Urban flows observatory, Sheffield
Data request API

Author Patricio F. Ortiz

Date November 2019

The way to extract data from the Urban Flows Observatory (Sheffield) is through a web tool called the Data Extraction Tool (ufdex), which can be accessed via a URL. The arguments to **ufdex** are passed as key=value pairs appended to the URL and separated by an ampersand (&)

Two arguments are mandatory: the initial and final instants of the time extraction window. We call these values Tfrom and Tto (but other values may be added later.

Bothe Tfrom and Tto take the form:
YYYY-MM-DD[THH:MM:SS][+hh:mm]
Both are interpreted as Local time, but the zone can be included if necessary

Specifying these two arguments only would retrieve all data stored in the system within that time window. In other words, users don't need to know the content of our databases to get data from the system.

Example:
`ufdex?Tfrom=2019-11-05T07:11:13&Tto=2019-11-05T10:11:13`

Obviously the amount of data retrieved may be overkill, so **ufdex** system works by adding additional filters on top of the temporal ones.

Spatial (geographical) filters:

There are four ways to filter by spatial features:

- **spatial=zone** Zone around a reference point. It requires to define **midLat** and **midLon**, the latitude and longitude of a reference point on Earth. Longitude is positive to the East, Latitude is positive to the north, and they are both measured in sexagesimal degrees. The distance to the reference point in kilometres (**cRad**) needs to be specified.
  Example:
  `spatial=zone&midLon=-1.23&midLat=53.354&cRad=2.5`

- **spatial=bbox** A box delimited by longitudes (minLon, maxLon) and latitudes (minLat, maxLat) The same conventions apply to longitude and latitude
  Example:
  `spatial=bbox&minLat=53.2&maxLat=53.5&minLon=-1.65&maxLon=-1.3509`

- **spatial=los** Expects a list of site names (**bySiteLoc**= comma separated list of site names) and a radius (**ncRad**) expressed in kilometres which applies to all sites. This alternative is like a multiple **spatial=zone**, but one in which it is not necessary to enter the longitude and latitude manually.
  Example:
  `spatial=los&ncRad=0.8&bySiteLoc=LD0006,LD0013,LD0045`

In addition, it is possible to filter by the height above sea level of the sitesby specifying the minimum and maximum heights above sea level in metres (minHasl and maxHasl)

Example: `minHasl=34&maxHasl=100`

Other temporal filters

The nature of the data stored in a urban observatory is driven by human activity, e.g., air pollution, traffic, etc. Sometimes it is important to filter datasets for certain periods only. The following are valid filters:

By day of the week. It is possible to select only certain days of the week if the overall interval is long, for instance:

`Tfrom=2019-08-01T08:21:25&Tto=2019-11-05T23:59:59&byDOW=5:Sat,6:Sun`

Will retrieve data for Saturdays and Sundays from August 1st to November 5th, 2019

By specific Month. This options filters data for the specified months only, for instance:

`Tfrom=2018-01-01&Tto=2022-11-05&byMONTH=06:Jun,07:Jul`

Will retrieve data for the months of June and July between 2018 and 2022.

By time of the day. Retrieves data which has been obtained during one more intervals during a certain time. The idea is to be possible to easily retrieve data for rush hour only or any other period of interest. **byTOD** specifies the comma separated intervals: HH:MM-hh:mm.

Example:
`Tfrom=2019-10-01&Tto=2019-10-31&byTOD=06:00-09:00,16:00-18:30`

Shall retrieve data between the specified limits. Note that the specific times refer to Civilian (Local) time, in other words, what the watches read. For the month of October, most days civilian time will be in BST but a few will be in GMT.

Types of data requests

How the data is provided to the user is defined by the key **action**, and several options are available. The main exchange formats are **CSV** and **netCDF** (others may be implemented later)

A typical **CSV** files looks like this:

```
# Begin CSV table for pair LD0076|25588
# The Urban Flows Observatory Sheffield @ 2019-11-14 09:46:29.979460
# From: 2019-08-01T00:00:00
# To: 2019-10-31T23:59:59
# site.id: LD0076
# site.longitude: -1.35195031864 [deg]
# site.latitude: 53.3660190482 [deg]
# site.heightAboveSeaLevel: 34.4 [m]
# sensor.id: 25588
# sensor.heightAboveGround: 3.99 [m]
# sensor.distanceToWall: 0.1 [m]
# sensor.family: luftdaten
```

```
# sensor.detectors: luftdaten
# ColDescription: name / units / UCD / description / type / minVal /
maxVal / Avge / last / no-data-value
# Column_1 / time / s / TIME_UTC_UNIX / Time in seconds since 1970.0, or
UNIX time / utime / 1564615559 / 1572539701 / 1568828557 / 1572539701 /
-32768
# Column_2 / sensor /  / ID_MAIN / Sensor identifier / int / 25588 /
25588 / 25588 / 25588 / -32768
# Column_3 / AirTemp / C / MET_TEMP / Air temperature / float /
-0.649999976158 / 34.1500015259 / 13.7379506904 / 7.71999931335 / -32768
# Column_4 / RelHum / % / MET_RH / Relative humidity / float / -32768.0 /
100.0 / 80.7037727566 / 100.0 / -32768
# Column_5 / AtmPress / hPa / MET_AP / Atmospheric pressure / float /
-32768.0 /1033.77270508 / 1006.25595609 / 1015.00469971 / -32768
# Number of points 10833
1564615559,25588,17.39,86.65,1010.9787
1564615854,25588,17.46,86.62,1010.9405
1564616150,25588,17.42,86.7,1011.1122
1564616446,25588,17.48,86.53,1011.137
…
1572539701,25588,7.7199993,100.0,1015.0047
# End CSV table for pair LD0076|25588
```

Any line starting with a hash (#) is a comment or not part of the column data, but information is stored in each line, like the geolocation of the site described in this table.

netCDF format is implemented but it will experience changes, so it will be described in more detail once the changes are operational.

There are several options on how to get these CSV files.

&action=CSV_zip The format of the files is CSV, one file per site/sensor combination. This is probably the most efficient way to transfer data in an automated harvesting pipeline.

&action=CSV_show CSV format, shown on the browser. This is an option when one wants to examine a small amount of data.

&action=CSV_save Ditto, but the user is prompted to save the results on a file

&**action=json_META** Retrieve metaData related to the observatory sites and sensors

To be continued…