SPRINT 04

Date	19 November 2022
Team ID	PNT2022TMID11066
Project Name	Smart solutions for railways
Maximum Marks	20 marks

FEED INFORMATION:

```
#Feed_Information
# Python program to find PNR
# status using RAILWAY API

import required modules
import requests, json

# Enter API key here
api_key = "Your_API_key"

# base_url variable to store url
base_url = "https://api.railwayapi.com/v2/pnr-status/pnr/"

# Enter valid pnr_number
pnr_number = "6515483790"

# Stores complete url address
complete_url = base_url + pnr_number + "/apikey/" + api_key + "/"

# get method of requests module
# return response object
response_ob = requests.get(complete_url)

# json method of response object convert
# json format data into python format data
result = response_ob.json()

# now result contains list
# of nested dictionaries
if result["response_code"] == 200:

        # train name is extracting
        # from the result variable data
        train_name = result["train"]["name"]

        # train number is extracting from
        # the result variable data
        train_number = result["train"]["number"]

        # from station name is extracting
```

```
# from the result variable data
from_station = result["from_station"]["name"]

# to_station name is extracting from
# the result variable data
to_station = result["to_station"]["name"]

# boarding point station name is
# extracting from the result variable data
boarding_point = result["boarding_point"]["name"]

# reservation upto station name is
# extracting from the result variable data
reservation_upto = result["reservation_upto"]["name"]

# store the value or data of "pnr"
# key in pnr_num variable
pnr_num = result["pnr"]

# store the value or data of "doj" key
# in variable date_of_journey variable
date_of_journey = result["doj"]

# store the value or data of
# "total_passengers" key in variable
total_passengers = result["total_passengers"]

# store the value or data of "passengers"
# key in variable passengers_list
passengers_list = result["passengers"]

# store the value or data of
# "chart_prepared" key in variable
chart_prepared = result["chart_prepared"]

# print following values
print(" train name : " + str(train_name)
        + "\n train number : " + str(train_number)
        + "\n from station : " + str(from_station)
        + "\n to station : " + str(to_station)
        + "\n boarding point : " + str(boarding_point)
        + "\n reservation upto : " + str(reservation_upto)
        + "\n pnr number : " + str(pnr_num)
        + "\n date of journey : " + str(date_of_journey)
        + "\n total no. of passengers: " + str(total_passengers)
        + "\n chart prepared : " + str(chart_prepared))

# looping through passenger list
for passenger in passengers_list:

        # store the value or data
        # of "no" key in variable
        passenger_num = passenger["no"]

        # store the value or data of
```

```
                # "current_status" key in variable
                current_status = passenger["current_status"]

                # store the value or data of
                # "booking_status" key in variable
                booking_status = passenger["booking_status"]

                # print following values
                print(" passenger number : " + str(passenger_num)
                      + "\n current status : " + str(current_status)
                      + "\n booking_status : " + str(booking_status))

else:
        print("Record Not Found")
```

ANY QUERIES:

```
#Any_Queries
import email, smtplib, ssl

from email import encoders
from email.mime.base import MIMEBase
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

subject = "An email with attachment from Python"
body = "This is an email with attachment sent from Python"
sender_email = "my@gmail.com"
receiver_email = "your@gmail.com"
password = input("Type your password and press enter:")

# Create a multipart message and set headers
message = MIMEMultipart()
message["From"] = sender_email
message["To"] = receiver_email
message["Subject"] = subject
message["Bcc"] = receiver_email  # Recommended for mass emails

# Add body to email
message.attach(MIMEText(body, "plain"))

filename = "document.pdf"  # In same directory as script

# Open PDF file in binary mode
with open(filename, "rb") as attachment:
    # Add file as application/octet-stream
    # Email client can usually download this automatically as attachment
    part = MIMEBase("application", "octet-stream")
    part.set_payload(attachment.read())

# Encode file in ASCII characters to send by email
encoders.encode_base64(part)
```

```
# Add header as key/value pair to attachment part
part.add_header(
    "Content-Disposition",
    f"attachment; filename= {filename}",
)

# Add attachment to message and convert message to string
message.attach(part)
text = message.as_string()

# Log in to server using secure context and send email
context = ssl.create_default_context()
with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
    server.login(sender_email, password)
    server.sendmail(sender_email, receiver_email, text)
```

