

TOAR Data User Guide #6

Quick Start for TOAR Analysis Service

toar-data.fz-juelich.de

Version 2.0.0 | September 11, 2025

The TOAR Data Team



CONTENTS:

1	Introduction	1
2	Learning Objectives	3
3	Services basics	5
3.1	Query arguments	5
3.2	Response format	5
3.3	Description of services	5
4	Data	7
4.1	Time Series	7
4.2	Map	7
5	Statistics	9
5.1	Map	10
5.2	Available Statistics	10
6	Trends	13
7	Status and result	15
7.1	Status	15
7.2	Result	15
8	Aggregated output format	17
8.1	Raw	17
8.2	By_statistic	17
8.3	Json_simple	17
8.4	By_stat_quant	18
9	Indices and tables	19

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

This is a quick overview of the Representational State Transfer (REST) service that allows you to get the analysis of products from the Tropospheric Ozone Assessment Report (TOAR) database of surface ozone observations.

For further information on the URL architecture and query options of the TOAR analysis REST interface check the [documentation](#)¹.

and you can find the FAQ [here](#)².

¹ <https://toar-data.fz-juelich.de/api/v2/analysis/>

² https://toar-data.fz-juelich.de/sphinx/TOAR_UG_Vol04_FAQ/build/html/index.html

LEARNING OBJECTIVES

After reading the guide, you should be able to:

- data: get hourly data from the database
 - timeseries: get hourly time series data
 - map: get snapshot of one point in time of one variable
- statistics: get aggregated data from the database
 - map: get snapshot of aggregated values of one variable
- trends: get trends of aggregated data from the database
- status: check the current status of your query
- result: get the query result

SERVICES BASICS

3.1 Query arguments

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

3.2 Response format

The response can be either synchronous or asynchronous. If the response is synchronous you will receive the requested result directly. If the response is asynchronous you will not get your requested result but instead a unique task identifier for your request. This id can be used to check the status of your request. When your result is ready the id will redirect you to the requested result. This type of approach is chosen for queries that are expected to take more time to process.

3.3 Description of services

The following pages will guide you through some services of the FastAPI and grant you access to the hourly data of the TOAR database.

4.1 Time Series

The following [link](#)³ contains some data, which stores the date and time of the measurement, a representing value, an id and a version, which should be 1.0 for every measurement.

With a combination of query options from both [TOARDB REST interface - 2.4 Stationmeta](#)⁴ and [TOARDB REST interface - 2.5 Timeseries](#)⁵ you can filter the data you want to get. ([https://toar-data.fz-juelich.de/api/v2/data/timeseries/\[?QUERY-OPTIONS\]](https://toar-data.fz-juelich.de/api/v2/data/timeseries/[?QUERY-OPTIONS]))

Here are some examples to use as query options:

daterange = <list of two datetimes: date range for which to extract data> flags = <list of strings: only select data points with the specified quality flags> (for a description of flags and all available flag names see [User Guide - 5.2 Data Quality Flags](#)⁶) format = <string> (json|csv) (default: json)

Response: The query will return a unique task identifier and a link to check the status of your query. **Example:** https://toar-data.fz-juelich.de/api/v2/analysis/data/timeseries/?country=DE&variable_id=5&limit=3&daterange=2010-01-01T00:00:00,2020-12-31T23:59:59&flags=ALLOC&format=csv

Result: { "task_id": "94e3888a-33f8-4adf-a6d6-4d8627c9ecc0", "status": "https://toar-data.fz-juelich.de/api/v2/analysis/status/94e3888a-33f8-4adf-a6d6-4d8627c9ecc0" }

To retrieve the result send a request to the [status endpoint](#)⁷ with your task identifier. If the result is there you will be redirected. The result will be a zip archive containing one file per time series in the format you have chosen.

4.2 Map

The following [link](#)⁸ contains some data, which stores an id and a value for every measurement.

