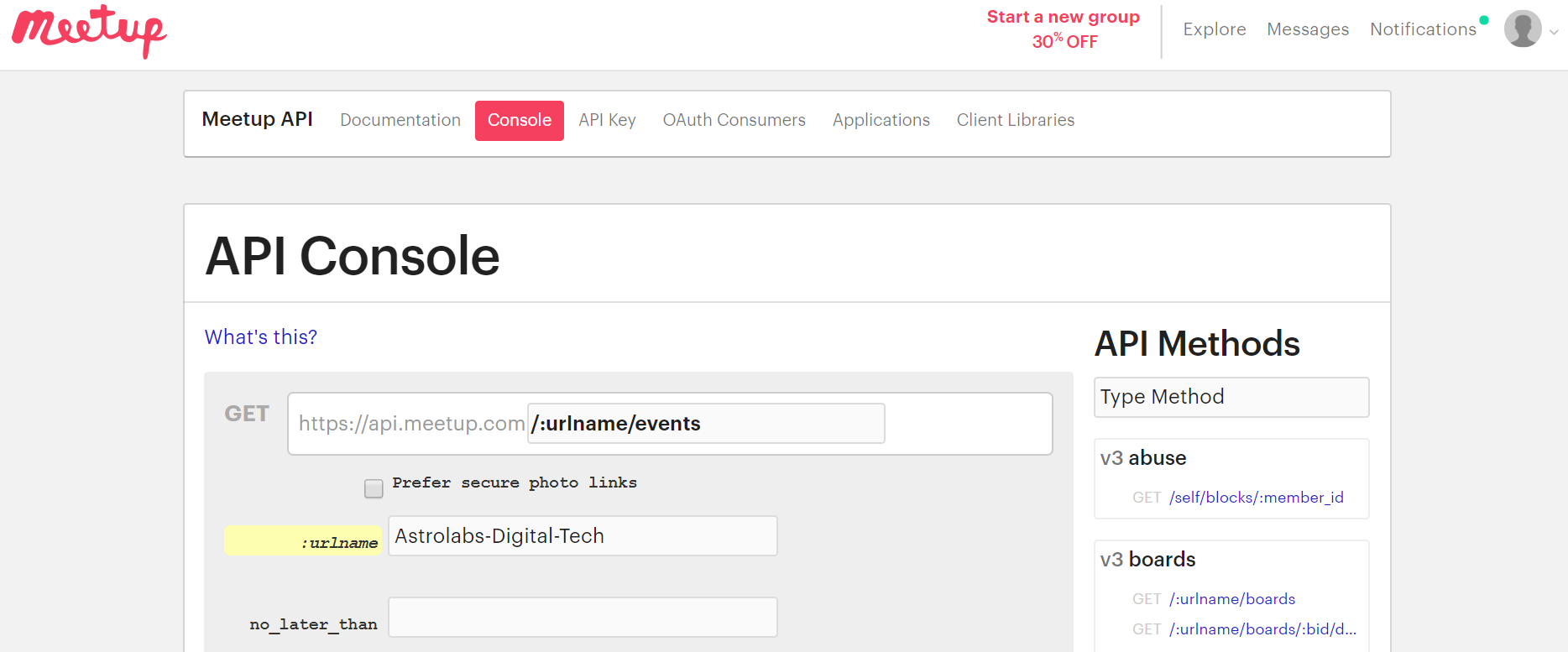
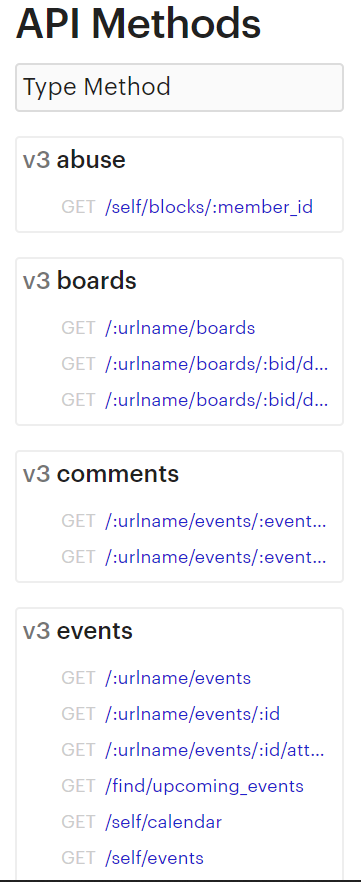
**RETRIEVING DATA FROM MEETUP.COM USING THEIR API CONSOLE**

