**Installation and usage instructions**

So firstly, to do this project we need to install some software tools in CMD which are as follows:

1. In first step we need to install python by using the below link:

<https://realpython.com/installing-python/>

1. Once you done with installation, we need to install pip to download other dependencies’ using the link below:

<https://phoenixnap.com/kb/install-pip-windows>

1. In this step we need to install request package for authentication and requests of the Tickets using simple command in CMD:

**pip install requests**

1. Before we are executing code to request tickets, we need to create Zendesk account where you will get API keys, API domain and password using the following link:

<https://www.zendesk.com/register/#step-1>

1. Now fill this detail in config.py file for e.g.:

API\_LIST\_ENDPOINT="https://abc.zendesk.com/api/v2/tickets.json"

API\_TICKET\_ENDPOINT="https://abc.zendesk.com/api/v2/tickets/"

USER="xyz@gmail.com"

PASSWORD="xyz@18"

1. So, now we import some essential libraries such as requests, json and defaultdict for the sending the tickets.
2. Now we are ready to connect with Zendesk API by following line of code:

# Trying to reach the API using username and password

req=re.get(configs.API\_LIST\_ENDPOINT,  auth=HTTPBasicAuth(configs.USER,configs.PASSWORD))

1. In next step we are going to display all the tickets which are as follows:

def get\_all\_tickets()->json:

    '''

    This function  returns all the tickets

    '''

    try:

        # Trying to reach the API using username and password

        req=re.get(configs.API\_LIST\_ENDPOINT,  auth=HTTPBasicAuth(configs.USER,configs.PASSWORD))

        if req.status\_code!=200:

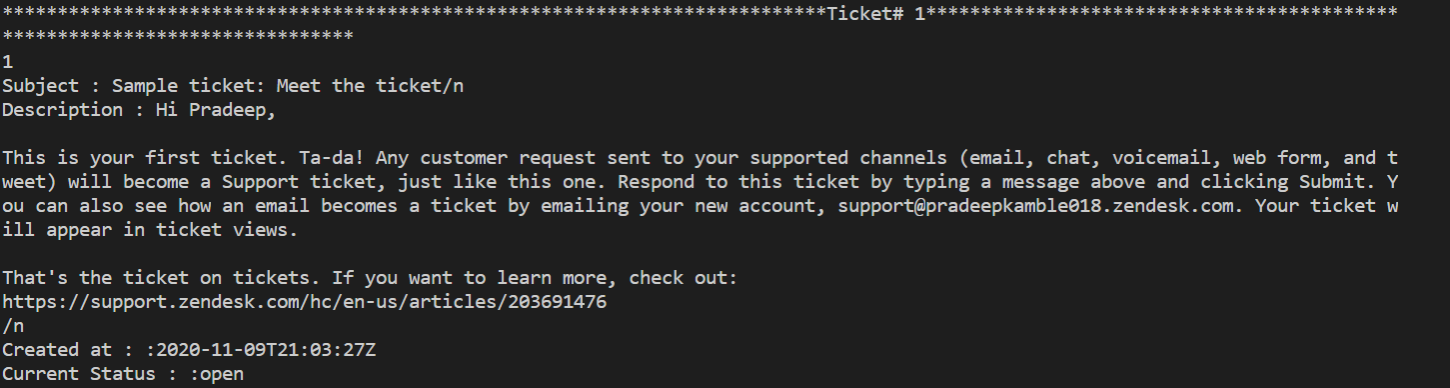
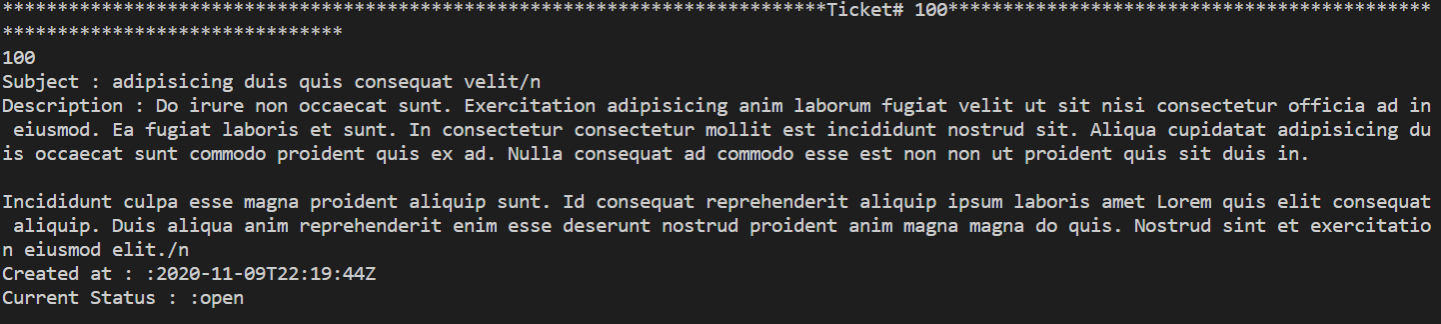
            raise Exception((req.status\_code))

        return req.json()

    except Exception as e:

        exception\_handler(e)