Here are some examples to use as query options:

datetime = <datetime: date and time for which to extract data> variable_id = <integer: variable to extract> bounding_box = <list of four numbers: bounding_box (min_lat,min_lon,max_lat,max_lon) in degrees_north/degrees_east to define a geographical rectangle (do not set anything for global extraction)> (default: None) format = <string> (json|csv) (default: json)

³ <https://toar-data.fz-juelich.de/api/v2/data/timeseries/>

⁴ <https://toar-data.fz-juelich.de/api/v2/#stationmeta>

⁵ <https://toar-data.fz-juelich.de/api/v2/#stationmeta>

⁶ https://toar-data.fz-juelich.de/sphinx/TOAR_UG_Vol03_Database/build/html/data-quality.html#data-quality-flags

⁷ <https://toar-data.fz-juelich.de/api/v2/analysis/#status>

⁸ <https://toar-data.fz-juelich.de/api/v2/data/map/>

Response: The query will return tuples of latitude, longitude and value at the location in the specified format. **Example:** https://toar-data.fz-juelich.de/api/v2/analysis/data/map/?datetime=2020-07-23T13:00:00&variable_id=5&bounding_box=47.5,6.5,54.5,14.5

Result: [{"lat":47.81564999946587,"lon":13.03488,"value":68.0628783549876}, {"lat":47.8055555994659,"lon":13.043333,"value":65.96268275498761}, ... {"lat":53.2465,"lon":6.60894,"value":43.9787975}, {"lat":52.0918,"lon":6.60537,"value":54.1289075949876}]

STATISTICS

All statistics are calculated and reported in the local time without shifts for daylight saving time of the station where the data originated from [https://toar-data.fz-juelich.de/api/v2/analysis/statistics/\[?QUERY-OPTIONS\]](https://toar-data.fz-juelich.de/api/v2/analysis/statistics/[?QUERY-OPTIONS])⁹ —> link does not work

where QUERY-OPTIONS are:

any combination of query options from both [TOARDB REST interface - 2.4 Stationmeta](#)¹⁰ and [TOARDB REST interface - 2.5 Timeseries](#)¹¹ daterange = <list of two datetimes: date range for which to extract data> flags = <list of strings: only select data points with the specified quality flags> (for a description of flags and all available flag names see [User Guide - 5.2 Data Quality Flags](#)¹²) sampling = <string: temporal aggregation to use> (for available values see [ALLOWED_SAMPLING_VALUES](#)¹³) statistics = <list of strings: statistics to calculate> (for available values and details see [5.2. Available Statistics](#)) seasons = <list of strings: seasons to use for seasonal aggregations> (for available values see [SEASON_DICT](#)¹⁴) (default: "DJF,MAM,JJA,SON") crops = <list of strings: crops to use for vegeseason aggregations> (for available values see [ALLOWED_CROPS_VALUES](#)¹⁵) (default: "wheat,rice") min_data_capture = <number: minimal fraction of available hourly values in the aggregation interval to report an aggregated value, must be between 0 and 1> (default: 0.75) metadata_scheme = <string: select how much metadata is returned> (basic|extended|full) (default: full) format = <string>(raw|by_statistic) (for details on the formats see [8. Aggregated Output Formats](#))

Response: The query will return a unique task identifier and a link to check the status of your query.

Example: https://toar-data.fz-juelich.de/api/v2/analysis/statistics/?country=DE&variable_id=5&limit=3&daterange=2010-01-01T00:00:00,2020-12-31T23:59:59&flags=AllOK&sampling=annual&statistics=mean,median,min,max

Result: {"task_id": "e2b17c39-6f80-4083-9bb8-f90cd72812b9", "status": "https://toar-data.fz-juelich.de/api/v2/analysis/status/e2b17c39-6f80-4083-9bb8-f90cd72812b9"} To retrieve the result send a request to the status endpoint with your task identifier. If the result is there you will be redirected. The result will be a zip archive containing files in the format you have chosen.

⁹ <https://toar-data.fz-juelich.de/api/v2/analysis/statistics/>

¹⁰ <https://toar-data.fz-juelich.de/api/v2/#stationmeta>

¹¹ <https://toar-data.fz-juelich.de/api/v2/#stationmeta>

¹² https://toar-data.fz-juelich.de/sphinx/TOAR_UG_Vol03_Database/build/html/data-quality.html#data-quality-flags

¹³ <https://gitlab.jsc.fz-juelich.de/esde/toar-public/toarstats/-/blob/master/toarstats/metrics/constants.py#L12-21>

¹⁴ <https://gitlab.jsc.fz-juelich.de/esde/toar-public/toarstats/-/blob/master/toarstats/metrics/constants.py#L93-147>

