Day - 35

**API authentication**

API Keys, Authentication, Environment Variables and Sending SMS

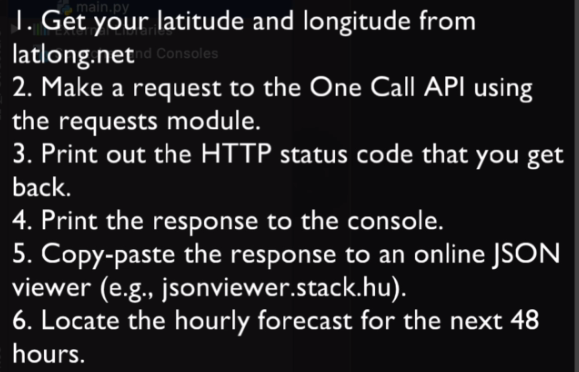
**35.1 Today's Goals**

* API Keys, Authentication: More advanced way of using APIs, Authenticate ourselves with API-provider
* Sending SMS: Use ***twilio*** and similar thing.
* Environment Variables: Safely storing API-Keys
* Build a Rain-Alert application.

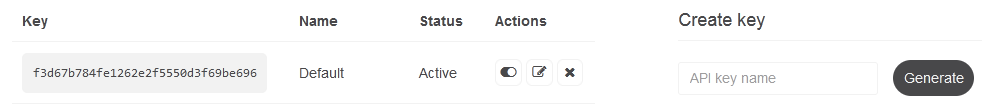
**35.2 API authentication**

* Why do people charge for API anyways?: For example consider a weather API, API-provider has to open weather map, they have access to over 4,000 weather stations across the globe, and then their data scientists will take that weather data and look at the satellite images and process that data in order to figure out the weather and predict the weather for each city in the world.
* As you can imagine, that's going to be really resource-intensive and it's going to require a lot of employees and also server maintenance and electricity costs. So for all of that hassle, if they're going to provide this data for you, then it's reasonable that you should be paying for it.
* API-key: Most of commercial APIs are free for learning. But they need authentication via API-key (like personal AC.No/Password). To prevent their data to be used commercially without payment. By API-key they can track the usage and access to the data.
* Exercise 35.1: Authenticate an Weather API.
* Login / Register to *https://openweathermap.org*
* Go to *https://openweathermap.org/api/one-call-api*

|  |
| --- |
|  |



* Generate Key:



**import** requests

my\_lat = 23.810331

my\_lan = 90.412521

parAm = {

    "lat" : my\_lat,

    "lon" : my\_lan,

    "exclude" : "current, minutely, daily",

    "appid" : "----------"

}

dt\_req = requests**.get**(url = "https://api.openweathermap.org/data/2.5/onecall", params= parAm)

dt\_req**.raise\_for\_status**()

data = dt\_req**.json**()

**print**(data)

#*python authn\_api\_demo.py*

* Exercise 35.2: Accessing the weather data.

Practiced Version

**import** requests

**import** json

my\_lat = 23.810331

my\_lan = 90.412521

parAm = {

    "lat" : my\_lat,

    "lon" : my\_lan,

    #*ignore/exclude the other data. No space used only ","*

    "exclude" : "current,minutely,daily",

    "appid" : "--------"

}

dt\_req = requests**.get**(url = "https://api.openweathermap.org/data/2.5/onecall", params= parAm)

dt\_req**.raise\_for\_status**()

daTa = dt\_req**.json**()

**print**(daTa)

#*save Data to a .json file*

**with** **open**("weather\_data.json", "w") **as** data\_file:

    #*Saving the new data. Notice "json module" is used*

    json**.dump**(daTa, data\_file, indent=4)

#*read from Json*

**with** **open**("weather\_data.json") **as** read\_file:

    json\_data = json**.load**(read\_file)

bring\_umbrla = **False**

**for** i **in** **range**(0, 12):

    ids = json\_data["hourly"][i]["weather"][0]["id"]

**print**(ids)

**if** 500 **<=** **int**(ids) **<** 532:

        bring\_umbrla = **True**

**if** bring\_umbrla:

**print**("Bring Umbrella")

#*python authn\_api\_demo.py*

Instructors version

#*read from Json*

**with** **open**("weather\_data.json") **as** read\_file:

    json\_data = json**.load**(read\_file)

bring\_umbrla = **False**

#*List-slicing: Lst[ Initial : End : IndexJump ]*

sliced\_hour = json\_data["hourly"][0 : 12 : 1]

**print**(sliced\_hour)

**for** hUr **in** sliced\_hour:

**print**(hUr)

    ids = hUr["weather"][0]["id"]

**print**(ids)

**if** 500 **<=** **int**(ids) **<** 532:

        bring\_umbrla = **True**

**if** bring\_umbrla:

**print**("Bring Umbrella")

**35.3 Python List Slicing**

Python List Slicing: In Python, list slicing is a common practice and it is the most used technique for programmers to solve efficient problems.

Syntax:

Lst**[** Initial **:** End **:** IndexJump **]**

If ***Lst*** is a list, then the above expression returns the portion of the list from index Initial to index ***End***, at a step size ***IndexJump***.

|  |  |
| --- | --- |
| Positive index | Negative index |
|  |  |

**35.4 Sending SMS: *pythonanywhere* and *twilio***

Twilio: Create account, verify, answer questions.

Create a phone number. Get a trial number.

Docs quickstarts SMS

pip install twilio

**35.5 Environment Variables: Hide Passwords**

* Convenience: Normally when you deploy a ***large application, the process is quite complicated***. And once you've done it, you kind of don't want to mess around with the code base and update the code files like your ***main.py*** for example.
* Instead, you could have these environment variables, which you can change. For example, if you had an application that was sending you emails out to your clients, then your client base emails might change day to day. So certain variables that are being used in your code base could be set as environment variables and you can modify those variables without having to touch the code.
* Security: Used to hide ***key/password/personal\_info*** from ***public***. Useful for ***github***, drop-box or other open places, where your code is visible.

All code Before TWILIO

**import** requests

**import** json

my\_lat = 23.810331

my\_lan = 90.412521

parAm = {

    "lat" : my\_lat,

    "lon" : my\_lan,

    #*ignore/exclude the other data. No space used only ","*

    "exclude" : "current,minutely,daily",

    "appid" : "-----"

}

dt\_req = requests**.get**(url = "https://api.openweathermap.org/data/2.5/onecall", params= parAm)

dt\_req**.raise\_for\_status**()

daTa = dt\_req**.json**()

#*print(daTa)*

#*save Data to a .json file*

#*with open("weather\_data.json", "w") as data\_file:*

#*# Saving the new data. Notice "json module" is used*

#*json.dump(daTa, data\_file, indent=4)*

#*# read from Json*

#*with open("weather\_data.json") as read\_file:*

#*json\_data = json.load(read\_file)*

#*# List-slicing: Lst[ Initial : End : IndexJump ]*

#*sliced\_hour = json\_data["hourly"][0 : 12 : 1]*

bring\_umbrla = **False**

sliced\_hour = daTa["hourly"][0 : 12 : 1]

**for** hUr **in** sliced\_hour:

**print**(hUr)

    ids = hUr["weather"][0]["id"]

**print**(ids)

**if** 500 **<=** **int**(ids) **<** 532:

        bring\_umbrla = **True**

**if** bring\_umbrla:

**print**("Bring Umbrella")

#*python authn\_api\_demo.py*