RAISE QUERIES:

```
    #Raise_Queries
    import smtplib, ssl
    from email.mime.text import MIMEText
    from email.mime.multipart import MIMEMultipart

    sender_email = "my@gmail.com"
    receiver_email = "your@gmail.com"
    password = input("Type your password and press enter:")

    message = MIMEMultipart("alternative")
    message["Subject"] = "multipart test"
    message["From"] = sender_email
    message["To"] = receiver_email

    # Create the plain-text and HTML version of your message
    text = """\
    Hi,
    How are you?
    Real Python has many great tutorials:
    www.realpython.com"""
    html = """\
    <html>
      <body>
        <p>Hi,<br>
           How are you?<br>
           <a href="http://www.realpython.com">Real Python</a>
           has many great tutorials.
        </p>
      </body>
    </html>
    """

    # Turn these into plain/html MIMEText objects
```

```
part1 = MIMEText(text, "plain")
part2 = MIMEText(html, "html")

# Add HTML/plain-text parts to MIMEMultipart message
# The email client will try to render the last part first
message.attach(part1)
message.attach(part2)

# Create secure connection with server and send email
context = ssl.create_default_context()
with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
    server.login(sender_email, password)
    server.sendmail(
        sender_email, receiver_email, message.as_string()
    )
```

TICKET CANCELLATION:

```
#Ticket_Cancellation
from pickle import load,dump
import time
import random
import os
class tickets:
    def __init__(self):
        self.no_ofac1stclass=0
        self.totaf=0
        self.no_ofac2ndclass=0
        self.no_ofac3rdclass=0
        self.no_ofsleeper=0
        self.no_oftickets=0
        self.name=''
        self.age=''
        self.resno=0
        self.status=''
    def ret(self):
        return(self.resno)
    def retname(self):
        return(self.name)
    def display(self):
        f=0
        fin1=open("tickets.dat","rb")
        if not fin1:
            print "ERROR"
        else:
            print
            n=int(raw_input("ENTER PNR NUMBER : "))
            print "\n\n"
            print ("FETCHING DATA . . .".center(80))
            time.sleep(1)
            print
            print('PLEASE WAIT...!!'.center(80))
            time.sleep(1)
            os.system('cls')
```

```
        try:
            while True:
                tick=load(fin1)
                if(n==tick.ret()):
                    f=1
                    print "="*80
                    print("PNR STATUS".center(80))
                    print"="*80
                    print
                    print "PASSENGER'S NAME :",tick.name
                    print
                    print "PASSENGER'S AGE :",tick.age
                    print
                    print "PNR NO :",tick.resno
                    print
                    print "STATUS :",tick.status
                    print
                    print "NO OF SEATS BOOKED : ",tick.no_oftickets
                    print
        except:
            pass
        fin1.close()
        if(f==0):
            print
            print "WRONG PNR NUMBER..!!"
            print
def pending(self):
    self.status="WAITING LIST"
    print "PNR NUMBER :",self.resno
    print
    time.sleep(1.2)
    print "STATUS = ",self.status
    print
    print "NO OF SEATS BOOKED : ",self.no_oftickets
    print
def confirmation (self):
    self.status="CONFIRMED"
    print "PNR NUMBER : ",self.resno
    print
    time.sleep(1.5)
    print "STATUS = ",self.status
    print
def cancellation(self):
    z=0
    f=0
    fin=open("tickets.dat","rb")
    fout=open("temp.dat","ab")
    print
    r= int(raw_input("ENTER PNR NUMBER : "))
    try:
        while(True):
            tick=load(fin)
            z=tick.ret()
            if(z!=r):
                dump(tick,fout)
```

```
            elif(z==r):
                f=1
        except:
            pass
        fin.close()
        fout.close()
        os.remove("tickets.dat")
        os.rename("temp.dat","tickets.dat")
        if (f==0):
            print
            print "NO SUCH RESERVATION NUMBER FOUND"
            print
            time.sleep(2)
            os.system('cls')
        else:
            print
            print "TICKET CANCELLED"
            print"RS.600 REFUNDED...."
    def reservation(self):
        trainno=int(raw_input("ENTER THE TRAIN NO:"))
        z=0
        f=0
        fin2=open("tr1details.dat")
        fin2.seek(0)
        if not fin2:
            print "ERROR"
        else:
            try:
                while True:
                    tr=load(fin2)
                    z=tr.gettrainno()
                    n=tr.gettrainname()
                    if (trainno==z):
                        print
                        print "TRAIN NAME IS : ",n
                        f=1
                        print
                        print "-"*80
                        no_ofac1st=tr.getno_ofac1stclass()
                        no_ofac2nd=tr.getno_ofac2ndclass()
                        no_ofac3rd=tr.getno_ofac3rdclass()
                        no_ofsleeper=tr.getno_ofsleeper()
                    if(f==1):
                        fout1=open("tickets.dat","ab")
