M. G. Schultz et al.

Tropospheric Ozone Assessment Report: Database and Metrics Data of Global Surface Ozone Observations

SUPPLEMENT 2: Documentation of the Jülich Open Web Interface for accessing TOAR surface ozone data

JOIN User Guide

https://join.fz-juelich.de

The JOIN web application has been launched as a testbed for developing automated, "interoperable" analysis of atmospheric data sets across platforms (i.e. measurement platforms and numerical models). It began as an activity under the Task Force on Hemispheric Transport of Air Pollution (TFHTAP) with the original intention to establish a node of the US-based Datafed system (http://www.datafed.net/). However, it soon became clear that a more fundamental development was needed in order to realize the vision that scientists anywhere in the world can easily find, retrieve, and work with (geospatial) data without having to possess expert knowledge in data repositories, data formats, etc.. JOIN is still at an early stage of its development. Seizing the opportunity to provide a real application as a web interface to TOAR data, the development of JOIN over the past two years focused on providing an interface to the TOAR surface ozone database. Its main purpose is to allow researchers to search for and identify surface ozone datasets, to visualize various aspects of these datasets, and to allow download of ozone metrics in an easy and user-friendly way.

JOIN provides access to TOAR data in two ways: (i) as graphical web interface, and (ii) as REST services. The first method allows researchers to easily explore the vastness of the TOAR surface data collection, while the second method constitutes a machine-readable interface which can be integrated in your own applications or web services.

This documentation only describes those parts of JOIN which are relevant to TOAR surface ozone data. As there is no full publication on JOIN available yet, please refer to JOIN by citing this supplement ("JOIN User Guide - Supplementary material to Schultz, MG, et al.: …" as given in the title of this document).

If you make use of TOAR data products obtained through JOIN, we request that you acknowledge TOAR as follows:

> "[We thank] the Tropospheric Ozone Assessment Report (TOAR) initiative for providing the surface ozone data [*and/or:* analysis] shown in [*or:* used by] this report [*or:* publication]."

Also, please include a reference to the TOAR Database and Metrics Data paper:

Schultz, MG, et al. Tropospheric Ozone Assessment Report: Database and Metrics Data of Global Surface Ozone Observations, Elem. Sci. Anth. 2017. https://doi.org/10.1525/elementa.244 .

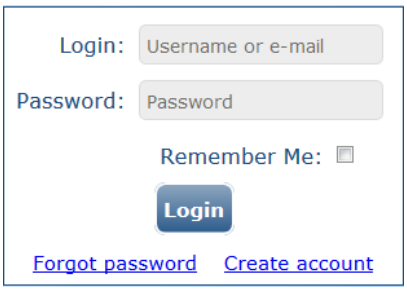
Contents:

1. Accessing the JOIN graphical web interface

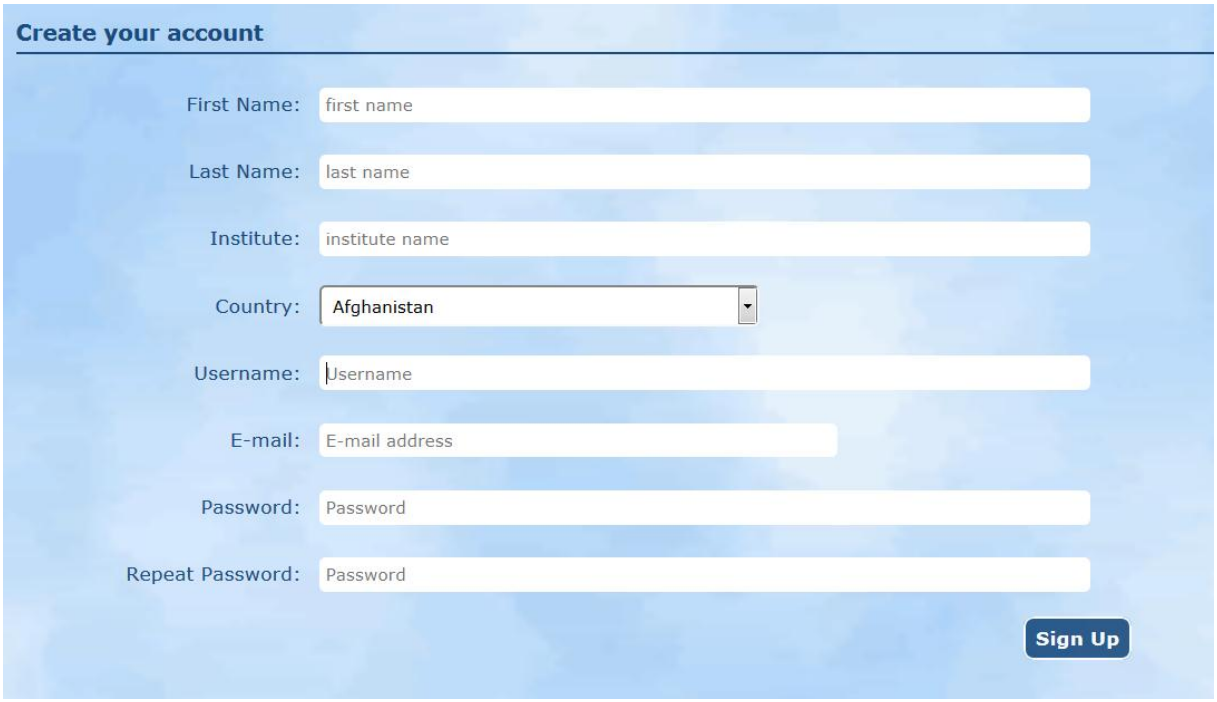
The documentation of the graphical web interface is provided in the form of “Frequently asked questions”:

1.1. How do I access JOIN?

If you visit the JOIN homepage at https://join.fz-juelich.de (or any URL within this domain), you will see a very plain homepage and a login box on the top right of the page:



If you do already have a JOIN account, use your login name and password to access the system. Otherwise, click on the "Create account" link on the lower right of this box and you will see the user registration form:



Please fill in at least your first name, last name, user name, email address and password, then click on the Sign up button. JOIN will send you an email with a verification link. You need to verify your registration before you can use the system. Once you have verified your email address, you can log in to JOIN and start visualizing your data.

1.2. Why do I have to register in order to use JOIN?

Honestly, we would have loved to grant also anonymous access to JOIN, but we didn't find a suitable technical solution for this. A JOIN user automatically accesses some hardware resources on our server for temporary files etc., and we need to be able to relate such data to individual users (otherwise person A's click would influence person B's display). Another issue here is internet security as we need to get past certain security tests before we are allowed to open the web interface to the public. We will further investigate technical options to allow anonymous access, but we ask you to please accept that you need to register in order to use JOIN. By offering JOIN as https service, we protect your personal data from misuse.

Another reason why we ask you to register is to allow us to build a use statistics. This kind of information (how many users do we have? How often is JOIN being used?) is very relevant for funding agencies and might help to secure the future development of the system.

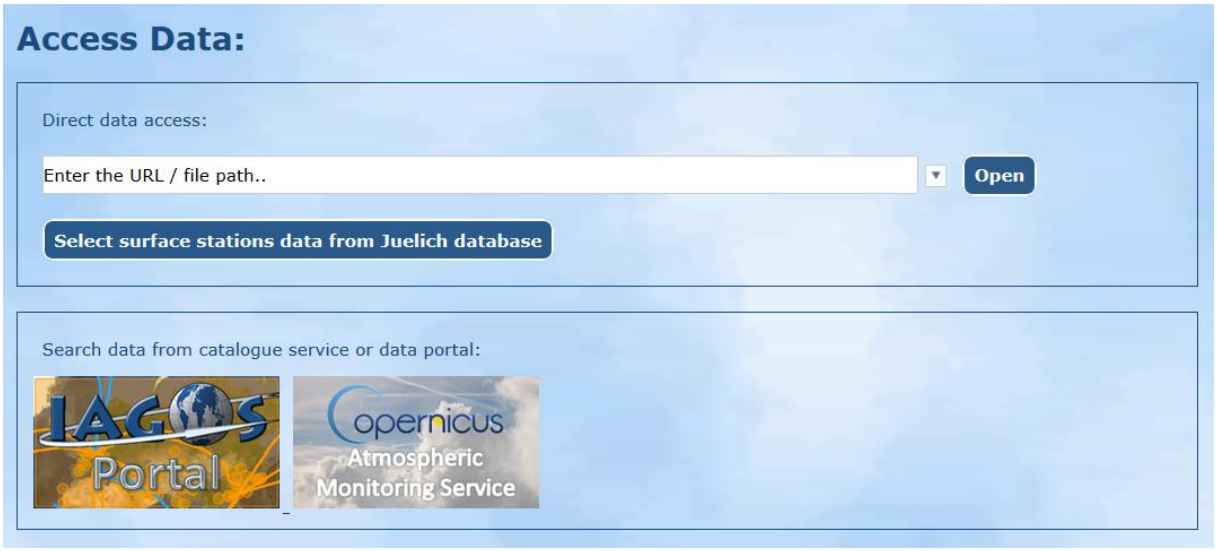
1.3. How do I change my user profile data?

Once you are logged in, you will see a menu bar appear which contains your user name at the right end:

If you click on the small arrow to the right of your name, a menu will fall down which allows you to change the settings of "My account", modify "User settings" (to be implemented), or log out. Click on "My account" to access and modify your account information, including your password [password change to be added soon].