¹⁵ <https://gitlab.jsc.fz-juelich.de/esde/toar-public/toarstats/-/blob/master/toarstats/metrics/constants.py#L149-152>

5.1 Map

[https://toar-data.fz-juelich.de/api/v2/analysis/statistics/\[?QUERY-OPTIONS\]](https://toar-data.fz-juelich.de/api/v2/analysis/statistics/[?QUERY-OPTIONS])¹⁶ #link does not work

where QUERY-OPTIONS are:

daterange = <str: comma separated start and end date and time for which to extract data> variable_id = <integer: variable to extract> bounding_box = <list of four numbers: bounding_box (min_lat,min_lon,max_lat,max_lon) in degrees_north/degrees_east to define a geographical rectangle (do not set anything for global extraction)> (default: None) statistics = <list of strings: statistics to calculate> (for available values and details see 3. Available Statistics) format = <string> (json|csv) (default: json)

Response: The query will return a unique task identifier and a link to check the status of your query.

Example: https://toar-data.fz-juelich.de/api/v2/analysis/statistics/map/?daterange=2010-01-01T00:00:00,2020-12-31T23:59:59&variable_id=5&bounding_box=50,6,52,8&statistics=avgdma8epax&format=csv **Result:** {"task_id": "5a0beddf-a1c4-4584-9fb7-d5e98bafcd46", "status": "https://toar-data.fz-juelich.de/api/v2/analysis/status/5a0beddf-a1c4-4584-9fb7-d5e98bafcd46"} To retrieve the result send a request to the status endpoint with your task identifier. If the result is there you will be redirected. The result will be a zip archive containing files in the format you have chosen.

5.2 Available Statistics

Remarks about the minimal fraction of available hourly data use 75% (the default) in the descriptions below. When you define a different min_data_capture that value is used instead.

For more details see [supplement 1 of Schultz et al. \(2017\)](#)¹⁷

Name	Description
aot40	Daily 12-h AOT40 values are accumulated using hourly values for the 12-h period from 08:00h until 19:59h.
avgdma8epax	Average value of the daily dma8epax statistics during the aggregation period.
count	Number of available values in the aggregation period.
dark_aot40	As aot40, but using solar elevation <= 5 degrees to identify "dark" hours.
dark_avg	As mean, but using solar elevation <= 5 degrees to identify "dark" hours.
data_capture	Fraction of valid (hourly) values available in the aggregation period.
daylight_aot40	As aot40, but using solar elevation > 5 degrees to identify "daytime" hours.
daylight_avg	As mean, but using solar elevation > 5 degrees to identify "daytime" hours.
daytime_avg	Daytime average is defined as average of hourly values for the 12-h period from 08:00h to 19:59h. All values are included.
diurnal_cycle	Diurnal cycle (must be given without any other statistics).
dma8epa	Daily maximum 8-hour average statistics according to the US EPA definition. 8-hour averages are calculated from 7 h local time to 23 h.
dma8epa_strict	As dma8epa, but additionally, a diurnal 8-hour maximum value is only saved if at least 18 out of the 24 hours are available.
dma8epax	As dma8epa, but using the new US EPA definition of the daily 8-hour window from 7 h local time to 23 h.
dma8epax_strict	As dma8epax, but additionally, a diurnal 8-hour maximum value is only saved if at least 13 out of the 24 hours are available.
dma8eu	As dma8epa, but using the EU definition of the daily 8-hour window starting from 17 h of the previous day.
dma8eu_strict	As dma8eu, but additionally, a diurnal 8-hour maximum value is only saved if at least 18 out of the 24 hours are available.
drmdmax1h	Maximum of the 3-months running mean of daily maximum 1-hour mixing ratios during the aggregation period.
m7_avg	Daytime mean values (9-16h).
max	Maximum in the aggregation period.
max1h	Daily maximum hourly value.
mean	Average value in the aggregation period.

