Day - 33

**API**

API, API-applications, requests

**33.1 API**

An Application Programming Interface (API) is a set of commands, functions, protocols, and objects that programmers can use to create software or interact with an external system.

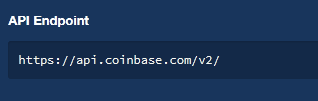
**33.2 Get Data from API (interact with an external system)**

API is like an interface to the External data and Application (that uses that data). Different API servers require different rules. To grab the data from the API, we need to follow those rules. Do following:

First read the documentation of the API server. Connect to that API from our app, and send request to the API server, in return we get our data.

**33.3 API endpoint**

Is actually a link, where we send our request. For example:



curl https://api.coinbase.com/v2/user \

* ISS Current Location:

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

**33.4 Fetching URLs in Python:**

* Get-request& response:

**import** requests

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

**print**(resPonse)

#*python api\_first\_demo.py*

* Response code: There are different Response code we get from a server:

|  |  |  |  |
| --- | --- | --- | --- |
| 1×× Informational      100 Continue      101 Switching Protocols      102 Processing      2×× 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 | 3×× Redirection      300 Multiple Choices      301 Moved Permanently      302 Found      303 See Other      304 Not Modified      305 Use Proxy      307 Temporary Redirect      308 Permanent Redirect | 4×× Client Error      400 Bad Request      401 Unauthorized      402 Payment Required      403 Forbidden      404 Not Found      405 Method Not Allowed      406 Not Acceptable      407 Proxy Authentication Required      408 Request Timeout      409 Conflict      410 Gone      411 Length Required      412 Precondition Failed      413 Payload Too Large      414 Request-URI Too Long      415 Unsupported Media Type      416 Requested Range Not Satisfiable      417 Expectation Failed      418 I'm a teapot      421 Misdirected Request      422 Unprocessable Entity      423 Locked      424 Failed Dependency      426 Upgrade Required      428 Precondition Required      429 Too Many Requests      431 Request Header Fields Too Large      444 Connection Closed Without Response      451 Unavailable For Legal Reasons      499 Client Closed Request | 5×× Server Error      500 Internal Server Error      501 Not Implemented      502 Bad Gateway      503 Service Unavailable      504 Gateway Timeout      505 HTTP Version Not Supported      506 Variant Also Negotiates      507 Insufficient Storage      508 Loop Detected      510 Not Extended      511 Network Authentication Required      599 Network Connect Timeout Error |

Instead checking those code one by one for exception, we use simply ***requests***,

* Requests: Requests is an elegant and simple HTTP library for Python, built for human beings.

***pip install requests***

**import** requests

**import** urllib3

* Errors and Exceptions: In the event of a network problem (e.g. DNS failure, refused connection, etc), Requests will raise a ConnectionError exception.

*Response.raise\_for\_status()*

will raise an *HTTPError* if the *HTTP* request returned an *unsuccessful* *status* *code*.

* If a request times out, a ***Timeout*** exception is raised. If a request exceeds the configured number of maximum redirections, a ***TooManyRedirects*** exception is raised. All exceptions that Requests explicitly raises inherit from *requests.exceptions.RequestException*.
* Specify exception:

**import** requests

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

**print**(resPonse)

only\_status\_code = resPonse**.**status\_code

**print**(only\_status\_code)

#*specify exception*

**if** resPonse**.**status\_code **==** 404:

**raise** **Exception**("The resource not exists")

**elif** resPonse**.**status\_code **==** 401:

**raise** **Exception**("You have no Access")

Or we can simply use:

**import** requests

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

**print**(resPonse)

only\_status\_code = resPonse**.**status\_code

**print**(only\_status\_code)

#*specify exception*

resPonse**.raise\_for\_status**()

**33.5 Accessing data from API response**

**import** requests

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

**print**(resPonse)

only\_status\_code = resPonse**.**status\_code

**print**(only\_status\_code)

#*specify exception*

resPonse**.raise\_for\_status**()

#*Accessing the data: Same way we are used to retrive from JSON*

data = resPonse**.json**()

**print**(data)

**print**(data["iss\_position"]["longitude"])

**print**(data["iss\_position"]["latitude"])

coor\_Dnitate = (data["iss\_position"]["longitude"], data["iss\_position"]["latitude"])

* Exercise 33.1: Use API for ***KANYE*** get his quotes and apply it to the given app-code. Use *kanye.rest*.

