Script Demonstrating Sphere API Usage

This script fetches blocklog data from the Sphere API, processes it, and optionally syncs it to an S3 bucket or the influx db.

WARNING: Influx db support hasn't been tested.

Script Usage Guide

The script queries data from the `SphereApi` using both synchronous and asynchronous methods for a given time range and checks if both methods return the same amount and content of data.

It then splits the time range into two parts and queries the data for each part separately. It checks if the sum of the data from these two parts matches with the data queried for the whole time range.

At the end of the script, if `S3_BUCKET` and `INFLUXDB_BUCKET` variables are defined, it exports the queried data to an Amazon S3 bucket and an InfluxDB bucket respectively.

Installation

Before running the script, ensure you have the necessary Python libraries installed. You can install them using pip:

```
pip install pandas boto3 requests argparse influxdb_client
```

Configuration

1. **Sphere Configuration**: This is the API key for the Sphere API, defined in the script `sphereapi.py` as `API_KEY`. Replace the API key with your own.

2. **S3 Configuration**: If you intend to sync the data to an S3 bucket, specify the bucket name, AWS access key ID, AWS secret access key, and AWS region name. They are defined in the script `s3export.py` as `S3_BUCKET`, `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, and `AWS_REGION_NAME` respectively. Replace them with your own.

3. **Influx Configuration**: If you intend to sync the data to influxdb, specify the influx bucket name, influx url, influx token, and influx org. They are defined in the script `influxexport.py` as `INFLUXDB_BUCKET`, `INFLUXDB_URL`, `INFLUXDB_TOKEN`, and `INFLUXDB_ORG` respectively. Replace them with your own.

Running the script

Here's an example of how to run the script:

```
python download.py
```

Key Functions

- `create_dataframe(data, schema)`: Converts the data received from the API into a pandas DataFrame.
- `write_to_file(query_execution_id, data, column_names)`: Writes the data to a CSV file with the name as the query execution id.
- `getData(api, timestamp_start, timestamp_end)`: Queries data from the API synchronously.
- `getDataAsync(api, timestamp_start, timestamp_end)`: Queries data from the API asynchronously.
- `main()`: The main function that orchestrates the querying, testing, and exporting of data.