¹⁶ <https://toar-data.fz-juelich.de/api/v2/analysis/statistics/>

¹⁷ https://ucp.silverchair-cdn.com/ucp/content_public/journal/elementa/5/10.1525_elementa.244/3/elementa-5-244-s1.pdf

Name	Description
median	Median value in the aggregation period.
min	Minimum in the aggregation period.
nighttime_avg	Same as daytime_average but accumulated over the daily interval from 20:00 h to 07:59 h.
nvgt050	Number of days with exceedance of the dma8epax value above 50 ppb. The value is marked as missing if no data is available.
nvgt060	Number of days with exceedance of the dma8epax value above 60 ppb. The value is marked as missing if no data is available.
nvgt070	Number of days with exceedance of the dma8epax value above 70 ppb. The value is marked as missing if no data is available.
nvgt080	Number of days with exceedance of the dma8epax value above 80 ppb. The value is marked as missing if no data is available.
nvgt090	Number of days with exceedance of the daily max1h_values above 90 ppb. The value is marked as missing if no data is available.
nvgt100	Number of days with exceedance of the daily max1h_values above 100 ppb. The value is marked as missing if no data is available.
nvgt120	Number of days with exceedance of the daily max1h_values above 120 ppb. The value is marked as missing if no data is available.
nvgtall	nvgt050+nvgt060+nvgt080+nvgt090+nvgt100+nvgt120.
p05	Fifth-percentile of hourly values in the aggregation period.
p10	As p05, but for the 10th-percentile.
p25	As p05, but for the 25th-percentile.
p75	As p05, but for the 75th-percentile.
p90	As p05, but for the 90th-percentile.
p95	As p05, but for the 95th-percentile.
p98	As p05, but for the 98th-percentile.
p99	As p05, but for the 99th-percentile.
percentiles1	p25+p50+p75.
percentiles2	p5+p10+p25+p50+p75+p90+p95(+p98+p99 if aggregation period is "summer" or "annual").
somo10	Sum of excess of daily maximum 8-h means (EU Airbase standard with relaxed criterion: dma8eu) over 10 ppb.
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.
stddev	Standard deviation in the aggregation period.
w126	Daily W126 index is accumulated using hourly values for the 12-h period from 08:00h until 19:59h. W126 is the maximum of the 12-h period.
w126_24h	As w126, but using all 24 hours of a day.
w90	Daily maximum W90 5-h Experimental Exposure Index: $EI = \text{SUM}(wiCi)$ with weight $wi = 1/[1 + \text{Mexp}(-Ci/90)]$

TRENDS

All statistics are calculated in the local time without shifts for daylight saving time of the station where the data originated from. Daily aggregates will report the trend in ppbv/day and monthly aggregates will report the trend in ppbv/month.

[https://toar-data.fz-juelich.de/api/v2/analysis/trends/\[?QUERY-OPTIONS\]](https://toar-data.fz-juelich.de/api/v2/analysis/trends/[?QUERY-OPTIONS])¹⁸

where QUERY-OPTIONS are:

any combination of query options from both [TOARDB REST interface - 2.4 Stationmeta](#)¹⁹ and [TOARDB REST interface - 2.5 Timeseries](#)²⁰ daterange = <list of two datetimes: date range for which to extract data> flags = <list of strings: only select data points with the specified quality flags> (for a description of flags and all available flag names see User Guide - [5.2 Data Quality Flags](#)²¹) sampling = <string: temporal aggregation to use> (daily|monthly) statistics = <list of strings: statistics to calculate> (for available values and details see [5.2. Available Statistics](#)) seasons = <list of strings: seasons to use for seasonal aggregations> (for available values see [SEASON_DICT](#)²²) (default: "DJF,MAM,JJA,SON") crops = <list of strings: crops to use for vegseason aggregations> (for available values see [ALLOWED_CROPS_VALUES](#)²³) (default: "wheat,rice") min_data_capture = <number: minimal fraction of available hourly values in the aggregation interval to report an aggregated value, must be between 0 and 1> (default: 0.75) method = <string: regression analysis method to use> (OLS|quant) (default: quant) quantiles = <list of numbers: quantiles to use when using quantile regression, must be between 0 and 1> num_samples = <number: number of sampled trends in moving block bootstrap> (default: 50) metadata_scheme = <string: select how much metadata is returned> (basic|extended|full) (default: full) format = <string> (json_simple|by_stat_quant) (for details on the formats see Reference [8. Aggregated Output Formats](#)) (default: json_simple)

Response: The query will return a unique task identifier and a link to check the status of your query. **Example:** https://toar-data.fz-juelich.de/api/v2/analysis/trends/?country=DE&variable_id=5&limit=3&daterange=2010-01-01T00:00:00,2020-12-31T23:59:59&flags=ALL&sampling=monthly&statistics=avgdma8epax,dma8eu&quantiles=0.25,0.5,0.75 **Result:** {"task_id": "24666af1-5a51-4223-b8fc-c7d2d5f0070e", "status": "https://toar-data.fz-juelich.de/api/v2/analysis/status/24666af1-5a51-4223-b8fc-c7d2d5f0070e"} To retrieve the result send a request to the status endpoint with your task identifier. If the result is there you will be redirected. The result will be a zip archive containing files in the format you have chosen.