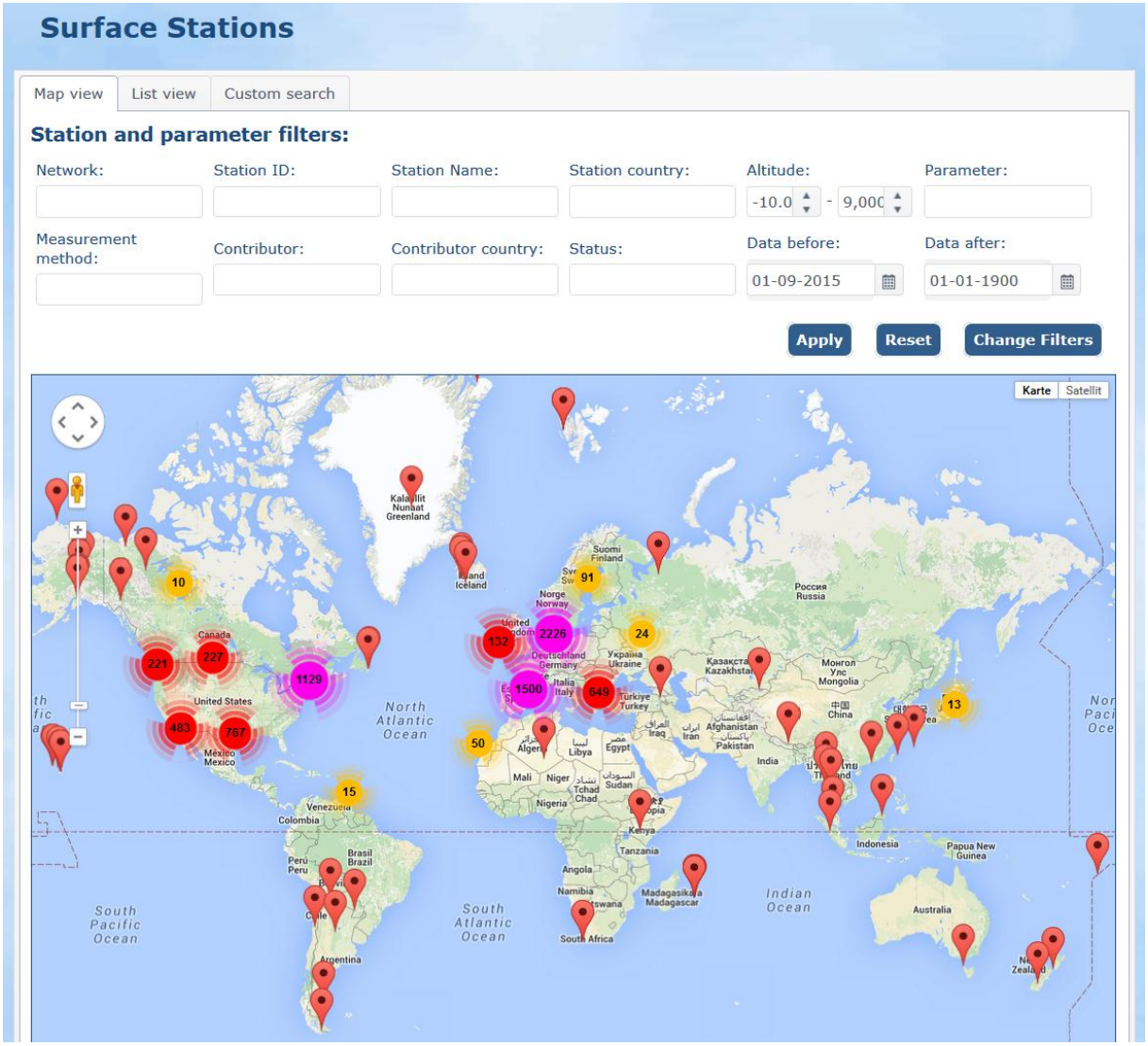
1.4. How do I access data from the TOAR database of surface ozone observations?

From your user homepage you can access all data resources by clicking on the "Data access" menu item (in fact, at present the user home page is re-directed to the data access page). On the data access page you will find a section for "Direct data access" on the top, and another section to find and retrieve data through other portal pages on the bottom. The "Direct data access" section provides a dropdown box with pre-installed data files (rudimentary feature - this shall allow the upload of user data files in the future), and a button to "Select surface stations data from Jülich database".



Click on the “Select surface stations from Juelich database” button, and a map will appear which contains one marker for each surface site we have in our database. This is a normal google map, so you can zoom in or out and change to satellite view if you want.

On top of the map are a couple of filters which you can use to limit the selection of stations offered on the screen. These filters are configurable: you can add or remove filters by clicking on the "Change filters" button.

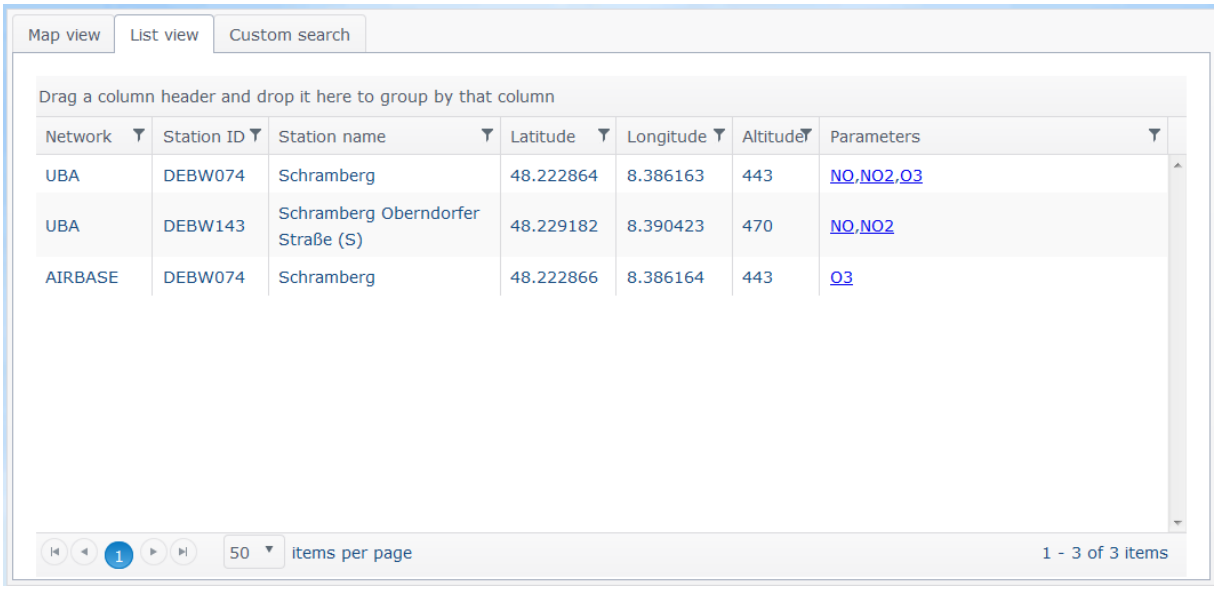


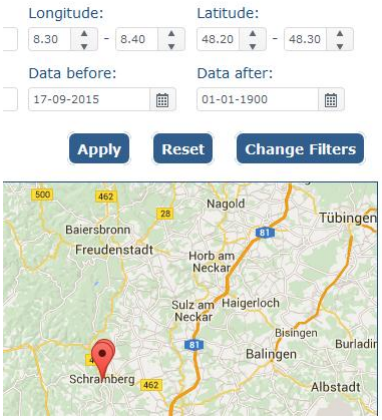
Once you found your station, you can click on the marker. This will display the station name as a link that points you to the station and time series metadata, the coordinates, and a list of parameters available from this site. Click on any parameter to get the data extraction dialog which allows you to display the hourly data of the chosen parameter at the selected site, a data summary plot, or some daily, monthly, or annual statistics. In case of ozone (parameter "O3") you will find additional options to plot or download various ozone metrics that have been defined in the TOAR initiative. Note that we do not make the hourly time series data available for download. Please consult the respective data providers (see time series metadata) if you are interested in the original data.

1.5. I have clicked on the marker of a station and I see the wrong dataset. What shall I do?

We have organized the data in our database according to the network to which they contributed. In some cases, station PIs submit their data to various networks (and unfortunately, these data and metadata are not always identical). In our database we then have two or more station entries at the same location, and google maps will allow to click on only one of these. For example, station DEBW031 is registered in both the "UBA" and the "Airbase" networks. The marker on the map only shows this site as UBA station.

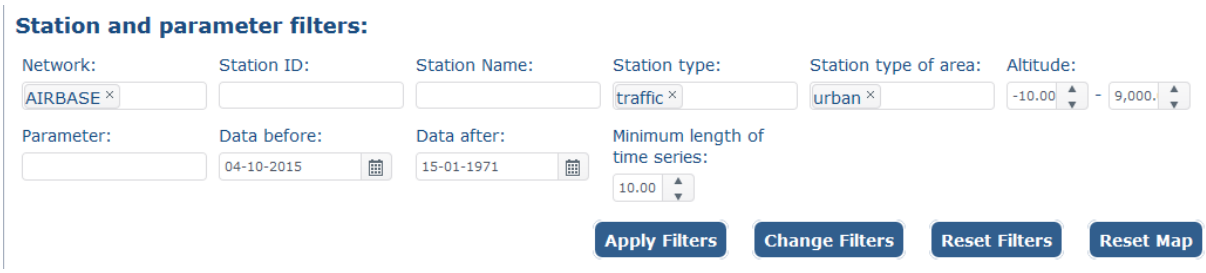
You can switch to the list view (second tab on the tab menu on top of the "Station and parameter filters", see image below) in order to see all stations. If you combine this with a station selection through filtering (here use the "station id" filter and enter "DEBW031", then click "Apply"), you will see only the stations you are interested in. We recommend to use this feature with the longitude and latitude filters (see example below), so that you don't miss stations which are listed under different station IDs or where the coordinate information is not precisely identical.





1.6. What is the function of the “Station and parameter filters”?

The Juelich surface station database uses a comprehensive, harmonized set of metadata to describe each measurement site and each measurement (“parameter”) made at this site. We defined an extensive set of filters which you can flexibly combine in order to identify a specific set of stations (or an individual station), and/or a specific parameter set. Note that the majority of data in the database are presently surface ozone observations collected for the TOAR initiative. Up to 12 filters can be selected and used at any time. If you need more, you can use the “Custom search” tab instead. The figure below shows an example of filter settings, which selects 168 (ozone) data series at 168 stations from the European Airbase network, which are marked as “urban traffic” sites and have at least 10 years of data. All filters are combined as logical AND.



Here is a brief description of the meaning and function of each filter:

Network: The short name of the network under which the station is stored. Available networks are: Airbase (Europe), Airmap (US), AQS (US), AUSAQN (Australia), CAPMon (Canada), CASTNET (US), EANET (Asia), EMEP (Europe), GAW (global), ISRAQN (Israel), KRAQN (Korea), NAPS (Canada), NIES (Japan), Other (individual research or country stations), RSA (South Africa), UBA (Germany). This filter works as dropdown box; multiple selections are possible.

Station ID: The station identifier or the beginning of a station identifier. Example: to find all German stations in Bavaria in the Airbase or UBA networks, you can enter “DEBY”. As this search is case-insensitive, “deby” will also work.

Station Name: The string you enter in this filter will be evaluated as regular expression against all available station names. For example, if you enter “heim”, all stations containing “heim” in their name will be found. “^Bratt” will find all stations beginning with “Bratt”, and “Lake$” will find all stations ending with “Lake”.

Station type: A station characterization according to the EU Airbase classification. Available categories are “background”, “industrial”, “traffic”, “other”, and “unknown”. Since not all networks maintain such station classification, the information may not be robust to identify

all global “background” stations, for example. This will require further work on the database. This field is a dropdown box with multiple selections possible.

Station type of area: Another EU Airbase station characterization labeling the surrounding area of a station. Available categories are “remote”, “rural”, “suburban”, “urban”. Note that “remote” was added to the existing Airbase classifiers. This field is a dropdown box with multiple selections possible.

Station category: Originally meant for GAW station categories, this field also stores landuse information from other databases where it is not straightforward to map these onto the “station type” and “station type of area” fields. This field is a dropdown box with multiple selections possible.