                        print
                        self.name=raw_input("ENTER THE PASSENGER'S NAME ")
                        print
                        self.age=int(raw_input("PASSENGER'S AGE : "))
                        print
                        print"\t\t SELECT A CLASS YOU WOULD LIKE TO TRAVEL IN :- "
                        print "1.AC FIRST CLASS"
                        print
                        print "2.AC SECOND CLASS"
                        print
                        print "3.AC THIRD CLASS"
```

```
                print
                print "4.SLEEPER CLASS"
                print
                c=int(raw_input("\t\t\tENTER YOUR CHOICE = "))
                os.system('cls')
                amt1=0
                if(c==1):
                   self.no_oftickets=int(raw_input("ENTER NO_OF FIRST CLASS AC
SEATS TO BE BOOKED : "))
                   i=1
                   while(i<=self.no_oftickets):
                      self.totaf=self.totaf+1
                      amt1=1000*self.no_oftickets
                      i=i+1
                   print
                   print "PROCESSING. .",
                   time.sleep(0.5)
                   print ".",
                   time.sleep(0.3)
                   print'.'
                   time.sleep(2)
                   os.system('cls')
                   print "TOTAL AMOUNT TO BE PAID = ",amt1
                   self.resno=int(random.randint(1000,2546))
                   x=no_ofac1st-self.totaf
                   print
                   if(x>0):
                      self.confirmation()
                      dump(self,fout1)
                      break
                   else:
                      self.pending()
                      dump(tick,fout1)
                      break
                elif(c==2):
                   self.no_oftickets=int(raw_input("ENTER NO_OF SECOND CLASS AC
SEATS TO BE BOOKED :  "))
                   i=1



   def menu():
      tr=train()
      tick=tickets()
      print
      print "WELCOME TO PRAHIT AGENCY".center(80)
      while True:
            print
            print "="*80
            print " \t\t\t\t  RAILWAY"
            print
            print "="*80
            print
            print "\t\t\t1. **UPDATE TRAIN DETAILS."
            print
```

```
            print "\t\t\t2. TRAIN DETAILS. "
            print
            print "\t\t\t3. RESERVATION OF TICKETS."
            print
            print "\t\t\t4. CANCELLATION OF TICKETS. "
            print
            print "\t\t\t5. DISPLAY PNR STATUS."
            print
            print "\t\t\t6. QUIT."
            print"** - office use....."
            ch=int(raw_input("\t\t\tENTER YOUR CHOICE : "))
            os.system('cls')
            print "\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t\tLOADING. .",
            time.sleep(1)
            print ("."),
            time.sleep(0.5)
            print (".")
            time.sleep(2)
            os.system('cls')
            if ch==1:
               j="*****"
               r=raw_input("\n\n\n\n\n\n\n\n\n\n\t\t\t\tENTER THE PASSWORD: ")
               os.system('cls')
               if (j==r):
                  x='y'
                  while (x.lower()=='y'):
                     fout=open("tr1details.dat","ab")
                     tr.getinput()
                     dump(tr,fout)
                     fout.close()
                     print"\n\n\n\n\n\n\n\n\n\n\t\t\tUPDATING TRAIN LIST PLEASE
WAIT . .",
                     time.sleep(1)
                     print ("."),
                     time.sleep(0.5)
                     print ("."),
                     time.sleep(2)
                     os.system('cls')
                     print "\n\n\n\n\n\n\n\n\n\n"
                     x=raw_input("\t\tDO YOU WANT TO ADD ANY MORE TRAINS
DETAILS ? ")
                     os.system('cls')
                  continue
               elif(j<>r):
                  print"\n\n\n\n\n"
                  print "WRONG PASSWORD".center(80)
            elif ch==2:
               fin=open("tr1details.dat",'rb')
               if not fin:
                  print "ERROR"
               else:
                  try:
                     while True:
                        print"*"*80
                        print"\t\t\t\tTRAIN DETAILS"
```

```
                print"*"*80
                print
                tr=load(fin)
                tr.output()



                raw_input("PRESS ENTER TO VIEW NEXT TRAIN DETAILS")
                os.system('cls')
            except EOFError:
                pass
        elif ch==3:
            print'='*80
            print "\t\t\t\tRESERVATION OF TICKETS"
            print'='*80
            print
            tick.reservation()
        elif ch==4:
            print"="*80
            print"\t\t\t\tCANCELLATION OF TICKETS"
            print
            print"="*80
            print
            tick.cancellation()
        elif ch==5:
            print "="*80
            print("PNR STATUS".center(80))
            print"="*80
            print
            tick.display()
        elif ch==6:
            quit()

        raw_input("PRESS ENTER TO GO TO BACK MENU".center(80))
        os.system('cls')

menu()
```