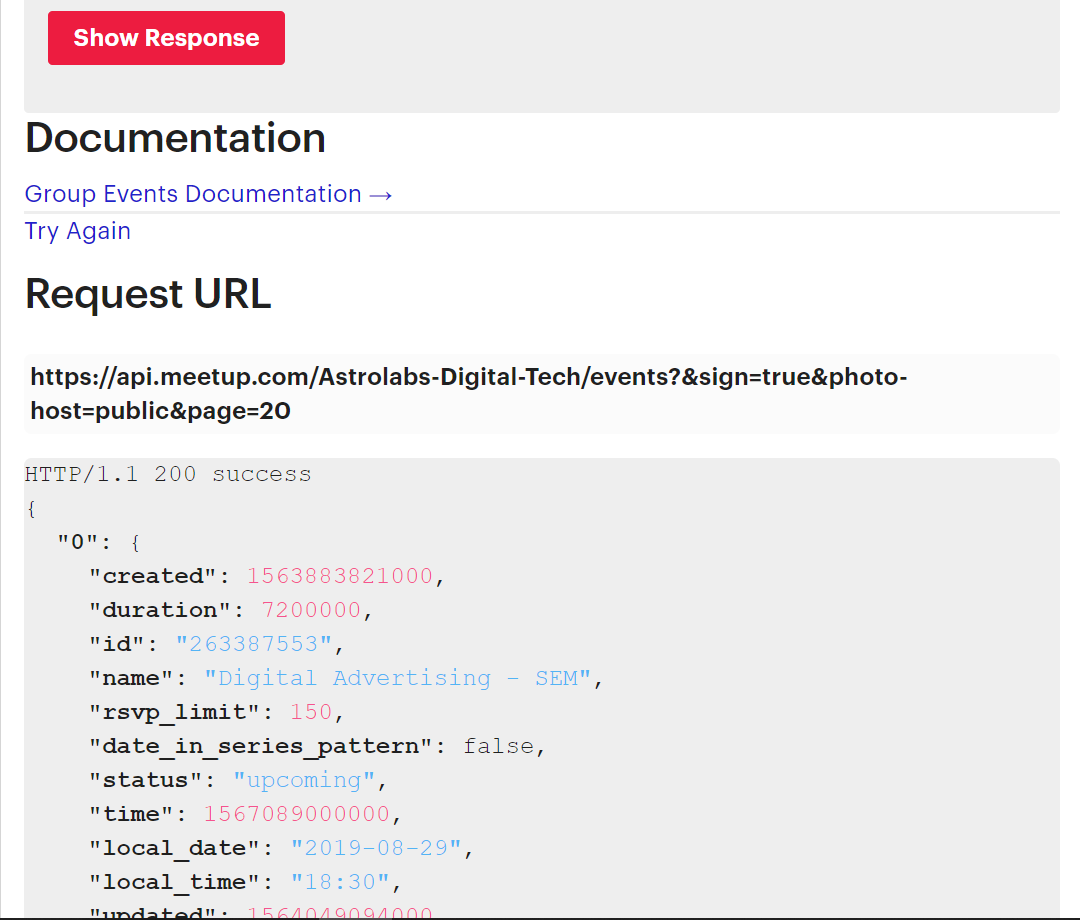
The below picture is the API console of meetup.com. This console communicates with the Meetup API over JavaScript.



1. Select an API method from the list on the right or enter a method's path below. This console only supports API methods that use the HTTP GET method.



1. Click the Show Response button to see the API response in [JSON](http://www.json.org/) format.



**CONVERTING DATA FROM URL TO .JSON AND .CSV FILE TYPE.**

import pandas as pd #library for data manipulation and analysis

import meetup.api #libraray for meetup API

import json

import requests #HTTP library for python

import time

import codecs

import sys

import io

**#request for data from the Request URL highlighted on the API console page above**

df = requests.get('https://api.meetup.com/Astrolabs-Digital-Tech/events?&sign=true&photo-host=public&page=20')

df.headers['content-type']

df.encoding

df.json()

data = df.json()

**#Converting data to JSON format**

try:

to\_unicode = unicode

except NameError:

to\_unicode = str

with io.open('dfw4.json', 'w', encoding='utf8') as outfile:

str\_ = json.dumps(data,

indent=4, sort\_keys=True,

separators= (',', ': '), ensure\_ascii=False)

outfile.write(to\_unicode(str\_))

**#Libraries required for plotting a graph**

%matplotlib inline

import matplotlib.pyplot as plt #plotting library which produces publication quality figures for python

import pandas as pd

import numpy as np

**#Reading from csv file**

df = pd.read\_csv("convertcsv.csv")

plt.rcParams["figure.figsize"] = (30,10)

plt.plot(df.name,df.yes\_rsvp\_count)

plt.xlabel('Events at AstroLabs')

plt.ylabel('RSVP count')

plt.show()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**#Plotting a pie chart**

plt.pie(

df['yes\_rsvp\_count'],

labels=df['name'],

shadow=False,

startangle=90,

autopct='%1.1f%%') #percent listed as a fraction

# View the plot drop above

plt.axis('equal')

# View the plot

plt.tight\_layout()

plt.show()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**#Plotting a categorical plotting**

names=df['name']

plt.figure(figsize=(9, 3))

plt.subplot(131)

plt.bar(names, df['yes\_rsvp\_count'])

plt.subplot(132)

plt.scatter(names, df['yes\_rsvp\_count'])

plt.subplot(133)

plt.plot(names, df['rsvp\_limit'])

plt.suptitle('Categorical Plotting')

plt.show()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**#Plotting a frequency chart**

df['duration'].value\_counts().plot('bar')

References

1. Frequency plots: <https://etav.github.io/python/count_basic_freq_plot.html>
2. Pyplot: <https://matplotlib.org/3.1.1/tutorials/introductory/pyplot.html#sphx-glr-tutorials-introductory-pyplot-py>
3. Pie charts: <https://chrisalbon.com/python/data_visualization/matplotlib_pie_chart/>