Station country: The country in which the station resides. Note that country names have been harmonized in case different spelling were used in the various original databases. This field is a dropdown box with multiple selections possible.

Station State: The state (or province) in which a station resides. This field is a dropdown box with multiple selections possible.

Longitude: Two numeric text boxes allow specification of a longitude range for filtering (together with a latitude range this constitutes a “bounding box”). Allowed values are from -180 to +180. If the first value is larger than the second one, the search will “wrap” around the dateline. For example: “90” to “-70” will find all stations east of 90 degrees east and west of 70 degrees west.

Latitude: Two numeric text boxes to select a latitude range. Allowed values are from -90 to +90, and the first value must be lower (or equal) than the second one. Negative latitudes are in the Southern hemisphere.

Altitude: An altitude range in metres. Note that there is no exact definition how station altitudes should be provided (inlet line, ground, …). In order to capture all stations, choose “-10” as lower range (there are 12 stations with altitudes below zero in the database) and “9000” as upper range (in case someone builds a station on top of Mt. Everest).

Minimum relative altitude: A value in metres describing the minimum altitude difference between the station and the lowest point in the surrounding 5 km (based on ETOPO1 altitude data).

Maximum relative altitude: As above, but describing the maximum altitude difference.

HTAP region: Select a geographic region as defined in the Task Force on Hemispheric Transport of Air Pollution (TFHTAP); see http://iek8wikis.iek.fz-juelich.de/HTAPWiki/WP2.1, tier 1 regions.

Climatic zone: Select a climatic zone according to the IPCC (2007) Classification scheme for default climate regions.

Dominant landcover: Select a landcover type which dominates at the station location; see https://lpdaac.usgs.gov/dataset_discovery/modis/modis_products_table/mcd12c1

Nighttime light intensity: Choose a range of brightness index values from the DMSP satellite dataset on "stable nighttime lights"; see http://ngdc.noaa.gov/eog/dmsp/downloadV4composites.html.

Population density: choose a range of population density values at the station location; see http://sedac.ciesin.columbia.edu/data/collection/gpw-v3

Y2010 NOx emissions: Choose a range of NOx emission values at the station location; see http://edgar.jrc.ec.europa.eu/htap_v2/index.php?SECURE=123)

Y2010-2015 OMI NO2 column: Choose a range of NO2 column values in which the station must fit; see Krotkov et al., 2016.

Y2000 rice production: Choose a range of annual rice production (in tons) at the station locations.

Y2000 wheat production: dto. For wheat production.

Parameter: A dropdown list to select one or several parameters to search for. Available parameters are "humidity", "no", "no2", "o3", "pm10", "press", "so2", "temp", "wdir" and "wspeed". Note, however, that presently almost all data series only contain surface ozone data. Other parameters are only available for the UBA network.

Parameter attribute: There are some stations which have submitted more than one data series of the same parameter. For example, the GAW station Cape Point, South Africa has a filtered and an unfiltered ozone series. In order to distinguish these, the database allows the addition of an attribute to the parameter name. You can use this filter to specifically select such a dataset. There is no standardization of attribute names and not many data series make use of attributes.

Status: Currently not used (all data sets have a status of "0" meaning "OK"). In the future this field will be used to identify various aspects of the dataset such as "questionable" (if there are unexplainable features in the dataset found while screening of data or comparison with other datasets), or "TOAR dataset" to identify datasets which are used in the TOAR surface ozone analyses.

Data before: A date field to search for datasets which contain data before the given date. For example, if you wish to identify stations which measured ozone before 1990, you can select "o3" as parameter and "01-01-1990" as "Data before". Note that this doesn't imply that there are no data after the selected date, only that there are at least some measurements before.

Data after: Similarly to “Data before”, this field allows to select datasets which extend at least up to a certain date. For example, in order to find all stations which have data in 2013 or later, select “01-01-2013” as “Data after”. Combined, these two filters allow you to select for example all stations which have data throughout the 1990s: “Data before 01-01-1990” and “Data after 31-12-1999”.

Minimum length of timeseries: This filter allows to select datasets with a given minimum data series length, regardless of when these data were sampled. The value is interpreted as length in years. Note that periods that are missing in between are not evaluated so that a station which would have measurements in, say, 2000, no data in 2001, and then again one year of data in 2002 would still be counted as “minimum length” of 3 years.

Sampling type: Currently, all data in the database are of sampling type “continuous”. This may change at some point when flask sampled data are added.

Measurement method: A description of the measurement method as provided in the original metadata. This is not harmonized so that there are, for example, about 10 different descriptions of “UV absorption” for ozone instruments.

Contributor: The name of the institution who contributed this dataset if this is given in the original dataset metadata.

Contributor shortname: A shortname (abbreviation” of the contributing institution.

Contributor Country: The country where the contributing institution resides.

Principial investigator: A few datasets contain the name of the PI who is responsible fir the dataset. If given this field can be used to select data from a specific PI.

Type of dataset: Currently, all datasets in the Juelich database are of type “hourly”.

Created before: Find all datasets which were entered into the Juelich database before a given date.

Created after: Find all datasets which were entered into the Juelich database after a given date.

Modified before: Find all datasets which were modified in the Juelich database before a given date. Note that minor changes in metadata will not influence the data modification date.

Modified after: Find all datasets which were modified in the Juelich database after a given date. Note that minor changes in metadata will not influence the data modification date.

1.7. How can I plot an ozone timeseries?

From the map view, select your station and click on the marker. In the small popup you should see a link labeled "O3". Alternatively, you should see a similar link on the list view window next to your station of interest. Click on this link, and a dialog box (see picture below) will open, where you can fine tune the data extraction and select the type of plot you want. You can narrow the date range of the data you wish to show (note that shorter timeseries display faster; a 30 year ozone record comprises ~260,000 hourly data values which need to be processed in Javascript!). The default date range always covers the entire dataset range. You can choose if you wish the data to be shown as local time (default), UTC time, or solar time. Solar time is the "true" time at the station location, i.e. longitude/15. Local time in JOIN ignores daylight savings time and will in most cases be the same as solar time.

Below these "data extraction options" you will see a couple of buttons to either plot or download certain "views" of the data. Note that hourly data can only be plotted and not downloaded. Furthermore, display of hourly data is limited to a maximum of one year at a time.

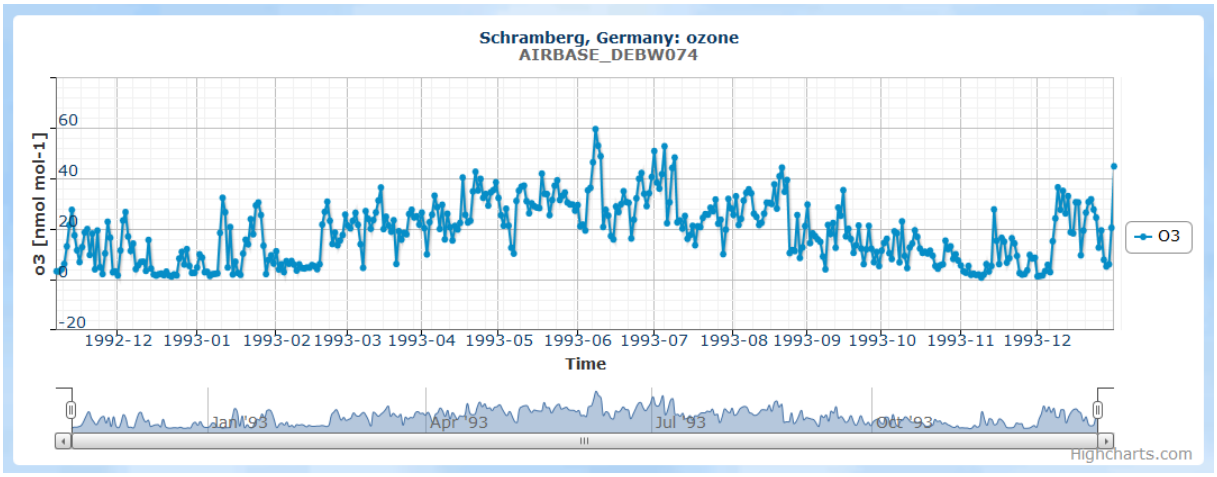
The "Data summary plots" are standardized multi-panel plots, which show various statistics of a dataset, such as seasonal, weekly, and diurnal cycles, frequency distributions, and a preliminary trend analysis using Theil slopes (these trends are "preliminary", because TOAR might decide on some outlier filtering or other filtering options before the final trend analysis). Note that “seasons” are always defined according to the boreal hemisphere (DJF: winter, MAM: spring, JJA: summer, SON: fall). Various statistics and ozone metrics which have been defined in TOAR are available as either daily, monthly, or annual products.

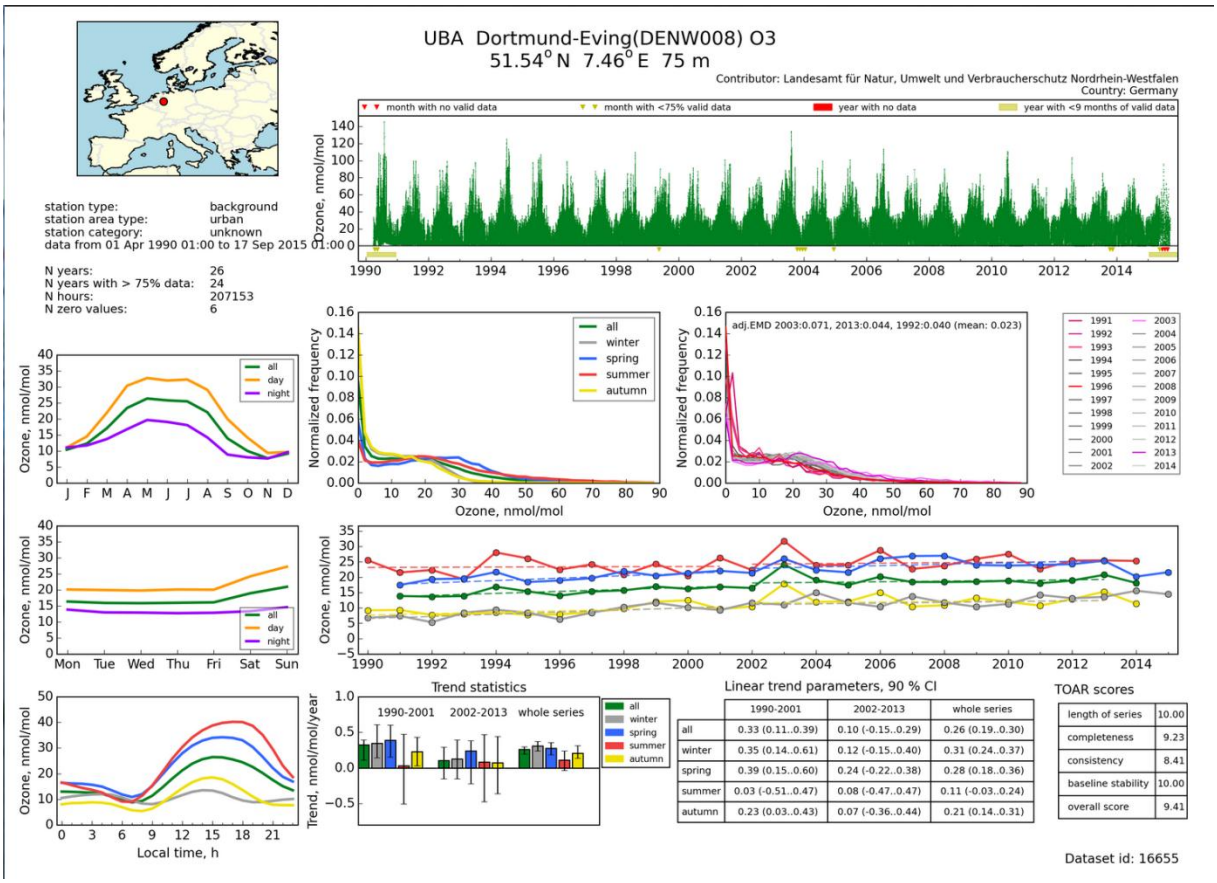
Data summary plots and metrics are calculated online if you access them through JOIN. This means that you may have to wait a little until you see results, but it offers you the chance to calculate metrics on subsets of data as you wish. Also note that currently, all statistics and

metrics (and the data summary plots) always use “solar time” as reference (TOAR has defined some metrics to be based on “local time” instead; for the most part the two should be identical).

