NDAWN API

The aim of this code is to retreive data from the NDAWN API using user prompts. Data comes in the form of a csv file. The temporal range for this script is hourly to monthly. This does not include daily or month normal dataset retrieval.

In [1]:
```
import requests
```

In [2]:
```
def get_variables(table):
    '''
    parameters
    ----------

    return
    ------
    '''
    print("")

    for key in table:
        print(f"{key}: {table[key]}")

    in_vars = input("\nEnter the variable codes you are interested in seperat
    list_vars = in_vars.split(",")

    for var in list_vars:
        assert var in table, "Invalid variable entered"

    format_vars = f"&variable={'&variable='.join(list_vars)}"
    return format_vars
```

In [3]:
```
def extract_data(url, ttype, start_date):
    '''
    parameters
    ----------

    prints
    ------
    '''
    r = requests.get(url)
    assert r.status_code is 200, ("Url is not valid")
    assert "Error" not in r.text, ("Url is not valid")
    print(url)
    print("Url valid. Extracting...")

    with open(f"{ttype}_{start_date.replace('-', '')}.csv", "wb") as file:
        file.write(r.content)
        print("File extracted.")
```

In [4]:
```
base_url = "https://ndawn.ndsu.nodak.edu/table.csv?"

time_options = ("hourly", "daily", "weekly", "monthly")
print(f"Time options are: {time_options}")
ttype = input("Enter the time type: ")
assert ttype in time_options, "Invalid time type"

station = input("Enter the station number: ")
```

```
Time options are: ('hourly', 'daily', 'weekly', 'monthly')
Enter the time type: weekly
Enter the station number: 78
```

In [5]:
```
# Run this cell. Lookup dictionaries for each time type
# how can I get these variables dynamically?

hourly_variables = {
    "hdt": "avg air temp", "hdrh": "avg relative humidity", "hdbst": "avg bar
    "hdtst": "avg turf soil temp", "hdws": "avg wind speed", "hdmxws": "max w
    "hdsdwd": "avg wind direction std dev","hdsr": "total solar radiation","h
    "hdbp": "avg barometric pressure","hddp": "avg dew point","hdwc": "avg wi
    "hdt9": "avg air temp at 9 meters","hdrh9": "avg relative humidity at 9 m
    "hdws10": "avg wind speed at 10 meters","hdmxws10": "max wind speed at 10
    "hdwd10": "avg wind direction at 10 meters","hdsdwd10": "avg wind direct
}

daily_variables = {
    "ddmxt": "max air temp","ddmxtt": "max(time) air temp","ddmnt": "min air
    "ddavt": "avg air temp","dddtr": "diurnal range - air temp","ddbst": "avg
    "ddtst": "avg turf soil temp","ddws": "avg wind speed","ddmxws": "max wir
    "ddmxwst": "max(time) wind speed","ddwd": "avg wind direction","ddwdsd":
    "ddsr": "total solar radiation","ddtpetp": "total pet (penman)","ddtpetjh
    "ddr": "total rainfall","dddp": "avg dew point","ddwc": "avg wind chill",
    "ddmxt9": "max air temp at 9 meters","ddmxtt9": "max(time) air temp at 9
    "ddmnt9": "min air temp at 9 meters","ddmntt9": "max(time) air temp at 9
    "ddmxws10": "max wind speed at 10 meters","ddmxwst10": "max(time) wind sp
    "ddwd10": "avg wind direction at 10 meters","ddwdsd10": "avg wind directi
}

weekly_variables = {
    "wdmxt": "max air temp","wdmnt": "min air temp","wdavt": "avg air temp","
    "wdtst": "avg turf soil temp","wdws": "avg wind speed","wdmxws": "max wir
    "wdsr": "total solar radiation","wdapet": "avg pet (penman)","wdtpet": "t
    "wdr": "total rainfall","wddp": "avg dew point","wdwc": "avg wind chill"
}


monthly_variables = {
    "mdmxt": "max air temp","mdmnt": "min air temp","mdavt": "avg air temp","
    "mdtst": "avg turf soil temp","mdws": "avg wind speed","mdmxws": "max wir
    "mdsr": "total solar radiation","mdapet": "avg pet (penman)","mdtpet": "t
    "mdr": "total rainfall","mddp": "avg dew point","mdwc": "avg wind chill"
}
```

In [6]:

```
# define parameters based on ttype

if ttype == "hourly":
    table = hourly_variables

    start_date = input("Enter dataset begin date YYYY-MM-DD: ")
    end_date = input("Enter dateset end date YYYY-MM-DD: ")
    assert end_date.replace("-","") >= start_date.replace("-", ""), "End date


    variables = get_variables(table)

    url = f"{base_url}station={station}{variables}&ttype={ttype}&quick_pick=&
    extract_data(url, ttype, start_date)


elif ttype == "daily":
    table = daily_variables

    start_date = input("Enter dataset begin date YYYY-MM-DD: ")
    end_date = input("Enter dateset end date YYYY-MM-DD: ")
    assert end_date.replace("-","") >= start_date.replace("-", ""), "End date

    variables = get_variables(table)

    url = f"{base_url}station={station}{variables}&ttype={ttype}&quick_pick=&
    extract_data(url, ttype, start_date)



elif ttype == "weekly":
    table = weekly_variables

    start_date = input("Enter dataset begin date YYYY-MM-DD: ")
    count = input("Enter the number of weeks interested: ")

    variables = get_variables(table)

    url = f"{base_url}station={station}{variables}&ttype={ttype}&quick_pick=&
    extract_data(url, ttype, start_date)


elif ttype == "monthly":
    table = monthly_variables

    start_date = input("Enter dataset begin month YYYY-MM: ")
    count = input("Enter the number of months interested: ")

    variables = get_variables(table)

    url = f"{base_url}station={station}{variables}&ttype={ttype}&quick_pick=&
    extract_data(url, ttype, start_date)
```

```
Enter dataset begin date YYYY-MM-DD: 2021-01-02
Enter the number of weeks interested: 2
```

```
wdmxt: max air temp
wdmnt: min air temp
wdavt: avg air temp
wdbst: avg bare soil temp
wdtst: avg turf soil temp
wdws: avg wind speed
wdmxws: max wind speed
wdsr: total solar radiation
wdapet: avg pet (penman)
wdtpet: total pet (penman)
wdr: total rainfall
wddp: avg dew point
wdwc: avg wind chill

Enter the variable codes you are interested in seperated by a comma: wdmx
t,wdws,wdavt
https://ndawn.ndsu.nodak.edu/table.csv?station=78&variable=wdmxt&variable
=wdws&variable=wdavt&ttype=weekly&quick_pick=&begin_date=2021-01-02&count
=2 (https://ndawn.ndsu.nodak.edu/table.csv?station=78&variable=wdmxt&vari
able=wdws&variable=wdavt&ttype=weekly&quick_pick=&begin_date=2021-01-02&c
ount=2)
Url valid. Extracting...
File extracted.
```

```
In [ ]:
```