**from** tkinter **import** \*

**import** requests

**def** **get\_quote**():

    #*Write your code here.*

    reQest = requests**.get**(url = "https://api.kanye.rest")

    reQest**.raise\_for\_status**()

    data = reQest**.json**()

**print**(data["quote"])

    canvas**.itemconfig**(quote\_text, text = data["quote"])

window = **Tk**()

window**.title**("Kanye Says...")

window**.config**(padx=50, pady=50)

canvas = **Canvas**(width=300, height=414)

background\_img = **PhotoImage**(file="background\_2.png")

canvas**.create\_image**(150, 207, image=background\_img)

quote\_text = canvas**.create\_text**(150, 207, text="Kanye Quote Goes HERE", width=250, font=("Arial", 20, "bold"), fill="#be2edd")

canvas**.grid**(row=0, column=0)

kanye\_img = **PhotoImage**(file="kanye\_2.png")

kanye\_button = **Button**(image=kanye\_img, highlightthickness=0, command=get\_quote, bd = 0)

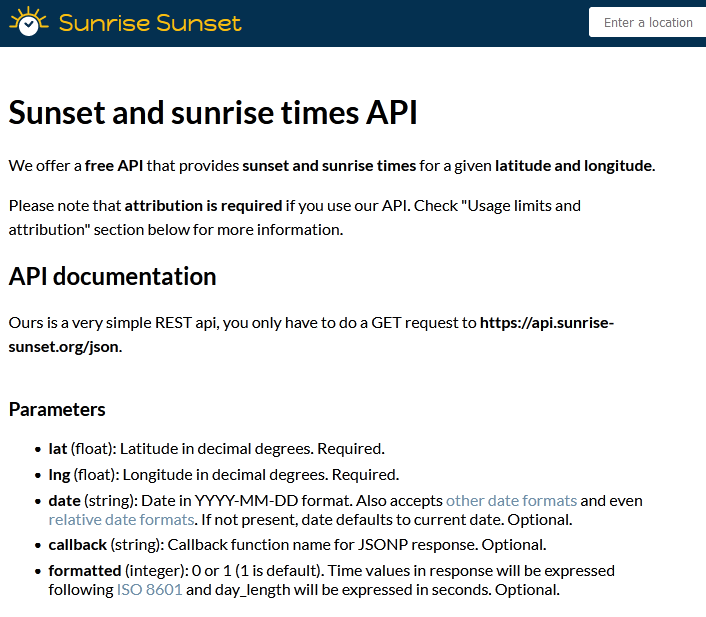
kanye\_button**.grid**(row=1, column=0)

window**.mainloop**()

#*python kanye\_main.py*

**33.6 API parameters**

* API parameters: Some parameters are default and some are required.
* We first create ***Dictionary*** using the ***parameters***.
* Then place the ***Dictionary*** in the ***get-request***.



* Parameters in URL string:

https://api.sunrise-sunset.org/json?lat=36.7201600&lng=-4.4203400

|  |
| --- |
|  |

* Parameters in Python get-request:

poStn = {

    "lat" : My\_Lat,

    "lng" : My\_long,

    "formatted" :0

}

sun\_resp = requests**.get**(url = "https://api.sunrise-sunset.org/json", params= poStn)

* Exercise 33.2: Splitting Sunrise and Sunset hour. Use **split()** method.

