FINE DUST DATA VISUALIZATION BACKEND MANUAL

Load data from CSV-files into the Influx Database:

Recuirements:

- Your files are for **SDS011** or **DHT22** sensors.
- You started InfluxDB (**localhost**, port **8086**) and created the Database with the name **testdb.**
- You have a directory with the CSV-files. (Alernatively you can load your files into Backend/Archive/ directory

Usage:

- Create a **SensorManager** instance
- Call **LoadDataFromDirectoryToDB(string pathToDir)** function.

```
SensorManager sm;
sm.LoadDataFromDirectoryToDB("../Backend/Archive/");
```

Create backup of the Database

Recuirements:

- You started InfluxDB (**localhost**, port **8086**) and created the Database with the name **testdb.**
- The data in this database is created according to the description in the previous step

Usage:

- Crate a **Connection** instance
- Call **backup(QString startTime, QString endTime, QString frequence)** function
- Start and end time can have different precision
  - Day only: "2020-06-01"
  - Day and time: "2020-06-01T00:00:00Z"
- Frequency: how many values you want to backup per sensor
  - One value every *x* hours: "*x*h"
  - One value every *x* minutes: "*x*m"

```
Connection c;
c.backup("2020-06-01T00:00:00Z", "2020-06-07T22:00:00Z", "1h");
```

QUERIES Examples

```
SensorManager sm;
//Examples for valid queries
cout << sm.avgCityValue("Ober-Ramstadt", "P1", "2020-06-06", "2020-06-07") << endl;
cout << sm.avgCityValue("Dieburg", "P1", "2020-06-02", "2020-06-04") << endl;
cout << sm.avgCityValue("Frankfurt", "P2", "2020-06-07T05:00:00Z", "2020-06-07T07:00:00Z") << endl;
cout << sm.avgCityValue("Darmstadt", "humidity", "2020-06-01T05:00:00Z", "2020-06-02T05:00:00Z") << endl;

//examples for errors:
//Not existig city
//--> average value for the whole Germany is returned. Output: "The city was not found!"
cout << sm.avgCityValue("Gibts nicht", "P2", "2020-06-02", "2020-06-04") << endl;
//No sensor data in the time range/invalid time input/invalid measure
//--> -9999 is returned. Output: "Something went wrong!"
cout << sm.avgCityValue("Frankfurt", "P2", "1999-06-02", "1999-06-04") << endl;
cout << sm.avgCityValue("Frankfurt", "P2", "abc", "2020-06-04") << endl;
cout << sm.avgCityValue("Frankfurt", "Gibts nicht", "1999-06-02", "1999-06-04") << endl;
```

Last and average value of the specific sensor can also be received **directly from the Database** with the instance of the Connection class if you know the exactly coordinates or id of the sensor. (The same requirements as for creating backup data)

Methods:
double **getLastValue**(double lon, double lat, QString measure);
double **getLastValue**(int id, QString measure);
double **getAverageWithinTimeRange**(double lon, double lat, QString timeFrom, QString timeUntil, QString measure);

These methods are unused now. **Frontend only needs the SensorManager instance**/instances to query the data.