92.6809', '15.0806')
2021-01-23 17:03:05.826481 ('93.0902', '15.6010')
2021-01-23 17:03:16.271343 ('93.5018', '16.1203')
2021-01-23 17:03:26.717033 ('93.9157', '16.6387')
2021-01-23 17:03:37.161437 ('94.3123', '17.1314')
2021-01-23 17:03:47.709735 ('94.7313', '17.6478')
2021-01-23 17:03:58.154701 ('95.1529', '18.1631')
2021-01-23 17:04:08.599894 ('95.5772', '18.6773')
2021-01-23 17:04:19.044464 ('96.0045', '19.1903')
2021-01-23 17:04:29.476898 ('96.4346', '19.7021')
2021-01-23 17:04:39.932977 ('96.8679', '20.2127')

```

API parameter

API အားလုံးအတွက် Parameter မလိုအပ်ပါ။ ဥပမာ- sunset sunrise time API အတွက်

<https://sunrise-sunset.org/api>

API documentation

API documentation တွင် မည်ကဲ့သို့ structure လုပ်ရမည်ကို အတိအကျ ဖော်ပြထားသည်။

GET request ရန်အတွက် API endpoint မှာ <https://api.sunrise-sunset.org/json> ဖြစ်သည်။

Parameters

lat (float): Latitude in decimal degrees. **Required.**

lng (float): Longitude in decimal degrees. **Required.**

date (string): Date in YYYY-MM-DD format. Also accepts other date formats and even relative date formats. If not present, date defaults to current date. Optional.

callback (string): Callback function name for JSONP response. Optional.

formatted (integer): 0 or 1 (1 is default). Time values in response will be expressed following ISO 8601 and day_length will be expressed in seconds. Optional.

lat (float) နှင့် lng (float) ကို မဖြစ်မနေထည့်ပေးရန် လိုအပ်သည်။ ကျန် parameter များမှာ Optional ဖြစ်သည်။

နေထွက်ချိန်နှင့် နေဝင်ချိန်ကို ဖတ်ယူခြင်း

API မှ ပြန်ပေးသည့် အချိန်များ အားလုံးသည် UTC(Coordinated Universal Time) ဖြစ်သည်။ Myanmar(Burma) စံတော်ချိန်သည် Coordinated Universal Time ထက် ၆နာရီခွဲ ပိုစောသည်။ (Myanmar (Burma) is 6 hours and 30 minutes ahead of Coordinated Universal Time.)

Myanmar (Burma) (GMT+6:30)

```
import requests

# Yangon ၏ လောင်ဂျီဒ် နှင့် လတ္တီဒ်ကို https://www.latlong.net မှ ဖတ်ယူသည်။
MY_LAT = 16.866070
MY_LNG = 96.195129

parameters = {
    "lat":MY_LAT,
    "lng":MY_LNG,
    "formatted": 0,
}

response = requests.get(url="https://api.sunrise-sunset.org/json", params
= parameters)
response.raise_for_status()
data=response.json()
print(data)
```

```
{'results': {
    'sunrise': '2021-01-23T00:07:41+00:00',
    'sunset': '2021-01-23T11:26:38+00:00',
    'solar_noon': '2021-01-23T05:47:10+00:00',
    'day_length': 40737,
    'civil_twilight_begin': '2021-01-22T23:44:56+00:00',
    'civil_twilight_end': '2021-01-23T11:49:24+00:00',
    'nautical_twilight_begin': '2021-01-22T23:18:38+00:00',
    'nautical_twilight_end': '2021-01-23T12:15:42+00:00',
    'astronomical_twilight_begin': '2021-01-22T22:52:32+00:00',
    'astronomical_twilight_end': '2021-01-23T12:41:47+00:00'
},
'status': 'OK'}
```



```
sunrise = data["results"]["sunrise"]  
print(sunrise)
```

2021-01-23T00:07:41+00:00

UTC အချိန်မှ မြန်မာစံတော်ချိန်သို့ပြောင်းရန် (၆)နာရီခွဲ နှုတ်ရမည်။ 05:37:41 နံနက် (၅)နာရီ (၃၇) မိနစ်သည် နေထွက်ချိန် ဖြစ်သည်။