¹⁸ <https://toar-data.fz-juelich.de/api/v2/analysis/trends/>

¹⁹ <https://toar-data.fz-juelich.de/api/v2/#stationmeta>

²⁰ <https://toar-data.fz-juelich.de/api/v2/#stationmeta>

²¹ https://toar-data.fz-juelich.de/sphinx/TOAR_UG_Vol03_Database/build/html/data-quality.html#data-quality-flags

²² <https://gitlab.jsc.fz-juelich.de/esde/toar-public/toarstats/-/blob/master/toarstats/metrics/constants.py#L93-147>

²³ <https://gitlab.jsc.fz-juelich.de/esde/toar-public/toarstats/-/blob/master/toarstats/metrics/constants.py#L149-152>

STATUS AND RESULT

7.1 Status

[https://toar-data.fz-juelich.de/api/v2/analysis/status/\[task_id\]](https://toar-data.fz-juelich.de/api/v2/analysis/status/[task_id])²⁴ Response: If the result is not ready yet the response will return the task id and the URL itself again. If the result is ready you will be redirected to the result endpoint. Example: <https://toar-data.fz-juelich.de/api/v2/analysis/status/e2b17c39-6f80-4083-9bb8-f90cd72812b9> Result: {"task_id": "e2b17c39-6f80-4083-9bb8-f90cd72812b9", "status": "https://toar-data.fz-juelich.de/api/v2/analysis/status/e2b17c39-6f80-4083-9bb8-f90cd72812b9"} or redirect to <https://toar-data.fz-juelich.de/api/v2/analysis/result/e2b17c39-6f80-4083-9bb8-f90cd72812b9>

7.2 Result

[https://toar-data.fz-juelich.de/api/v2/analysis/result/\[task_id\]](https://toar-data.fz-juelich.de/api/v2/analysis/result/[task_id])²⁵ Response: A zip archive containing the query result in the format you requested. Example: <https://toar-data.fz-juelich.de/api/v2/analysis/result/e2b17c39-6f80-4083-9bb8-f90cd72812b9> Result: zip archive

²⁴ <https://toar-data.fz-juelich.de/api/v2/analysis/status/>

²⁵ <https://toar-data.fz-juelich.de/api/v2/analysis/result/>

AGGREGATED OUTPUT FORMAT

Following are the descriptions of available output formats for aggregated time series data. If you want to try different output formats to find the best one for your needs you can run your query with a low limit (e.g. limit=3) to check out the different outputs.

8.1 Raw

This output format will create csv files in the same way as [TOARDB REST interface - 2.7 Data](#).²⁶ The zip archive will contain one csv file per time series. The name of each individual csv file will be "<time_series_id>.csv"

8.2 By_statistic

This output format will create one csv file per requested statistic and one additional csv file with all the metadata. Each row in all the files will contain the information (either metadata or aggregated values) for one time series. All files have the same number and order of rows so that you can match the metadata and different aggregates for each time series via the row position. The metadata file is called "metadata.csv" and the files for the aggregated values are called "<statistic>.csv".

8.3 Json_simple

This output format will create one JSON file per time series, statistic and quantile (if using quantile regression). Each JSON file will contain a dictionary with one key which holds all the metadata and a second key holding the calculated trend, uncertainty and p-value. The files are called "<time_series_id><statistic><quantile>.json".

²⁶ <https://toar-data.fz-juelich.de/api/v2/#data>

8.4 By_stat_quant

This output format will create one csv file per requested statistic and quantile and one additional csv file with all the metadata. Each row in all the files will contain the information (either metadata or trend values) for one time series. All files have the same number and order of rows so that you can match the metadata and different trends for each time series via the row position. The metadata file is called "metadata.csv" and the files for the trend values are called "<statistic>_<quantile>.csv".

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`