

TOAR Data User Guide #2

TOAR Database Quick Start

toar-data.fz-juelich.de

Version 2.0.0 | December 6, 2025

The TOAR Data Team



CONTENTS:

1	Introduction	1
2	Learning Objectives	3
3	The basics	5
4	Create and use an own Mytoken	7
4.1	Creation	7
4.2	Usage	8
4.3	Token Information	8
5	Examples	9
6	Usage scenarios	11
7	Indices and tables	13

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

This is a quick overview of querying the TOAR database. It demonstrates how you can retrieve specific data from the TOAR database either by your web browser of choice or with a script. For further information on the available services and data extractions check [User Guide¹](https://toar-data.fz-juelich.de/documentation/TOAR_UG_Vol03_Database.pdf) and you can find the FAQ [here²](https://toar-data.fz-juelich.de/sphinx/TOAR_UG_Vol04_FAQ/build/html/index.html).

¹ https://toar-data.fz-juelich.de/documentation/TOAR_UG_Vol03_Database.pdf

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

LEARNING OBJECTIVES

After reading, you should be able to:

- find stations, variables, time series metadata
- filter stations by region or other characteristics
- filter time series by stations or variables
- use either a web browser or script to do so

THE BASICS

The TOAR database offers a REST service to retrieve its data and metadata.

A REST API is a web service, where you can format the URL with question marks and ampersands and issue commands to a web page. These will then be processed and the result returned to you.

<https://toar-data.fz-juelich.de/api/v2/> provides a description of the available REST services.

There are four core endpoints:

- stationmeta (describing the station),
- timeseries (describing the time series)
- variables (describing the variables), and
- data (describing the data).

CREATE AND USE AN OWN MYTOKEN

Since accessing TOAR services can consume considerable resources, we are forced to introduce user registration and login to the TOAR data infrastructure.

[Here³](#), you can read more about our concept for “Authentication And Authorization”.

To make (larger) requests to the TOAR Data Infrastructure, you need an Access Token. You can obtain this token manually through your profile in the TOAR Dashboard.

However, this manually created Access Token is only valid for 4000 seconds (about one hour), which makes it unsuitable for long-running or automated queries.

To enable automated workflows, you can create your own mytoken, which allows you to request new Access Tokens programmatically and use them when sending requests to the TOAR Data Infrastructure.

4.1 Creation

A mytoken can be created using the web application provided by KIT: <https://mytoken.data.kit.edu/#mt>. When creating your mytoken, make sure to select the following settings to avoid issues:

4.1.1 necessary settings

OpenID Provider: **Helmholtz-AAI** Capabilities: *at least AT* (Allows obtaining OpenID Connect Access Tokens) Restrictions:

- Scopes (necessary):
 - **eduperson_unique_id**
 - **email**
 - **profile**

³ https://toar-data-dev.fz-juelich.de/sphinx/TOAR_UG_Vol03_Database/build/html/data-access.html#authentication-and-authorization

4.1.2 other settings

Other settings that might be of interest:

Rotation

- Lifetime: *default 0* (= infinite lifetime)

Restrictions:

- Expires At: (If set, the mytoken cannot be used after this time)
- Usages AT: *default 15* (If set, the mytoken can only be used this often to request access tokens)

Save your token! (at creation time, you will not be able to access the credentials at a later point in time)

4.2 Usage

With the following code snippet you can create an AccessToken for the Helmholtz AAI using your mytoken

```
import os
import requests

MYTOKEN = "TheTokenThatYouSavedFromTheAboveStep"
MYTOKEN_ENDPOINT = "https://mytoken.data.kit.edu/api/v0/token/access"

def get_access_token():
    payload = {
        "grant_type": "mytoken",
        "mytoken": MYTOKEN
    }

    resp = requests.post(MYTOKEN_ENDPOINT, json=payload, timeout=10)
    resp.raise_for_status()

    data = resp.json()
    at = data.get("access_token")

    return at

if __name__ == "__main__":
    access_token = get_access_token()
    print("Access Token:", access_token)
```

4.3 Token Information

You can view your active mytokens by logging in to <https://mytoken.data.kit.edu/> via Helmholtz-AAI and selecting List and revoke mytokens. There, you can see all your active mytokens, revoke them if needed, and view the event history of each token. You can also visit <https://mytoken.data.kit.edu/#token-info> and enter a mytoken to receive information about it, such as whether it is valid or when it expires.

EXAMPLES**1. find a station**

<https://toar-data.fz-juelich.de/api/v2/stationmeta/?country=DE,NL&limit=None>
provides all stations from Germany and The Netherlands.

2. find all variables

From the first command, we found the station Niederzier (next to Jülich Research Centre). Its station code is DENW074. We now require all time series for this specific station that are stored in the TOAR database: <https://toar-data.fz-juelich.de/api/v2/search/?codes=DENW074> provides all timeseries of station DENW074.

3. download data

From the previous results, we choose the time series of ozone from the resource_provider(roles) German Environment Agency (UBA).
Its id is 18763.

We want to download all available data in csv format:

<https://toar-data.fz-juelich.de/api/v2/data/timeseries/18763?format=csv>
provides all data from time series with id 18763.

4. Python example

<https://toar-data.fz-juelich.de/api/v2/#python-example>
gives an example of how to use the REST service with python.
Prerequisites are the packages io, json, pandas, and requests.

USAGE SCENARIOS

1. You are interested in all stations of one country

<https://toar-data.fz-juelich.de/api/v2/stationmeta/?country=DE> provides all stations from Germany. The country must be specified with its ISO-3166 ALPHA-2 code (DE -> Germany).

2. You are interested in time series data for variables ozone and pm1 of a specific region

<https://toar-data.fz-juelich.de/api/v2/variables/o3> and <https://toar-data.fz-juelich.de/api/v2/variables/pm1> will get you the variable id for ozone (5) and pm1 (4).

https://toar-data.fz-juelich.de/api/v2/search/?bounding_box=49,7,50,8&variable_id=5,4 will get you all the time series within an area between 49°N 7°E and 50°N 8°E that record ozone or pm1.

INDICES AND TABLES

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