Data summary plots and statistics can be produced for all variables in the database except wind direction (where simple averaging makes no sense). Ozone metrics are only available for surface ozone records.

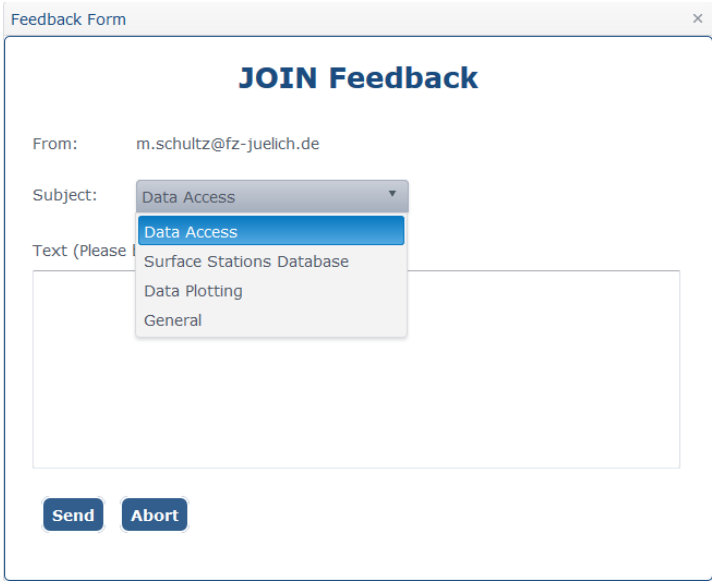
An example timeseries plot and a data summary plot are shown below.





1.8. What do I do if something doesn't work or if I detect errors in the data of the TOAR database?

If you are logged into JOIN you will see a Feedback link on the bottom right of the page. Click there, and a dialog will open up which you can use to send us an email about your problem. Please use the selector to identify your issue as belonging to one of the categories "Data access", "Surface stations database", "Data plotting", or "General". In your message, please try to be specific and describe under which conditions the error showed up.



Of course you can also use the feedback form to request additional features or tell us if you like the web interface.

2. Documentation of the JOIN REST interface

A Representational State Transfer (REST) service allows querying all metadata and data products from the Tropospheric Ozone Assessment Report (TOAR) database of surface ozone observations. This database is described in Schultz et al. (2017a). The main TOAR data products are available as supplement to that article on PANGAEA (Schultz et al., 2017b). Note that the *online* database, which is queried via the REST service, differs from the database described in Schultz et al. (2017) as more data have been added since writing the manuscript. The primary purpose of the REST interface described here is machine-machine communication, i.e. the inclusion of TOAR data into other web services such as JOIN. However, you can also employ the TOAR REST services for specific queries of the database or in your data analysis software. An example script in python is given below.

This documentation describes the URL architecture and query options of the TOAR REST interface. For general information on REST, please consult other resources.

References:

Schultz, M. G. et al. (2017a) Tropospheric Ozone Assessment Report:Database and Metrics Data of Global Surface Ozone Observations, Elementa. [issue, page range, and doi to be added]

Schultz, M. G. et al. (2017b) Tropospheric Ozone Assessment Report:Global Surface Ozone Data Products, supplement to Schultz et al. (2017a), https://doi.org/10.1594/PANGAEA.876108.

1. General

1.1 Base URL

https://join.fz-juelich.de/services/rest/surfacedata/

Response: Description and documentation of available REST services (this document)

1.2 Services

The following information services are available and described individually below. Each service is invoked by appending its name and possible query arguments to the base URL.

- **parameters:** query the parameter (i.e. variable) names and properties of the database
- **networks:** query the observation networks and their description from the database
- **stations:** query station ids, station names, and station location from the database
- **series:** query the data series id and specific metadata of a series from the database
- **search:** execute a flexible database search query on stations and data series
- **stats:** retrieve TOAR data products for a selected data series

1.3 Query arguments

In order to control the database queries and hence the response of the TOAR REST service, you can add arguments to the service URL. These arguments must adhere to the format argumentname=value. The first argument is prepended by a ?character, all other arguments are separated by & characters.

Example:https://join.fz-juelich.de/services/rest/surfacedata/stations/?station_country=Germany&parameter_name=o3

For some arguments, multiple values are allowed. According to the REST standard, these should be specified by repeating the argumentname for each value. However, for convenience it is also possible to provide multiple argument values as comma-separated list. Note that for arguments, where only one value is allowed, the service will only use the last value and silently ignore all other values. This is a common feature of REST services.

Example: https://join.fz-juelich.de/services/rest/surfacedata/stations/?parameter_name=o3&parameter_name=no2

or (non-standard): https://join.fz-juelich.de/services/rest/surfacedata/stations/?parameter_name=o3,no2

Date values must be supplied as string with the format YYYY-MM-DD hh:mm. Note that the blank character is escaped as %20 in http URLs. You can write blanks in the address line of your browser, but when you copy a URL you will see that blanks are replaced by %20. Do not use quotes around a date string (or in fact any other string argument).

Value ranges can be specified by including the first and last value in square brackets, separated by comma.

Example:https://join.fz-juelich.de/services/rest/surfacedata/stats/?id=21919&daterange=[2010-01-01 00:00,2016-12-31 23:00]

1.4 Response format

The default response format is json. You can control the format with the format= option in all queries. Currently, only json and html are supported.

1.5 Error messages

The REST service may return a page with error code 500 if you try to open a malformed URL. Usually, a meaningful error message shall be returned in this case.

Note that queries which are formally correct, but return no results, return a valid page (HTML code 200) with empty content. If the response format is json, you will typically receive an empty array [] in this case.

2. Description of services

2.1 Parameters

https://join.fz-juelich.de/services/rest/surfacedata/parameters/[?QUERY-OPTIONS]

where QUERY-OPTIONS are:

name = <string> (examples: o3, ox, co, no, no2, nox, ch4, pm2.5, pm10, temp)

format = <string> (json|html)

metadata = <Boolean> (True|False; default: False)

Multiple parameter names can be given by repeating the *name=* option or as a comma-separated list.

Response: By default, the query will return a list of parameter names. If *metadata=True*, the complete parameter metadata will be returned as lists with field names parameter_name, parameter_long_name, parameter_display_name, parameter_cf_standard_name, parameter_units, parameter_formula.

If no QUERY-OPTIONS are given, the complete set of parameter names will be returned in json format.

Results are ordered alphabetically by parameter_name.

Example:https://join.fz-juelich.de/services/rest/surfacedata/parameters/

Result:

["albedo", "aswdifu", "aswdir", "benzene", "ch4", "cloudcover", "co", "ethane", "humidity", "no", "no2", "nox", "o3", "ox", "pblheight", "pm1", "pm10", "pm2p5", "press", "propane", "relhum", "so2", "temp", "toluene", "totprecip", "u", "v", "wdir", "wspeed"]

Example: https://join.fz-juelich.de/services/rest/surfacedata/parameters/?name=o3&name=no&name=ethane&format=json&metadata=True

or: https://join.fz-juelich.de/services/rest/surfacedata/parameters/?name=o3,no,ethane&format=json&metadata=True

Result:

[["ethane", "Ethane", "Ethane", "mole_fraction_of_ethane_in_air", "nmol mol-1", "C2H6"],["no", "nitrogenmonoxide", "NO", "mole_fraction_of_nitrogen_monoxide_in_air", "nmol mol-1", "NO"],["o3", "ozone", "Ozone", "mole_fraction_of_ozone_in_air", "nmol mol-1", "O3"]]

2.2 Networks

https://join.fz-juelich.de/services/rest/surfacedata/networks/[?QUERY-OPTIONS]

where QUERY-OPTIONS are:

name = <string> (examples: GAW, UBA, CAPMON, EANET)

format = <string> (json|html)

metadata = <Boolean> (True|False; default is False)

Multiple network names can be given by repeating the *name=* option (URL standard) or as comma-separated list.

Response: If no QUERY-OPTIONS are given, the complete set of network names will be returned as a list in json format. If *metadata=True*, the response will consist of lists with the field names network_name, datacenter_name, datacenter_fullname, datacenter_url.

Example:https://join.fz-juelich.de/services/rest/surfacedata/networks/

Result:

["AIRBASE", "AIRMAP", "AQS", "CAPMON", "CASTNET", "EANET", "EMEP", "GAW", "ISRAQN", "NAPS", "NIES", "OTHER", "UBA"]

Example:https://join.fz-juelich.de/services/rest/surfacedata/networks/?name=UBA&metadata=True

Result:

[["UBA", "Federal Environment Agency", "German Federal Environment Agency", "http://www.umweltbundesamt.de/en/data/current-concentrations-of-air-pollutants-in-germany"]]

2.3 Stations

https://join.fz-juelich.de/services/rest/surfacedata/stations/[?QUERY-OPTIONS]

where QUERY-OPTIONS are:

network_name = <string> (examples: GAW, UBA, CAPMON, EANET)

station_id = <string> (examples: DENW058, CGO540S00, CVO, 13-089-0002)

station_name = <string: regular expression> (examples: MOUNT PEARL, New Y)

station_country = <string> (examples: Germany, United Kingdom, France)

... and many other keywords as described in section 3 below.

format = <string> (json|html)

as_dict = <Boolean> (True|False; default is False). Only applicable if format is json.Returns results as json dictionaries instead of lists.

