**Exporting LaunchDarkly Flag List into a CSV File with Python**

At the moment, [LaunchDarkly](https://launchdarkly.com/) does not have functionality to export a list of flags as csv or excel file. This can change very near future (it may already have the functionality by the time you are reading this post). The workaround is to use API to ingest the data.

Here is the quick and dirty Python script to do it. You can replace the API key and endpoint and use it straight away!

I included epoch time conversion functions, too.

The catch is that I am not sure about the maximum number of flags we can get. I had about 140 flags and there was no pagenation. Documentation doesn’t really mention it, either. So, if you see the pagination in JSON file, you need to do a loop to get the next page.

[cc lang="python" tab\_size="4" lines="-1"]

import json, requests

import time

from time import strptime, strftime, mktime, gmtime, localtime

url = 'https://app.launchdarkly.com/api/v2/flags/default?env=production'

token = "<your api token>"

headers = {"Authorization": token, 'Content-Type':'application/json'}

r = requests.get(url, headers=headers)

prettyJson = json.dumps(r.json(), indent=4, sort\_keys=True)

print(r.status\_code)

# print(r.headers)

# print(prettyJson)

f = open('./data/feature\_flag\_list.json', 'w')

f.write(prettyJson)

print('file feature\_flag\_list.json created!')

print('Starting transformation')

def epoch\_to\_stamp(epochtime):

'''This function converts epochtime to timestamp'''

return time.strftime('%Y-%m-%d %H:%M:%S', time.gmtime(epochtime/1000.0))

def convert\_epoch\_to\_local(epoch):

'''Converting epoch to local time'''

return strftime('%Y-%m-%d %H:%M:%S', localtime(epoch))

featureFlagList = []

repsonse = r.json()

items = repsonse['items']

for i in items:

tmp = []

tmp.append(i["name"])

tmp.append(i["key"])

tmp.append(i["kind"])

tmp.append(epoch\_to\_stamp(i["creationDate"]))

tmp.append(epoch\_to\_stamp(i["environments"]["production"]["lastModified"]))

tmp.append(str(i["environments"]["production"]["on"]))

try:

tmp.append(i["\_maintainer"]["email"])

except KeyError:

tmp.append('NA')

featureFlagList.append(tmp)

headers = ['name','key','kind','creationDate','lastModified','flagValue', 'email']

csv = open('./data/feature\_flag\_list.csv', 'w')

csv.write(','.join(headers)+'\n')

for i in featureFlagList:

csv.write(','.join(i)+'\n')

print('feature\_flag\_list.csv created!')

csv.close()

[/cc]

But, wait. I’ve got something even better.

I created a React Feature Flag dashboard with TypeScript with AWS Cognito authentication. You can clone the repo and add the appropriate config file as in README. Then, you will have an awesome serverless web dashboard to visualise your flags.

Check out the repo here: [launch-darkly-flag-dashboard](https://github.com/mydatahack/launch-darkly-flag-dashboard)