```
sunset = data["results"]["sunset"]  
print(sunset)
```

2021-01-23T11:26:38+00:00

11:26:38သည် UTC ဖြစ်သည်။ နေဝင်ချိန်သည် မြန်မာစံတော်ချိန် ညနေ (၅)နာရီ (၅၆)မိနစ် ဖြစ်သည်။(05:56:41)

"formatted" ကို 1 ဟုပြောင်းလိုက်လျှင် AM နှင့် PM format ပြောင်းပေးလိမ့်မည်။

```
parameters = {  
    "lat":MY_LAT,  
    "lng":MY_LNG,  
    "formatted": 1,  
}
```

12:07:41 AM

11:26:38 AM

ISS သည် မိမိ နေထိုင်ရာ အနီး latitude နှင့် longitude +5 or -5 အတွင်းရောက်လာပါ။ ညအချိန်ဖြစ်ပါက မိမိ အသိပေးလိုသူထံသို့ email ပို့ပေးသည့် python code ဖြစ်သည်။

အပိုင်း(၃)ပိုင်းရှိသည်။

(၁) ISS တည်နေရာကို API ဖြင့် ရယူရန် (မိမိတည်နေရာ ၏ latitude နှင့် longitude သိထားရမည်)

(၂) မိမိ တည်နေရာ၏ နေထွက်ချိန်နှင့် နေဝင်ချိန်ကို API ဖြင့် ရယူရန်

(၃) မိမိ email account မှ မိမိသိစေလိုသူ ထံသို့ email ပို့ပေးရန်

gmail account မှာ python code ဖြင့် email ပို့နည်းကို အောက်က link တွေ ဖော်ပြပြီး ဖြစ်သည်။

http://www.acmv.org/python/send_email/smtp_test.py.html

```
import requests  
from datetime import datetime  
import smtplib  
import time  
  
MY_EMAIL = "____YOUR_EMAIL_HERE____"  
MY_PASSWORD = "____YOUR_PASSWORD_HERE____"  
MY_LAT = 51.507351 # Your latitude
```

```

MY_LONG = -0.127758 # Your longitude

def is_iss_overhead():
    response = requests.get(url="http://api.open-notify.org/iss-
now.json")
    response.raise_for_status()
    data = response.json()

    iss_latitude = float(data["iss_position"]["latitude"])
    iss_longitude = float(data["iss_position"]["longitude"])

    #Your position is within +5 or -5 degrees of the iss position.
    if MY_LAT-5 <= iss_latitude <= MY_LAT+5 and MY_LONG-5 <=
iss_longitude <= MY_LONG+5:
        return True

def is_night():
    parameters = {
        "lat": MY_LAT,
        "lng": MY_LONG,
        "formatted": 0,
    }
    response = requests.get("https://api.sunrise-sunset.org/json",
params=parameters)
    response.raise_for_status()
    data = response.json()
    sunrise =
int(data["results"]["sunrise"].split("T")[1].split(":")[0])
    sunset = int(data["results"]["sunset"].split("T")[1].split(":")[0])

    time_now = datetime.now().hour

    if time_now >= sunset or time_now <= sunrise:
        return True

while True:
    time.sleep(60)
    if is_iss_overhead() and is_night():
        connection = smtplib.SMTP("__YOUR_SMTP_ADDRESS_HERE__")
        connection.starttls()
        connection.login(MY_EMAIL, MY_PASSWORD)
        connection.sendmail(
            from_addr=MY_EMAIL,
            to_addrs=MY_EMAIL,
            msg="Subject:Look Up 🙌\n\nThe ISS is above you in the sky."
        )

```

Reference sites:

100 Days of Code - The Complete Python Pro Bootcamp for 2021 by Dr. Angela Yu

<https://wheretheiss.at/w/developer>

<https://www.dataquest.io/blog/python-api-tutorial/>

<http://open-notify.org/Open-Notify-API/ISS-Location-Now/>

ဆက်လက်လေ့လာရန်

<https://www.dataquest.io/blog/last-fm-api-python/>