Multiple argument values can be given by repeating the respective query option. Alternatively, you can provide multiple arguments as comma-separated list.

Response: Each query result consists of the fieldsnetwork_name, station_id, station_name, station_lon, station_lat, station_alt. If you wish to retrieve other fields of station metadata, you can use the **search** service (section 2.5) with the columns argument.

If no QUERY-OPTIONS are given, the complete set of stations will be returned.

Example: https://join.fz-juelich.de/services/rest/surfacedata/stations/

Result:

[["ISRAQN", "ILA02RB", "ariel", 35.168563, 32.103376, 546.0], ["GAW", "ANG638N00", "Angra do Heroismo", -27.22, 38.67, 74.0], ["GAW", "CAS639N00", "Castelo Branco", -7.47, 39.83, 386.0], ["GAW", "CGO540S00", "Cape Grim", 144.689938889, -40.6831194444, 94.0], . . .]

Example: https://join.fz-juelich.de/services/rest/surfacedata/stations/?parameter_name=o3&parameter_name=no2

or: https://join.fz-juelich.de/services/rest/surfacedata/stations/?parameter_name=o3,no2

The output will look similar, but here only stations which have an ozone **or** NO2 data series will be returned. Note that it is not possible to query the database for stations which have ozone **and** NO2 data. If you need this functionality you must perform separate queries for parameter_name=o3 and parameter_name=no2 and combine the two search results in your application.

Example: https://join.fz-juelich.de/services/rest/surfacedata/stations/?station_name=Cape

Result:

[["GAW", "CGO540S00", "Cape Grim", 144.689938889, -40.6831194444, 94.0], ["EANET", "GAWCOI", "Cape Ochiishi", 145.5, 43.15, 49.0], . . .]

Example: https://join.fz-juelich.de/services/rest/surfacedata/stations/?station_htap_region=EAS,SEA&station_toar_category=urban

Returns records of all stations in regions EAS or SEA (East Asia and South East Asia) which are classified as urban according to the TOAR station category.

Example: https://join.fz-juelich.de/services/rest/surfacedata/stations/?station_htap_region=EAS,SEA&station_toar_category=urban&as_dict=True

Returns the same result as the previous example, but as list of json dictionaries:

[{"network_name": "NIES", "station_id": "jp01101010", "station_name": "Senta", "station_lon": 141.3539, "station_lat": 43.0619, "station_alt": 19.0}, {"network_name": "NIES", "station_id": "jp14201020", "station_name": "Oppamagyouseisenta", "station_lon": 139.6319, "station_lat": 35.3183, "station_alt": 3.0}, {"network_name": "NIES", "station_id": "jp01102010", "station_name": "Shinoro", "station_lon": 141.3714, "station_lat": 43.1472, "station_alt": 4.0}, . . .]

Example: https://join.fz-juelich.de/services/rest/surfacedata/stations/?altitude=1000.&vtol=500.

The output looks similar to the one from the previous examples, but only stations with a station_alt between 500 and 1500 m will be returned (see description of the search options in section 3).

2.4 Series

https://join.fz-juelich.de/services/rest/surfacedata/series/[?QUERY-OPTIONS]

where QUERY-OPTIONS are:

parameter_name = <string> (examples: o3, no2, temp)

network_name = <string> (examples: GAW, UBA, CAPMON, EANET)

station_id = <string> (examples: DENW058, CGO540S00, CVO, 13-089-0002)

station_name = <string: regular expression> (examples: MOUNT PEARL, New Y)

station_country = <string> (examples: Germany, United Kingdom, France)

... and many other keywords as described in the search query (section 2.8).

format = <string> (json|html)

as_dict = <Boolean> (True|False; default is False). Only applicable if format is json.Returns results as json dictionaries instead of lists.

Multiple argument values can be given by repeating the respective query option. Alternatively, you can provide multiple arguments as comma-separated list.

Response: Each query result consists of the fieldsseries_id, network_name, station_id, parameter_label.

If no QUERY-OPTIONS are given, the complete set of stations will be returned.

Example:https://join.fz-juelich.de/services/rest/surfacedata/series/?station_name=Bay

Result:

[[22119, "CAPMON", "CAPMCANL1GOS", "O3"], [28469, "NAPS", "010601", "O3"]]

Example:https://join.fz-juelich.de/services/rest/surfacedata/series/?station_name=Bay&as_dict=True

Result:

[{"id": 22119, "network_name": "CAPMON", "station_id": "CAPMCANL1GOS", "parameter_label": "O3"}, {"id": 28469, "network_name": "NAPS", "station_id": "010601", "parameter_label": "O3"}]

Example:https://join.fz-juelich.de/services/rest/surfacedata/series/?parameter_name=temp

Result:

[[16650, "UBA", "DENW021", "TEMP"], [16698, "UBA", "DENW059", "TEMP"], [16639, "UBA", "DENW067", "TEMP"], . . .]

Example:https://join.fz-juelich.de/services/rest/surfacedata/series/?parameter_name=temp&as_dict=True

Result:

[{"id": 16650, "network_name": "UBA", "station_id": "DENW021", "parameter_label": "TEMP"}, {"id": 16698, "network_name": "UBA", "station_id": "DENW059", "parameter_label": "TEMP"}, . . .]

2.5 Search

https://join.fz-juelich.de/services/rest/surfacedata/search/[?QUERY-OPTIONS]

where QUERY-OPTIONS are:

network_name = <string> (examples: GAW, UBA, CAPMON, EANET)

station_id = <string> (examples: DENW058, CGO540S00, CVO, 13-089-0002)

station_name = <string: regular expression> (examples: MOUNT PEARL, New Y)

station_country = <string> (examples: Germany, United Kingdom, France)

parameter_name = <string> (examples: o3, no2, temp)

... and many other keywords as described in section 3.

columns = <string> (comma-separated list of database columns to be included in response; see sections 3.1 and 3.2)

format = <string> (json|html)

as_dict = <Boolean> (True|False; default is False). Only applicable if format is json.Returns results as json dictionaries instead of lists.

Aggregate = <Boolean>(True|False; default is False).If more than one data series is found for a station, combine results into one record per station.

Multiple argument values can be given by repeating the respective query option. Alternatively, you can provide multiple arguments as comma-separated list.

Response: Each query result consists of the fields that are specified in the columns argument. If columns are not specified, the output of each record will consist of the fieldsseries_id, network_name, station_id, parameter_label as the series query.

If no QUERY-OPTIONS are given, the complete set of stations will be returned.

Allowed column names are: numid (i.e. internal station number), network_name, station_id, station_type, station_type_of_area, station_category, station_name, station_country, station_state, station_lon, station_lat, station_alt, station_alt_flag, station_coordinate_status, station_reported_alt, station_google_alt, google_resolution, station_etopo_alt, station_etopo_min_alt_5km, station_etopo_relative_alt, station_timezone, station_population_density, station_max_population_density_5km, station_max_population_density_25km, station_nightlight_1km, station_nightlight_5km, station_max_nightlight_25km, station_nox_emissions, station_omi_no2_column, station_rice_production, station_wheat_production, station_climatic_zone, station_htap_region, station_dominant_landcover, station_landcover_description, station_toar_category, id (i.e. the data series internal number), parameter_name, parameter_label, parameter_attribute, parameter_sampling_type, parameter_measurement_method, parameter_original_units, parameter_calibration, parameter_contributor_shortname, parameter_contributor, parameter_contributor_country, parameter_dataset_type, parameter_status,

creation_date, modification_date, comments, data_start_date, data_end_date, parameter_pi, parameter_pi_email.

Note that in some cases the column name differs from the argument name of a search (or stations or series) query. For example, to search for stations in a given longitude range, you must write longitude=[4.2,5.7], whereas in order to retrieve the longitude values in the response, you must add station_lon to the columns list. See sections 3.1 and 3.2 for details on the database columns and query options.

Use of the columns argument:

Example: https://join.fz-juelich.de/services/rest/surfacedata/search/?station_name=Bay&parameter_name=o3,no2,temp&columns=id,station_id,station_name,station_country,station_state,parameter_pi

Result:

[[26355, "06-075-0006", "San Francisco - Bayview Hunters Point", "United States of America", "California", "unknown"], [28501, "60809", "421 JAMES STREET SOUTH _THunder Bay_", "Canada", "Ontario", "unknown"], [47859, "DECBAY01", "Deception Bay", "Australia", "Queensland", "David Wainwright"], [38271, "DEBY111", "Bayreuth/Hohenzollernring", "Germany", "Bayern", "unknown"], [48283, "HBA775S00", "Halley Bay", "United Kingdom", "unknown", "Neil Brough"], [28499, "60807", "615 JAMES STREET SOUTH _THunder Bay_", "Canada", "Ontario", "unknown"], [38272, "DEBY010", "Bayreuth/Rathaus", "Germany", "Bayern", "unknown"], [86943, "DEBY111", "Bayreuth/Hohenzollernring", "Germany", "Bayern", "Christian Ohlwein, Jan Keller"], [27989, "48-201-0055", "Houston Bayland Park", "United States of America", "Texas", "unknown"], [86848, "DEBY010", "Bayreuth/Rathaus", "Germany", "Bayern", "Christian Ohlwein, Jan Keller"], . . .]

Use of aggregate:

Example: https://join.fz-juelich.de/services/rest/surfacedata/search/?station_name=Bay&parameter_name=o3,no2,temp&columns=id,station_id,station_name,station_country,station_state&aggregate=True

Result:

[[[27989], "48-201-0055", "Houston Bayland Park", "United States of America", "Texas"], [[28500], "60808", "412 JAMES STREET SOUTH _THunder Bay_", "Canada", "Ontario"], [[19109, 86943], "DEBY111", "Bayreuth/Hohenzollernring", "Germany", "Bayern"], [[47859], "DECBAY01", "Deception Bay", "Australia", "Queensland"], [[28663], "64701", "1385 RIVER ROAD _Georgian Bay SouTH_", "Canada", "Ontario"], [[28539], "62001", "CHIPPEWA ST. - DND _NorTH Bay_", "Canada", "Ontario"], [[26983], "22-047-0009", "Bayou Plaquemine", "United States of America", "Louisiana"], [[27998], "48-201-1017", "Baytown Eastpoint", "United States of America", "Texas"], [[22119], "CAPMCANL1GOS", "Goose Bay", "Canada", "Newfoundland and Labrador"], [[48283], "HBA775S00", "Halley Bay", "United Kingdom", "unknown"], [[38271, 86942], "DEBY111", "Bayreuth/Hohenzollernring", "Germany", "Bayern"], [[26355], "06-075-0006", "San Francisco - Bayview Hunters Point", "United States

of America", "California"], [[27388], "34-017-0006", "Bayonne", "United States of America", "New Jersey"], [[28499], "60807", "615 JAMES STREET SOUTH _THunder Bay_", "Canada", "Ontario"], [[38272, 86847], "DEBY010", "Bayreuth/Rathaus", "Germany", "Bayern"], [[28501], "60809", "421 JAMES STREET SOUTH _THunder Bay_", "Canada", "Ontario"], [[18924, 18927, 86848], "DEBY010", "Bayreuth/Rathaus", "Germany", "Bayern"]]

