**import** **json**

**import** **requests**

**from** **datetime** **import** datetime

**from** **dateutil** **import** parser # pip install python-dateutil

**from** **pytz** **import** timezone # pip install pytz

# https://www.youtube.com/watch?v=BI4HxHn1HmQ

# i have used pytz for time zone

# to install pytz, use this comand : easy\_install --upgrade pytz

# this was the site where i have found about pytz : http://pytz.sourceforge.net/

# dateutil is also required. it is used for parsing dates.

# a function that changes timezoe of a date and returns a datetime object converted into the specified timezone

**def** **change\_timezone\_of\_datetime\_object**(date\_time\_object, new\_timezone\_name):

"""Return the \*date\_time\_object\* with it's timezone changed to \*new\_timezone\_name\*

:param date\_time\_object: The datetime object whose timezone is to be changed

:param new\_timezone\_name: The name of the timezone to which the \*date\_time\_object\* is to be changed to

"""

#https://www.youtube.com/watch?v=zY02utxcauo for Parsing and Formatting Dates in Python With Datetime

# for the new\_timezone

new\_timezone\_object = timezone(new\_timezone\_name)

# it will update our timezone of the datetime in object

date\_time\_object = date\_time\_object.astimezone(new\_timezone\_object)

# Return the converted datetime object

**return** date\_time\_object

# this is the URL of our API means base url

api\_end\_point = 'https://api.list.co.uk/v1/events'

# a variable to store our API key

api\_key = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiZWY1ODFkMjYtY2Q3NC00Yzg1LTgzMTQtNDY0MzRlZTlmODAwIiwia2V5X2lkIjoiZWYzZjczZGUtODkwMy00MjkxLTlkZjEtYzA2MDIyNWVlZTcwIiwiaWF0IjoxNTg0Njk1NTQ2fQ.9jD\_DmWBhdMcPVFglPXtR1XIpgGDi9pLSfYt5\_MFVO8"

# (from the docmentation guide)a variable to create headers for the request

headers = {

"Authorization": "Bearer " + api\_key

}

#i have watched youtube video to understand https://www.youtube.com/watch?v=g4wdm488mkE

# to call API and store the respose returned in a variable

response = requests.get(api\_end\_point, headers = headers)

# convert our string and display the returned response text as JSON

event\_list = json.loads(response.text)

event\_count = **1**

label\_prefix = "|\_ "

# i have used for and if command to use print my events

**for** event **in** event\_list:

**print**("**\n\n**EVENT # %d " % event\_count, end='')

**print**("-"\***80**) # just a line to separate the events

**print**(label\_prefix + "Name : " + event['name'])

**print**(label\_prefix + "Sort Name : " + event['sort\_name'])

**print**(label\_prefix + "Status : " + event['status'])

**for** descr **in** event['descriptions']:

**if** descr['type'] == "third-party":

**print**(label\_prefix + "Description : " + descr['description'])

place = event['schedules'][**0**]['place']['name']

place\_address = event['schedules'][**0**]['place']['address'] + ", " + event['schedules'][**0**]['place']['town']

place\_address += " - " + event['schedules'][**0**]['place']['postal\_code']

**print**(label\_prefix + "Place : " + place)

**print**(label\_prefix + "**\t** Address : " + place\_address)

# parse the string date using dateutil parser

dt = parser.parse(event['schedules'][**0**]['start\_ts'])

# covert the UTC datetime to local datetime

dt = change\_timezone\_of\_datetime\_object(dt, 'Europe/London')

**print**(label\_prefix + "Start time : " + dt.ctime()) # use the converted datetime object

dt = parser.parse(event['schedules'][**0**]['end\_ts'])

dt = change\_timezone\_of\_datetime\_object(dt, 'Europe/London')

**print**(label\_prefix + "End time : " + dt.ctime())

# for ticket summury if event has ticket info

**if** "ticket\_summary" **in** event['schedules'][**0**]:

**print**(label\_prefix + "Ticket Range : " + event['schedules'][**0**]['ticket\_summary'])

**else**:

**print**(label\_prefix + "Ticket Range : NOT SPECIFIED!")

performance\_count = len(event['schedules'][**0**]['performances'])

**print**("**\n** %d performances found..." % performance\_count)

performance\_count = **1**

# to print performance info

**for** performance **in** event['schedules'][**0**]['performances']:

**print**(label\_prefix + "**\t** Performance #%d" % performance\_count)

dt = parser.parse(performance['ts'])

dt = change\_timezone\_of\_datetime\_object(dt, 'Europe/London')

**print**(label\_prefix + "**\t\t** Start time : " + dt.ctime())

performance\_count += **1**

event\_count += **1**

exitline=input("ënter to exit")

**print** ("Thats all we Got")