

```
import requests
def make_request(endpoint, payload=None):

    return requests.get(
        f'https://v1.earthdata.nasa.gov/cdo-web/api/v2/{endpoint}',
        headers={
            'token': 'UKtGEIDTENETDdmCZdsafPPIgLoHXTpX'
        },
        params=payload
    )
```

```
import datetime
```

```
from IPython import display
```

```
current = datetime.date(2018, 1, 1)
end = datetime.date(2019, 1, 1)
```

```
results = []
```

```
while current < end:
```

```
    display.clear_output(wait=True)
    display.display(f'Gathering data for {str(current)}')
```

```
    response = make_request(
        'data',
        {
            'datasetid' : 'GHCND',
            'locationid' : 'CITY:US360019',
            'startdate' : current,
            'enddate' : current,
            'units' : 'metric',
            'limit' : 1000
        }
    )
```

```
    if response.ok:
        results.extend(response.json()['results'])
```

```
    current += datetime.timedelta(days=1)
```

```
    'Gathering data for 2018-12-31'
```

```
import pandas as pd
```

```
df = pd.DataFrame(results)
df.head()
```

	date	datatype	station	attributes	value
--	------	----------	---------	------------	-------

df.to_csv('data/nyc_weather_2018.csv', index = False)

```
import sqlite3
```

```
with sqlite3.connect('data/weather.db') as connection:
    df.to_sql(
        'weather', connection, index = False, if_exists = 'replace'
    )
```

```
response = make_request(
    'stations',
    {
        'datasetid' : 'GHCND',
        'locationid' : 'CITY:US360019',
        'limit' : 1000
    }
)
```

```
stations = pd.DataFrame(response.json()['results'])[['id','name','latitude','longitude','elevation']]
stations.to_csv('data/weather_stations.csv', index=False)
```

```
with sqlite3.connect('data/weather.db') as connection:
    stations.to_sql(
        'stations', connection, index=False, if_exists = 'replace'
    )
```