Using Geostreams API with Python

Using Python Requests

Get all sensors and save as csv

Get All Sensors

```
import requests
import json
import csv
from csv import DictWriter

api_server = r"https://greatlakestogulf.org/geostreams"
output_directory = r"downloads"

sensors = requests.get(api_server + "/api/sensors").json()["sensors"]

with open(output_directory + '/gltg_sensors.csv', 'w') as f:
    fieldnames = [
        'source','name','location', 'longitude', 'latitude', 'max_end_time', 'min_start_time',
        'parameters' , 'huc8', 'huc_name', 'online_status'
    ]

    writer = DictWriter(f, fieldnames=fieldnames)
    writer.writeheader()

    n_sensors = 0
    n_sensors_pos = 0
    for sensor in sensors:
        n_sensors += 1
        parameters_list = []
        for param in sensor['parameters']:
            if param in ['owner','source','unit_code']:
                continue
            if param[-3:] != "-qc":
                parameters_list.append(param + ',\n')
        parameters = "".join(parameters_list)

        huc8 = None
        if 'code' in sensor['properties']['huc']['huc8']:
            huc8 = sensor['properties']['huc']['huc8']['code']
        else:
            huc8 = sensor['properties']['huc']['huc8']

        if len(parameters) == 0:
            n_sensors_pos += 1
            continue

        writer.writerow({
            "source": sensor['properties']['type']['title'],
            "name": sensor['name'],
            "location": sensor['properties'].get('location', ""),
            'longitude': str(sensor['geometry']['coordinates'][0]),
            'latitude': str(sensor['geometry']['coordinates'][1]),
            'max_end_time': sensor.get('max_end_time',''),
            'min_start_time': sensor.get('min_start_time',''),
            'parameters': parameters,
            'huc8': huc8,
            'huc_name': sensor['properties']['huc'].get('huc_name',''),
            'online_status': sensor['properties'].get('online_status","")
        })

print("Sensors skipped " + str(n_sensors_pos) + " of Sensors total " + str(len(sensors)))
```

Get Datapoints by Sensor ID

We request that a user not try to pull all datapoints concurrently. It is preferred that datapoints be pulled in series by sensor id.

Download JSON of datapoints by Sensor ID

```
import requests
import json

sensor_id = 22
api_server = r"https://greatlakestogulf.org/geostreams"
output_directory = r"downloads"
user = {'identifier': '***email***', 'password': '***password***'}

r = requests.post(api_server + '/api/authenticate', data=json.dumps(user), headers={'Content-Type': 'application
/json'})
print("Authentication status:", r.status_code, "for", api_server)
headers = {"x-auth-token": r.headers["x-auth-token"], "Content-Encoding": "application/json"}

route = api_server + "/api/datapoints?sensor_id=" + str(sensor_id)

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

with open(output_directory + '/datapoints_sensor_' + str(sensor_id) + '.json', 'w') as f:
    f.write(json.dumps(r.json(), indent=2))

print("Route: " + route)
print("Request Status:", str(r.status_code))
print("Number of datapoints:", len(r.json()))
print("Datapoint JSON saved to " + output_directory + '/datapoints_sensor_' + str(sensor_id) + '.json')
```

Download Datapoints as CSV by Sensor ID

```
import requests
import json

sensor_id = 22
api_server = r"https://greatlakestogulf.org/geostreams"
output_directory = r"downloads"
user = {'identifier': '***email***', 'password': '***password***'}

r = requests.post(api_server + '/api/authenticate', data=json.dumps(user), headers={'Content-Type': 'application
/json'})
print("Authentication status:", r.status_code, "for", api_server)
headers = {"x-auth-token": r.headers["x-auth-token"], "Content-Encoding": "application/json"}

route = api_server + "/api/datapoints?sensor_id=" + str(sensor_id) + "&format=csv"

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

with open(output_directory + '/datapoints_sensor_' + str(sensor_id) + '.csv', 'w') as f:
    f.write(r.text)

print("Route: " + route)
print("Request Status:", str(r.status_code))
print("Datapoint JSON saved to " + output_directory + '/datapoints_sensor_' + str(sensor_id) + '.csv')
```

Jupyter Notebook

Jupyter notebook example can be downloaded here geostreams_jupyter.ipynb