Use of longitude and latitude bounds:

Example: https://join.fz-juelich.de/services/rest/surfacedata/search/?parameter_name=o3&longitude=[1.2,1.7]&latitude=[43.2,43.9]&columns=network_name,station_id,station_name,station_country,station_state,id,parameter_label

or: https://join.fz-juelich.de/services/rest/surfacedata/search/?parameter_name=o3&longitude=1.45&latitude=43.55&htol=0.35&columns=network_name,station_id,station_name,station_country,station_state,id,parameter_label

or: https://join.fz-juelich.de/services/rest/surfacedata/search/?parameter_name=o3&boundingbox=[1.2,43.3,1.7,43.9]&columns=network_name,station_id,station_name,station_country,station_state,id,parameter_label

Result:

[["AIRBASE", "FR12024", "BALMA", "France", "Midi-Pyr\u00e9n\u00e9es", 23785, "O3"], ["AIRBASE", "FR12030", "BERTHELOT", "France", "Midi-Pyr\u00e9n\u00e9es", 23796, "O3"], ["AIRBASE", "FR1054A", "BERTHELOT12", "France", "Midi-Pyr\u00e9n\u00e9es", 23797, "O3"], ["AIRBASE", "FR12001", "COLOMIERS", "France", "Midi-Pyr\u00e9n\u00e9es", 23863, "O3"], ["AIRBASE", "FR12037", "DOAS TOULOUSE", "France", "Midi-Pyr\u00e9n\u00e9es", 23912, "O3"], ["AIRBASE", "FR12004", "ECOLE M.JACQUIER", "France", "Midi-Pyr\u00e9n\u00e9es", 23930, "O3"], ["AIRBASE", "FR12021", "MAZADES", "France", "Midi-Pyr\u00e9n\u00e9es", 24091, "O3"], ["AIRBASE", "FR12041", "SICOVAL", "France", "Midi-Pyr\u00e9n\u00e9es", 24236, "O3"], ["AIRBASE", "FR12023", "calas", "France", "Midi-Pyr\u00e9n\u00e9es", 24393, "O3"]]

Use of altitude range:

Example: https://join.fz-juelich.de/services/rest/surfacedata/search/?parameter_name=o3&altitude=[-190,-10]&columns=network_name,station_id,station_name,station_country,station_alt

or: https://join.fz-juelich.de/services/rest/surfacedata/search/?parameter_name=o3&altitude=-100&vtol=90&columns=network_name,station_id,station_name,station_country,station_alt

Result:

[["AQS", "06-025-0002", "HOVLEY, BRAWLEY", "United States of America", -42.3], ["AQS", "06-025-1002", "1414 STATE ST., EL CENTRO", "United States of America", -12.7], ["AQS", "06-025-2001", "GENTRY & SINCLAIR, 6 MI NW OF CALIPATRIA", "United States of America", -69.8], ["AQS", "06-025-4001", "STATE ROUTE 86, WESTMORELAND", "United States of America", -61.6]]

Limit query to data series within a certain date range and with minimum length:

Example: https://join.fz-juelich.de/services/rest/surfacedata/search/?parameter_name=o3&data_before=1990-01-01%2000:00&data_after=1988-12-31%2023:00&min_data_length=10&columns=network_name,station_id,data_start_date,data_end_date,id

Result:

[["AQS", "37-051-1002", "1989-04-01T05:00:00", "1996-11-01T04:00:00", 27509], ["NAPS", "60607", "1988-01-01T05:00:00", "2004-07-09T15:00:00", 28438], ["UBA", "DEBW012", "1990-01-01T00:00:00", "2000-10-01T23:00:00", 20019], ["AQS", "45-031-0002", "1980-04-23T22:00:00", "1990-11-01T15:00:00", 27838], ["AIRBASE", "AT31904", "1988-01-01T00:00:00", "2012-12-31T22:00:00", 22254], ["EMEP", "AT0034G", "1990-01-01T00:00:00", "2013-12-31T23:00:00", 25683], . . .]

2.6 Stats

Calculate and return a set of statistics/metrics for a given series_id. Please note that hourly values cannot be retrieved from the TOAR database.

https://join.fz-juelich.de/services/rest/surfacedata/stats/?id=SERIESID[&QUERY-OPTIONS]

where QUERY-OPTIONS are:

sampling= <string> (statistical sampling interval: "daily", "monthly", "seasonal", "vegseason", "summer", "annual"; default: "monthly")

statistics= <string> (list of strings with the statistics/metrics names to be evaluated; see section 3.3 for details. Default: ["average_values", "standard_deviation", "value_count"]).

daterange = <2-element list of date strings with format YYYY-MM-DD hh:mm> (restrict processing of data to given daterange)

data_capture = <float> (data_capture threshold; default: 0.75)

format = <string> (json|html)

The seriesid (id=) must be given. Only one series can be processed with one request. Data series which are embargoed will not be processed.

Response: dictionary structure with datetime as key for the datetime values, the variable names as keys for the variable values, and metadata for the complete metadata of the seriesid.

Example:https://join.fz-juelich.de/services/rest/surfacedata/stats/?id=26688

Result:

{"datetime":["1998-04-01T00:00:00", "1998-05-01T00:00:00", . . . ,"2008-11-01T00:00:00"], "mean:[31.916784203102964,39.48547717842324, . . . ,NaN], "stddev":[19.290160626305752,30.47583388269126, . . . ,NaN], "count":[709.0,723.0, . . . ,5.0], "metadata":{"numid":11573,"network_name":"AQS", "station_id":"13-113-0001", . . . }}

Explanation:

This query returns monthly values of mean, standard deviation, and value count of ozone measurements at the station *Dot storage facility* in Georgia, USA (use the **series** service to obtain the series_id of this query).

Example:

https://join.fz-juelich.de/services/rest/surfacedata/stats/?id=21919&sampling=seasonal&statistics=dma8epax,somo35&daterange=[2010-01-01%2000:00,2016-12-31%2023:00

Result:

{"datetime":["1975-01-01T00:00:00", "1976-01-01T00:00:00", "1977-01-01T00:00:00", "1978-01-01T00:00:00", "1979-01-01T00:00:00", "1980-01-01T00:00:00", "1981-01-01T00:00:00", "1982-01-01T00:00:00", "1983-01-01T00:00:00", "1984-01-01T00:00:00", . . . ,"2016-01-01T00:00:00"], "dma8epax-DJF":[NaN,38.12499999999992,34.637499999999875, . . .], "dma8epax-MAM":[38.52499999999999,31.449999999999992, . . .], "dma8epax-JJA":[37.89999999999999,38.937500000000334, . . .], "dma8epax-SON":[39.93749999999998,31.637500000000006, . . .], "somo35-DJF":[NaN,50.91249999999816,8.463662790699093, . . .], . . . "metadata":{"numid":4610,"network_name":"GAW", "station_id": "SPO789S00", . . . }}

Explanation:

This query requests seasonal statistics of the ozone metrics DMA8EPAX (4th highest daily maximum 8-hour average) and SOMO35 (a European health-related metric) at the GAW station South Pole. Note that seasonal statistics are reported as one variable per season (with suffixes -DJF, -MAM, -JJA, and -SON) and that the datetime values for seasonal statistics always point to the beginning of the year.

Example:https://join.fz-juelich.de/services/rest/surfacedata/stats/?id=21843&sampling=vegseason&statistics=aot40,w126,average_values&format=json

Result:

{"datetime":["2004-01-01T00:00:00", "2005-01-01T00:00:00", . . . , "2014-01-01T00:00:00"], "aot40-wheat-warm_temperate_moist-SH":[36.18,1.9600000000000009, . . .], "aot40-rice-warm_temperate_moist-SH":[NaN,5.960371057513915, . . .], "w126-wheat-warm_temperate_moist-SH":[383.4789063213843, . . .], "w126-rice-warm_temperate_moist-SH":[NaN,88.82615555280175, . . .], "mean-wheat-warm_temperate_moist-SH":[30.077629151291493, . . .], "mean-rice-warm_temperate_moist-SH":[16.90771708683472, . . .], "metadata":{"numid":4615,"network_name":"GAW", "station_id": "CGO540S00", . . . }}

Explanation:

In this query vegetation-related query, the metrics mean, AOT40, and W126 are requested as aggregates over the respective growing seasons of rice and wheat (for details, see Schultz et al. (2017) and in particular supplement 1). Note that the variable names contain a suffix that describes the climatic zone of the station, which is then used to determine the rice and wheat vegetation periods at that location.

Example: https://join.fz-juelich.de/services/rest/surfacedata/stats/?sampling=annual&statistics=drmdmax1h,dma8eu,nvgt070&data_capture=0.9&id=40555

Result:

{"datetime": ["2003-01-01 00:00", "2004-01-01 00:00", "2005-01-01 00:00", "2006-01-01 00:00", "2007-01-01 00:00", "2008-01-01 00:00", "2009-01-01 00:00", "2010-01-01 00:00", "2011-01-01 00:00", "2012-01-01 00:00"], "drmdmax1h": [-999.0, 61.76856777777774, 71.27497555555554, 78.03873358024691, 88.69831481481471, 71.34341111111111, 68.40208522727261, 60.49619999999985, 74.232043181818, 65.4746465116277], "day_of_drmdmax1h": [1, 357, 47, 12, 280, 101, 295, 1, 55, 54], "dma8eu": [NaN, 54.12737500000007, NaN, NaN, NaN, 65.24050000000003, 64.58678750000038, 57.46133750000065, 62.03732500000059, 56.611512500000636], "nvgt070": [NaN, 4.0, NaN, NaN, NaN, 12.0, 9.0, 1.0, 4.0, 2.0], "metadata": {"numid": 8423, "network_name": "AIRBASE", "station_id": "FR03068", "station_local_id": "FR03068", "station_type": "traffic", "station_type_of_area": "urban", "station_category": "unknown", "station_name": "TOULON FOCH", "station_country": "France", "station_state": . . . ,"parameter_name": "no2", . . . }}

Explanation:

This query generates annual statistics of the three-month-running-average daily maximum value, the 26-highest daily maximum 8-hour-average (according to the EU time window), and the number of days where the maximum 8-hour average exceeds 70 ppb. Note that in

addition to drmdmax1h also the day of this maximum value is reported (day_of_drmdmax1h). Furthermore, the data_capture threshold for this query was changed to 0.9 (implying that 90% of the data must be valid in order to return a valid result) from the default value of 0.75. Note also that the selected data series is actually an NO2 measurement series. This is to demonstrate that the same statistics can technically be applied to different variables.