1. The output of the tickets from 1-100 which are as follows:



10)now we are going to display individual ticket details the code is as follows:

def get\_single\_ticket(i)->json:

    '''

    This Function retrieves 1 ticket at a time and returns its JSON

    '''

    try:

        if i in cached\_results:

            print("Entry Found in Cache")

            return cached\_results[i]

        else:

            req=re.get(configs.API\_TICKET\_ENDPOINT+str(i)+'.json?',  auth=HTTPBasicAuth(configs.USER,configs.PASSWORD))

            if req.status\_code!=200:

                raise Exception((req.status\_code))

            # print(req.json()['ticket']['subject'])

            response=req.json()

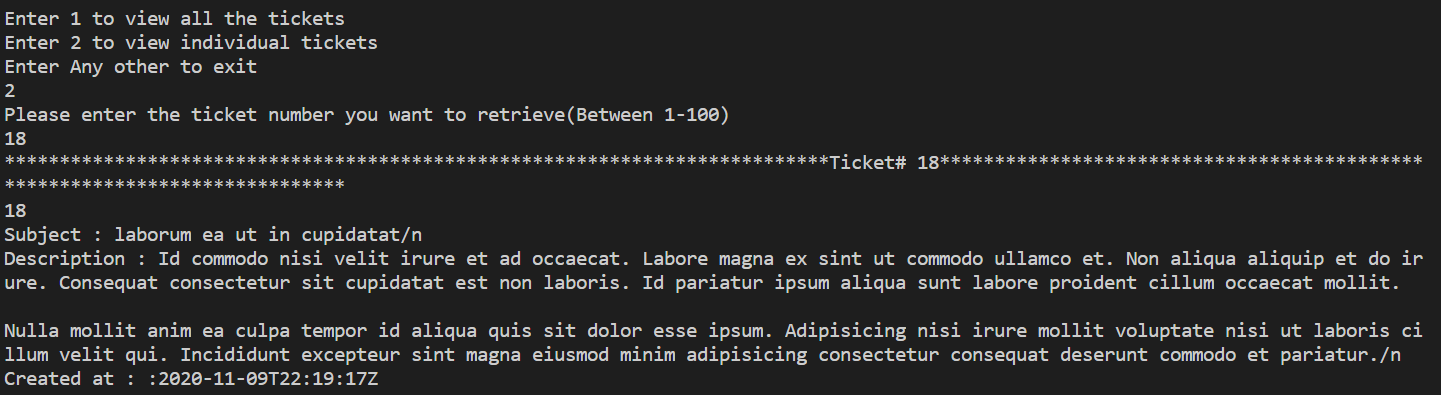
            ticket=response["ticket"]

            return ticket

    except Exception as e:

        exception\_handler()

1. The output of the individual tickets which are as follows:



1. Now we display them in a list the code is as follows:

def display\_ticket(i, ticket):

    '''This function is used for Displaying the tickets'''

    print("\*"\*75+"Ticket# "+str(i)+"\*"\*75)

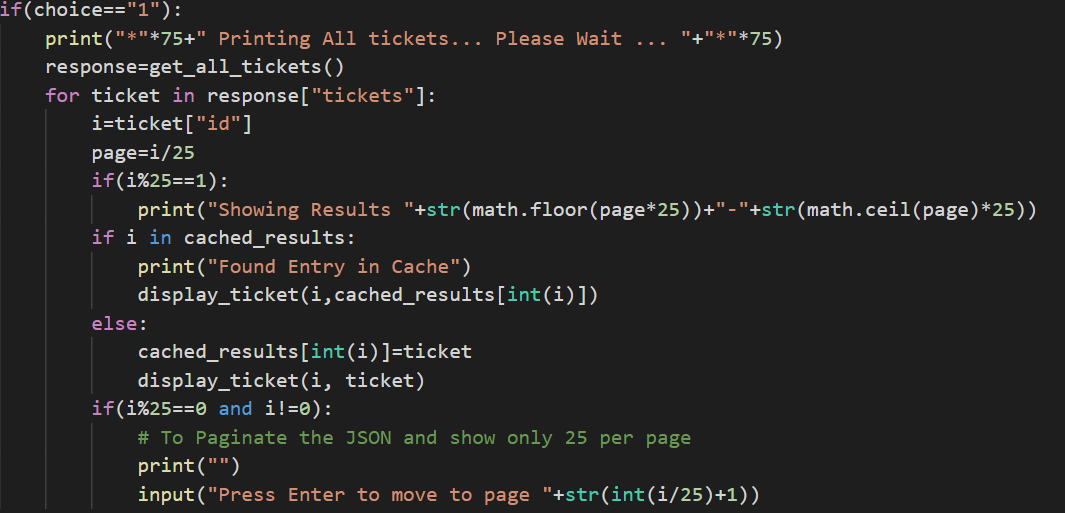
    print(ticket["id"])

