MN Geospatial Commons API

This script searches the MN Geospatial Commons using tags and returns search results. It then extracts one specified dataset from the search results or all of the datasets, and unzips the extracted datasets.

In [1]:
```
import requests
from zipfile import ZipFile
```

Exploring the API - possible datasets, tags, and groups

In [2]:
```
packages = requests.get("https://gisdata.mn.gov/api/3/action/package_list", v
tag_list = requests.get("https://gisdata.mn.gov/api/3/action/tag_list", verif
group_list = requests.get("https://gisdata.mn.gov/api/3/action/group_list", v
```

```
C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\lib\site-packages
\urllib3\connectionpool.py:1004: InsecureRequestWarning: Unverified HTTPS r
equest is being made to host 'gisdata.mn.gov'. Adding certificate verificat
ion is strongly advised. See: https://urllib3.readthedocs.io/en/latest/adva
nced-usage.html#ssl-warnings (https://urllib3.readthedocs.io/en/latest/adva
nced-usage.html#ssl-warnings)
  InsecureRequestWarning,
C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\lib\site-packages
\urllib3\connectionpool.py:1004: InsecureRequestWarning: Unverified HTTPS r
equest is being made to host 'gisdata.mn.gov'. Adding certificate verificat
ion is strongly advised. See: https://urllib3.readthedocs.io/en/latest/adva
nced-usage.html#ssl-warnings (https://urllib3.readthedocs.io/en/latest/adva
nced-usage.html#ssl-warnings)
  InsecureRequestWarning,
C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\lib\site-packages
\urllib3\connectionpool.py:1004: InsecureRequestWarning: Unverified HTTPS r
equest is being made to host 'gisdata.mn.gov'. Adding certificate verificat
ion is strongly advised. See: https://urllib3.readthedocs.io/en/latest/adva
nced-usage.html#ssl-warnings (https://urllib3.readthedocs.io/en/latest/adva
nced-usage.html#ssl-warnings)
  InsecureRequestWarning,
```

In [3]:
```
# All datasets available
packages_dict = packages.json()
assert packages_dict['success'] is True

# All tags availabile
tag_dict = tag_list.json()
assert tag_dict['success'] is True

# All categories available
group_dict = group_list.json()
assert group_dict['success'] is True
```

In [4]:
```
datasets = packages_dict['result']
tags = tag_dict['result']
groups = group_dict['result']
```

In [5]:
```
# adjust index to see tags more or less tags
print(f"tags: {tags[:10]}")
print(f"\ngroups: {groups}")
```

```
tags: ['100k index', '103e', '15 minute', '1994', '1-meter orthophoto', '1
water supply planning working groups', '2000 census', '2002', '2010', '2010
census']

groups: ['biota', 'boundaries', 'climatology', 'economy', 'elevation', 'env
ironment', 'farming', 'geoscientific', 'health', 'imagery-basemaps', 'inlan
d-waters', 'intelligence-military', 'location', 'planning-cadastre', 'socie
ty', 'structure', 'transportation', 'utilities-communication']
```

In [6]:
```
# run this cell to view all datasets possible
datasets
```

Out[6]:
```
['agri-agbmp-loans',
 'agri-agroecoregions',
 'agri-app-food-for-thought-maps',
 'agri-app-inspection-territories',
 'agri-bah-field-staff',
 'agri-cropland-data-layer-2006',
 'agri-cropland-data-layer-2007',
 'agri-cropland-data-layer-2008',
 'agri-cropland-data-layer-2009',
 'agri-cropland-data-layer-2010',
 'agri-cropland-data-layer-2011',
 'agri-cropland-data-layer-2012',
 'agri-cropland-data-layer-2013',
 'agri-cropland-data-layer-2014',
 'agri-cropland-data-layer-2015',
 'agri-cropland-data-layer-2016',
 'agri-cropland-data-layer-2017',
 'agri-cropland-data-layer-2018',
 'agri-cropland-data-layer-2019',
```

Querying and extracting data

In [7]:
```
# use '+' to seperate multiple tags in query i.e 2020+farming
query = "2030"

search_response = requests.get(f"https://gisdata.mn.gov/api/3/action/package_
```

```
C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\lib\site-packages
\urllib3\connectionpool.py:1004: InsecureRequestWarning: Unverified HTTPS r
equest is being made to host 'gisdata.mn.gov'. Adding certificate verificat
ion is strongly advised. See: https://urllib3.readthedocs.io/en/latest/adva
nced-usage.html#ssl-warnings (https://urllib3.readthedocs.io/en/latest/adva
nced-usage.html#ssl-warnings)
  InsecureRequestWarning,
```

In [8]:
```
search = search_response.json()
assert search['success'] is True
```

In [9]:
```
print(f"Search result count: {search['result']['count']}")
```

```
Search result count: 9
```

In [11]:
```
# Get url for first result only
data = search['result']['results'][0]['resources'][1]['url']
data_id = search['result']['results'][0]['resources'][1]['id']
out_name = f"{data_id}.zip"

print(data)

# Write file to specified output type. Out path is same as this jupyter noteb
r = requests.get(data)
assert r.status_code is 200

with open(out_name, "wb") as file:
    file.write(r.content)

with ZipFile(out_name, "r") as zipped:
    print(zipped.namelist())
    zipped.extractall()
```

```
https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_metc/plan_fr
mwrk2030dev_plan_ar/shp_plan_frmwrk2030dev_plan_ar.zip (https://resources.g
isdata.mn.gov/pub/gdrs/data/pub/us_mn_state_metc/plan_frmwrk2030dev_plan_a
r/shp_plan_frmwrk2030dev_plan_ar.zip)
['Framework2030PlanningAreas.shx', 'Framework2030PlanningAreas.sbx', 'Frame
work2030PlanningAreas.cpg', 'Framework2030PlanningAreas.prj', 'Framework203
0PlanningAreas.shp', 'Framework2030PlanningAreas.sbn', 'Framework2030Planni
ngAreas.dbf', 'Framework2030PlanningAreas.shp.xml', 'metadata/metadata.htm
l', 'metadata/metadata.xml']
```

In [12]:
```
# Get the urls for ALL the search results. Can specify format type.
# Could you use this to extract all data from each url?
search_urls = []
ids = []
for i in search['result']['results']:
    for x in i['resources']:
        if x['format'] == 'SHP': # return only specified shp file
            search_urls.append(x['url'])
            ids.append(x['id'])
```

In [ ]:
```
# Extract data from all the URLS and named with their ID
for i in range(len(search_urls)):
    r = requests.get(search_urls[i])
    assert r.status_code is 200

    with open(f"{ids[i]}.zip", "wb") as file:
        file.write(r.content)

    with ZipFile(f"{ids[i]}.zip", "r") as zipped:
        print(zipped.namelist())
        zipped.extractall()
```

In [ ]:
```

```