3. TOAR database columns and query arguments

3.1 Database table stations

Database field name (to be used in columns argument of search service)	**Description**	**Use in query arguments**
numid	internal serial number of the station	station_numid=<value or list>
network_name	name of the measurement network	network_name = <string or list>
station_id	station code within the network	station_id = <string or list>
station_type	characterisation of site. Normally one of "background", "industrial", "traffic"	station_type = <string or list>
station_type_of_area	characterisation of station environment. Normally one of "urban", "suburban", "rural", "remote"	station_type_of_area = <string or list> Note that this classification depends on the provider of the dataset and is not uniform. Use station_toar_category for a universal characterization.
station_category	other classification of stations (e.g. GAW category (global, regional, contributing))	station_category = <string or list>
station_name	full name of the station. Unicode characters are allowed.	station_name = <string> This argument is interpreted as regular expression.
station_country	country which operates the station	station_country = <string or list>

station_state	province/state/territory to which station belongs (may be blank)	station_state = <string or list>
station_lon	longitude coordinate of station (decimal degrees_east). This is our best estimate of the station location which is not always identical to the reported station coordinates.	longitude = <float or 2-element list of floats> Longitude ranges can be specified either by using a 2-element float list, or by using one center value and the htol = <float> argument. Htol applies to longitude and latitude. Another option is to use the boundingbox = [lon0,lat0,lon1,lat1] argument.
station_lat	latitude coordinate of station (decimal degrees_north). This is our best estimate of the station location which is not always identical to the reported station coordinates.	latitude = <float or 2-element list of floats> Latitude ranges can be specified either by using a 2-element float list, or by using one center value and the htol = <float> argument. Htol applies to longitude and latitude. Another option is to use the boundingbox = [lon0,lat0,lon1,lat1] argument.
station_alt	altitude of station (in m above sea level). This is our best estimate of the station altitude, which is not always identical to the reported station altitude, but frequently uses the elevation from google earth instead (see station_alt_flag).	altitude = <float or 2-element list of floats> Altitude ranges can be specified either by using a 2-element float list, or by using one center value and the vtol = <float> argument.
station_alt_flag	Flag value to document where station_alt was taken from. 0 = Reported station altitude 1 = Google maps elevation 2 = ETOPO1 elevation 3 = Station report or similar 4 = Personal communication 5 = Other source	station_alt_flag = <int or list of ints>

station_coordinate_status	an integer flag indicating our knowledge about the real station location. Note that this flag has been introduced rather late during the TOAR QA process and it may thus not always reflect the actual status of verification. Flag values are: -1: not checked (default value) 0: verified by google earth or other means. This means that a building or container which looks like a measurement site could be visibly identified or that a google earth feature is consistent with a detailed station description and is found at the location that is given in the station description. While in most cases the coordinates associated with a flag value of 0 will be exact within 10 metres or so, there are some stations where the accuracy is lower, for example if the air quality monitoring site is part of a larger campus and we could not exactly identify the building or container site of the air quality measurements. 1: verification not possible, but no reason to doubt that the measurement location should be accurate to within 100 metres or so. This means that no obvious station feature could be seen on google earth, but the area corresponds to the station description and could be a place where measurements are made. 2: unspecified potential issue with the station coordinates. This means that after checking the station location on google earth, comparing the reported station altitude to the google elevation, and looking at the station_type, station_type_of_area, and station_category information, something appears wrong, but for lack of better knowledge we retain station coordinates as given. This flag value is used particularly in cases when the coordinates of the same station are reported differently in various archives and if we could not locate the exact station location on google earth. 3: obvious error in station coordinate information. For example, a continental site is located in an ocean or lake, the measurement site is in the middle of a dense forest, etc. The station coordinates could not be corrected for lack of better information. 4: severe mismatch between reported station altitude and google elevation at station location (> 100 m) indicating wrong station coordinates. This flag value is only set after a potential correction of the station_alt value (see station_alt_flag), i.e. if we could not resolve a gross altitude difference.	station_coordinate_status = <int or list of ints>

	Note that for measurement sites on tall towers or in mountaineous terrain, altitude differences > 100 m may be correct and the coordinate status will not be flagged as 4 then. 5: no coordinates available -- given coordinates are completely invented! 6: no station metadata available -- given metadata is completely invented!	
station_reported_alt	This is the station altitude as reported by the data provider. Note: due to edits of obvious station coordinate errores before introducing the coordinate flagging scheme, there may be cases where the reported altitude in our database differs from the reported altitude in the original data sets.	station_reported_alt = <float or 2-element list of floats>
station_google_alt	Terrain elevation derived from the google maps API (see https://maps.googleapis.com/maps/api/elevation/json?locations=47.05444,12.958342; example coordinates of Sonnblick, Austria).	station_google_alt = <float or 2-element list of floats>
google_resolution	The horizontal resolution of google maps at the station location. This provides some indication of the accuracy of the station_google_alt information.	N.A.
station_etopo_alt	Terrain elevation at the station location from the ~1 km resolution ETOPO1 dataset.	station_etopo_alt = <float or 2-element list of floats>
station_etopo_min_alt_5km	Minimum elevation from the ETOPO1 dataset in an area of 5 km radius around the station location. This can be used to find out if a high altitude station is located in mountaineous terrain or on a plateau (see station_etopo_relative_alt).	N.A.
station_etopo_relative_alt	Station elevation above the surrounding area. Derived by subtracting the minimum altitude within a 5 km radius around the station location from the actual station altitude. The area altitude is obtained from the etopo1 map.	station_etopo_relative_alt = <float or 2-element list of floats>
station_timezone	Time zone of station; Note that all data will be stored as UTC, but the timezone information is needed to convert data back to local time for display.	station_timezone = <string or list of strings> Example string: Europe/Madrid
station_population_density	Year 2010 human population per square km from CIESIN GPW v3 (original horizontal resolution: 2.5 arc minutes)	station_population_density = <float or 2-element list of floats>
station_max_population_density_5km	Maximum population density in a radius of 5 km around the station location.	
station_max_population_density_25km	Maximum population density in a radius of 25 km around the station location.	
station_nightlight_1km	Year 2013 Nighttime lights brightness values from	

	NOAA DMSP (original horizontal resolution: 0.925 km)	
station_nightlight_5km	Year 2013 Nighttime lights brightness values (original horizontal resolution: 5 km)	
station_max_nightlight_25km	Maximum nighttime light intensity in a radius of 25 km around the station location.	
station_nox_emissions	Year 2010 NOx emissions from EDGAR HTAP inventory V2 in units of g m-2 yr-1 (original resolution: 0.1 degrees)	
station_omi_no2_column	Average 2011-2015 tropospheric NO2 columns from OMI at 0.1 degree resolution (Env. Canada) in units of 10^15 molecules cm-2.	
station_rice_production	Year 2000 rice production amount from FAO GAEZ at station location (units: thousand tons; original resolution: 5 arc minutes)	
station_wheat_production	Y2000 wheat production amount from FAO GAEZ at station location (units: thousand tons; original resolution: 5 arc minutes)	
station_climatic_zone	Climatic zone according to IPCC, 2006: 0: unclassified 1: Warm Temperate Moist 2: Warm Temperate Dry 3: Cool Temperate Moist 4: Cool Temperate Dry 5: Polar Moist 6: Polar Dry 7: Boreal Moist 8: Boreal Dry 9: Tropical Montane 10: Tropical Wet 11: Tropical Moist 12: Tropical Dry (original resolution: 5 arc minutes)	station_climatic_zone = <string or list of strings>
station_htap_region	An integer denoting the "tier1" region defined in the task force on hemispheric transport of air pollution (TFHTAP) coordinated model studies (see http://www.htap.org). Region codes are: 02: OCN Non-arctic/Antarctic Ocean 03: NAM US+Canada (upto 66 N; polar circle) 04: EUR Western + Eastern EU+Turkey (upto 66 N polar circle) 05: SAS South Asia: India, Nepal, Pakistan, Afghanistan, Bangadesh, Sri Lanka 06: EAS East Asia: China, Korea, Japan 07: SEA South East Asia 08: PAN Pacific, Australia+ New Zealand 09: NAF Northern Africa+Sahara+Sahel 10: SAF Sub Saharan/sub Sahel Africa 11: MDE Middle East: S. Arabia, Oman, etc, Iran, Iraq	station_htap_region =

	12: MCA Mexico, Central America, Caribbean, Guyanas, Venezuela, Columbia 13: SAM S. America 14: RBU Russia, Belarussia, Ukraine 15: CAS Central Asia 16: NPO Arctic Circle (North of 66 N)+Greenland 17: SPO Antarctic (original resolution: 0.1 degrees)	
station_dominant_landcover	The dominant IGBP landcover classification at the station location extracted from the MODIS MCD12C1 dataset (original resolution: 0.05 degrees). Landcover type values are: 0: Water 1: Evergreen Needleleaf forest 2: Evergreen Broadleaf forest 3: Deciduous Needleleaf forest 4: Deciduous Broadleaf forest 5: Mixed forest 6: Closed shrublands 7: Open shrublands 8: Woody savannas 9: Savannas 10: Grasslands 11: Permanent wetlands 12: Croplands 13: Urban and built-up 14: Cropland/Natural vegetation mosaic 15: Snow and ice 16: Barren or sparsely vegetated 255: Fill Value/Unclassified	station_dominant_landcover =
station_landcover_description	Text information about the landcover types and their area fractions in a radius of 25 km around the station location.	N.A.
station_toar_category	A station classification for the Tropsopheric Ozone Assessment Report based on the station proxy data that are stored in the database. 0: unclassified 1: rural, low elevation: derived as (station_omi_no2_column <= 8 and station_nightlight_5km <= 25 and station_population_density <= 3000 and station_max_population_density_5km <= 30000 and station_google_alt <= 1500 and station_etopo_relative_alt < 500). Note that this scheme may not catch all sites that are designated as rural. It will, however, provide a selection with reasonable certainty that no urban sites are included. 2: rural, high elevation: (station_omi_no2_column <= 8 and station_nightlight_5km <= 25 and station_population_density <= 3000 and	station_toar_category = <>

	(station_google_alt > 1500 or (station_google_alt > 800 and station_etopo_relative_alt < 500)). 3: urban; classified (station_population_density >= 15000 and station_nightlight_1km >= 60 and station_max_nightlight_25km == 63). Again, the intention here is to make reasonably sure that a site classified as urban really carries an urban signature.	

3.2 Database table parameter_series

Database field name (to be used in columns argument of search service)	**Description**	**Use in query arguments**
id	an internal serial number	series_id = <int or list of ints> Note that the stats service only accepts a single series_id, and the argument name for stats is id.
parameter_name	Name of the species or variable (all lower case)	parameter_name = <string or list of strings>
parameter_label	Automatically generated label of a parameter_series. The label consists of the capitalized parameter_name and additional information if this is needed to uniquely identify a data series by name. The additional information can consist of the parameter_contributor_shortname and/or of the parameter_attribute	N.A.
parameter_attribute	A series-specific attribute that may distinguish two series of the same parameter measured at the same station (e.g. all, filtered, etc.)	parameter_attribute = <string>
parameter_sampling_type	The method how observations were sampled. Standard values are: continuous, flask, filter	parameter_sampling_type = <string>
parameter_measurement_method	Instrument principle of measurement. Example (for ozone): UV absorption	parameter_measurement_method = <string>
parameter_original_units	Physical units in which parameter values were expressed in the original data	N.A.