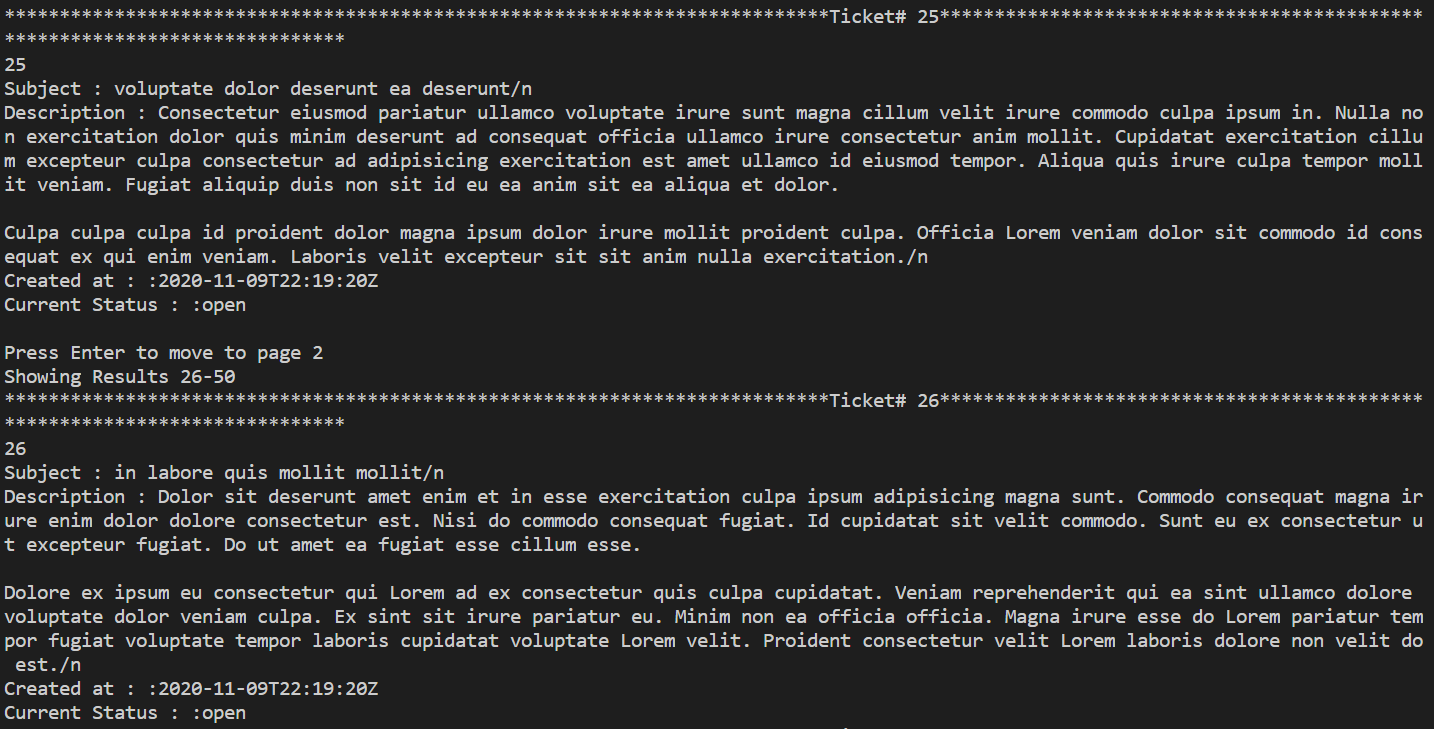
    print("Subject : "+ticket["subject"]+"/n")

    print("Description : "+ticket["description"]+"/n")

    print("Created at : "+":"+ticket["created\_at"])

    print("Current Status : "+":"+ticket["status"])

1. The output is same as we display all tickets earlier, so I do not attach same screenshot again.
2.  Now we are going through tickets when more than 25 are returned, the code is as follows:
3. So, we are showing 25 tickets in one page after 25 tickets it ask for another page to display next 25 tickets and it is goes on until 100 tickets.



1. So, we cover all the 5 Questions. Now I create separate README files for Task 2 to Task 7.

**THANK YOU**