**import** requests

**import** datetime **as** dTm

#*Moscow is used*

My\_Lat = 55.7558

My\_long = 37.6173

#*this Dictionary needs to contain exact keya wich are the params of API*

#*"formatted" :0, Used for 24-hour format*

poStn = {

    "lat" : My\_Lat,

    "lng" : My\_long,

    "formatted" :0

}

sun\_resp = requests**.get**(url = "https://api.sunrise-sunset.org/json", params= poStn)

sunrise\_data = sun\_resp**.json**()

sun\_rise\_row = sunrise\_data["results"]["sunrise"]

sun\_set\_row = sunrise\_data["results"]["sunset"]

#*splitting and grabbing the hour only from -- 2022-01-22T05:37:50+00:00,*

#*seperating "05"*

sun\_rise\_hour = sun\_rise\_row**.split**("T")[1]**.split**(":")[0]

sun\_set\_hour = sun\_set\_row**.split**("T")[1]**.split**(":")[0]

#*acessing PC current hour. Above is done for comare to following*

pc\_hour = dTm**.**datetime**.now**()**.**hour

**print**(f"sun\_rise\_hour :{sun\_rise\_hour},  sun\_set\_hour :{sun\_set\_hour}, And pc\_hour : {pc\_hour}")

#*python api\_params.py*

* Exercise 33.3: Download the starting code for the final project and do following.
* # If ISS is close to my position
* # and it is currently dark
* # Then send me an email to tell me to look
* # BONUS: run the code every 60 seconds.

Practiced version

**import** requests

**from** datetime **import** datetime

**import** smtplib

**import** time

My\_email = ""

My\_pw = ""

MY\_LAT = 23.835852 #*Your latitude*

MY\_LONG = 90.252404 #*Your longitude*

parameters = {

    "lat": MY\_LAT,

    "lng": MY\_LONG,

    "formatted": 0,

}

#*If the ISS is close to my current position*

**def** **iss\_close**():

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

    response**.raise\_for\_status**()

    data = response**.json**()

**print**(data)

    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.*

    lat\_err = iss\_latitude - MY\_LAT

    lng\_err = iss\_longitude - MY\_LONG

**if** (-5.0 **<** lat\_err **<** 5.0) **and** (-5.0 **<** lng\_err **<** 5.0):

**print**("ISS is close")

**return** **True**

**else**:

**print**("ISS is not Nearby")

**return** **False**

#*and it is currently dark*

**def** **is\_dark**():

    response = requests**.get**("https://api.sunrise-sunset.org/json", params=parameters)

    response**.raise\_for\_status**()

    data = response**.json**()

**print**(data)

    #*Used GMT location +6*

    sunrise = **int**(data["results"]["sunrise"]**.split**("T")[1]**.split**(":")[0]) + 6

    sunset = **int**(data["results"]["sunset"]**.split**("T")[1]**.split**(":")[0]) + 6

**if** (time\_now **>** sunset) **or** (time\_now **<** sunrise) :

**print**("this is night")

**return** **True**

**else**:

**return** **False**

#*Then send me an email to tell me to look up.*

#*Order matters: if one is false rest is not executed*

#*"if (is\_dark() and iss\_close()):" or "if (iss\_close() and is\_dark()):"*

**while** **True**:

    time**.sleep**(10)

    time\_now = datetime**.now**()**.**hour

**if** (**is\_dark**() **and** **iss\_close**()):

**with** smtplib**.SMTP**("smtp.gmail.com") **as** sender:

            sender**.starttls**()

            sender**.login**(user= My\_email, password= My\_pw)

            sender**.sendmail**(from\_addr= My\_email,

                            to\_addrs= My\_email,

                            msg = "Subject: Int. Space Station Aleart !!!\n\n Yo!! Look at the sky Int. Space Station is nearby")

#*BONUS: run the code every 60 seconds.*

#*python iss\_overhead.py*

Instructors Solution

**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."

        )