Database field name (to be used in columns argument of search service)	Description	Use in query arguments
	files	
parameter_calibration	Information on the calibration of the parameter, such as calibration procedure and/or calibration scale	N.A.
parameter_contributor_shortname	Abbreviated string of parameter_contributor	parameter_contributor_shortname = <string or list of strings>
parameter_contributor	Institute or person name who provided data to network datacenter. If more than one contributor exists, the names will be separated by ;	N.A.
parameter_contributor_country	Country of contributor	parameter_contributor_country = <string or list of strings>
parameter_pi	Name of principal investigator of dataseries	N.A.
parameter_pi_email	Email of parameter_pi	N.A.
parameter_dataset_type	The type of the high-frequency data ("hourly", "event", etc.); this determines the data table name (e.g. "o3_hourly", "co_event").	parameter_dataset_type = <string>
parameter_status	an internal status flag which may be used to suppress display or analysis of an individual timeseries. Flag values are: 0: everything OK - use this dataset in any analyses 1: data was embargoed by originator; do not display publicly 2: NRT data ingestion; no metadata available, metadata was invented	N.A.
comments	Any comments on a data series, for example from the data QA in TOAR	N.A.
creation_date	Creation date when this entry was added in parameter_series table	creation_date =
modification_date	Date when this entry is last modified	modification_date
data_start_date	Start date of the data series	data_start_date You can also search for series

Database field name (to be used in columns argument of search service)	Description	Use in query arguments
		with a minimum length via the min_data_length argument; it accepts a float value in years.
data_end_date	End date of the data series. Note: data start and end date are not considering gaps (missing data)	data_end_date

3.3 List of statistics/metrics for stats service

More information on these statistics and metrics can be found in supplement 1 of Schultz et al. (2017).

Name	Description
data_capture	Fraction of valid (hourly) values available in the aggregation period.
average_values	Daily, monthly, … average value. No data capture criterion is applied, i.e. a daily average is valid if at least one hourly value of the day is present.
daytime_avg	Daytime average is defined as average of hourly values for the 12-h period from 08:00h to 19:59h solar time. All hourly values in the aggregation period are averaged, and the resulting value is valid if at least 75% of hourly values are present.
nighttime_avg	Same as daytime_average but accumulated over the daily interval from 20:00 h to 07:59 h solar time.
median	Median mixing ratio over the aggregation period. At least 10 valid values must be present to accept a median value as valid.
perc05	Fifth-percentile of hourly values in the aggregation period. At least 10 valid values must be present to accept a percentile value as valid.
perc10	As perc05, but for the 10^{th}-percentile.
perc25	As perc05, but for the 25^{th}-percentile.
perc75	As perc05, but for the 75^{th}-percentile.
perc90	As perc05, but for the 90^{th}-percentile.
perc95	As perc05, but for the 95^{th}-percentile.
perc98	As perc05, but for the 98^{th}-percentile. This percentile is only calculated for “summer” or “annual” aggregation periods.
dma8epa	Daily maximum 8-hour average statistics according to the US EPA definition. 8-hour averages are calculated for 24 bins starting at 0 h local time. The 8-h running mean for a particular hour is calculated on the concentration for that hour plus the following 7 hours. If less than 75% of data are present (i.e. less than 6 hours), the average is considered missing.

Name	Description
	When the aggregation period is “seasonal”, “summer”, or “annual”, the 4th highest daily 8-hour maximum of the aggregation period will be computed. Note that in contrast to the official EPA definition, a daily value is considered valid if at least 1 8-hour average is valid.
dma8epa_strict	As dma8epa, but additionally, a diurnal 8-hour maximum value is only saved if at least 18 out of the 24 8-hour averages are valid. This is the official dma8epa definition.
dma8epax	As dma8epa, but using the new US EPA definition of the daily 8-hour window from 7 h local time to 23 h local time.
dma8epax_strict	As dma8epax, but additionally, a diurnal 8-hour maximum value is only saved if at least 13 out of the 17 8-hour averages are valid. This is the official dma8epax definition.
dma8eu	As dma8epa, but using the EU definition of the daily 8-hour window starting from 17 h of the previous day. When the aggregation period is “seasonal”, “summer”, or “annual”, the 26th highest daily 8-hour maximum of the aggregation period will be computed.
dma8eu_strict	As dma8eu, but additionally, a diurnal 8-hour maximum value is only saved if at least 18 out of the 24 8-hour averages are valid. This is the official dma8eu definition.
avgdma8epax	Average value of the daily dma8epax statistics during the aggregation period.
drmdmax1h	Maximum of the 3-months running mean of daily maximum 1-hour mixing ratios during the aggregation period. This statistics also produces day_of_max_drmdmax1h, which is the Julian day in the year when the maximum value of the 3-months running mean of daily maximum 1-hour concentrations occurred.
somo10	Sum of excess of daily maximum 8-h means (EU Airbase standard with relaxed criterion: dma8eu) over the cut-off of 10 ppb, i.e. 20 µg/m3 calculated for all days in the aggregation period. SOMO10 will be set to missing if less than 75% of days are available. The quantity will be weighted by the number of theoretical days over the number of available days.
somo10_strict	As somo10, but using dma8eu_strict for data capture.
somo35	As somo10, but accumulating ozone values above 35 ppb.
somo35_strict	As somo10_strict, but accumulating ozone values above 35 ppb.
w90	Daily maximum W90 5-h Experimental Exposure Index: EI = SUM(w_iC_i) with weight $w_i = 1/[1 + M \exp(-AC_i/1000)]$, where M = 1400, A = 90, and where C_i is the hourly average O3 mixing ratio in units of ppb (Lefohn et al., 2010). For each day, 24 W90 indices are computed as 5-hour sums, requiring that at least 4 of the 5 hours are valid data (75%). If a sample consists of only 4 data points, a fifth value shall be constructed from averaging the 4 valid mixing ratios. For aggregation periods “month”, “season”, “summer”, or “annual”, the 4th highest W90 value is computed, but only if at least 75% of days in this period have valid W90 values.

Name	Description
aot40	Daily 12-h AOT40 values are accumulated using hourly values for the 12-h period from 08:00h until 19:59h solar time interval. AOT40 is defined as cumulative ozone above 40 ppb. If less than 75% of hourly values (i.e. less than 9 out of 12 hours) are present, the cumulative AOT40 is considered missing. When there exist 75% or greater data capture in the daily 12-h window, the scaling by fractional data capture (ntotal/nvalid) is utilized. For monthly, seasonal, summer, or annual statistics, the daily AOT40 values are accumulated over the aggregation period and scaled by (ntotal/nvalid) days. If less than 75% of days are valid, the value is considered missing.
daylight_aot40	As aot40, but using solar elevation > 5 degrees to identify “daytime” hours.
w126	Daily W126 index is accumulated using hourly values for the 12-h period from 08:00h until 19:59h solar time interval. W126 = SUM(w_iC_i) with weight $w_i = 1/[1 +M \exp(-AC_i/1000)]$, where M = 4403, A = 126, and where C_i is the hourly average O3 mixing ratio in units of ppb. If there are less than 9 valid hourly values in the 12 hour window, the daily value is considered missing. When there exist 75% or greater data capture in the daily 12-h window, the scaling by fractional data capture (ntotal/nvalid) is utilized. Seasonal, summer, or annual statistics are calculated as sum over the daily W126 values. Results are marked as missing if less than 75% of daily values are valid.
w126_24h	As w126, but using all 24 hours of a day.
nvgt050	Number of days with exceedance of the dma8epax value above 50 ppb. The value is marked as missing if less than 75% of days contain valid data.
nvgt060	Number of days with exceedance of the dma8epax value above 60 ppb. The value is marked as missing if less than 75% of days contain valid data.
nvgt070	Number of days with exceedance of the dma8epax value above 70 ppb. The value is marked as missing if less than 75% of days contain valid data.
nvgt080	Number of days with exceedance of the dma8epax value above 80 ppb. The value is marked as missing if less than 75% of days contain valid data.
nvgt090	Number of days with exceedance of the daily max1h_values above 90 ppb. The value is marked as missing if less than 75% of days contain valid data.
nvgt100	Number of days with exceedance of the daily max1h_values above 100 ppb. The value is marked as missing if less than 75% of days contain valid data.
nvgt120	Number of days with exceedance of the daily max1h_values above 120 ppb. The value is marked as missing if less than 75% of days contain valid data.

4. A Python example

The following python code provides a recipe how ozone metrics can be obtained from the TOAR database through the REST services described above. In this example, ozone statistics are obtained for all sites in California, US. The STATION_STATE variable can of course be replaced by other database parameters as described above. We recommend that changes to the request URLs are tested via a web browser. The user is assumed to be familiar with python, including the numpy and pandas libraries. The actual data processing is not included in this program and must be added by the user.

```
 # -*- coding: utf-8 -*-

"""
toar_rest_demo: demonstrate use of JOIN REST interface to TOAR
database from python

Created on Thu Jan 26 09:01:19 2017

@author: m.g.schultz, forschungszentrum juelich, germany
"""

from urllib.request import urlopen
import json

# user vars
STATION_STATE="California"


BASEURL = "https://join.fz-juelich.de/services/rest/surfacedata/"
# first URL to find all sites matching certain conditions (here
station_state)
# note that %s will be replaced by STATION_STATE later
URL1 =
"stations/?station_state=%s&parameter_name=o3&columns=id,network_nam
e,station_id&format=json"
# scond URL to return data for one data series at a time
# note that %i will be replaced by the respective dataset id
URL2 =
"stats/?id=%i&sampling=seasonal&statistics=dma8epax,daytime_avg,medi
an,perc98&format=json"


# first: find all sites
print("Opening URL1...")
response = urlopen(BASEURL + URL1 %
(STATION_STATE)).read().decode('utf-8')
print("response = ", response[:400], " ... ")
metadata = json.loads(response)

# now loop over data series
# note: we assume that there is only one data series per station
for s in metadata:
    all_dataseries = s[0]
    if len(all_dataseries) > 1:
```

```
        raise ValueError("More than one data series found at %s.
Modify your code" % json.dumps(s))
    print("Opening URL2...")
    dresponse = urlopen(BASEURL + URL2 %
(all_dataseries[0])).read().decode('utf-8')
    data = json.loads(dresponse)
    # do something with the data
    print("Data columns: ", data.keys())
    print("metadata: ", data['metadata'])
    print("datetime: ", data['datetime'])
    print("springtime median: ", data['median-MAM'])
    # for demo purposes we break the loop here
    break
```