# Documentation

June 2023

## List of Figures



## Contents

# 1 Aasx Server Blazor

### Installation & Binaries

1. Install .NET Core 3.1 runtime. See `https://dotnet.microsoft.com/download/dotnet-core/3.1`
2. Download and install `https://github.com/admin-shell-io/aasx-server/releases/download/v2022-07-25.alpha/AasxServerBlazor.2022-07-25.alpha.zip`
3. To deploy the binaries, simply extract the release bundle (AasxServer-Core.zip) somewhere on your system
4. Copy your desired admin shells (.aasx files) to the aasxs subdirectory as needed

### Running on Windows

1. Change to the directory where you extracted the release bundle
2. Invoke the executable AasxsServerBlayor.exe:

```
AasxsServerBlayor.exe – rest –no-security –data-path aasxs –port 51310
```

1. You can see the UI in the browser: `http://localhost:5001/`
2. You can see the AAS on the server with:`http://localhost:51310/server/listaas`
3. To show the JSON files please use, e.g.: `http://localhost:51310/aas/Mold`
4. To show the submodel "Nameplate" please use: `http://localhost:51310/aas/Mold/submodels/Nameplate/complete`

### Submodel Element Interface

1. In order to modify a particular property use e.g. Postman API Platform to send API requests on the server (see image 1

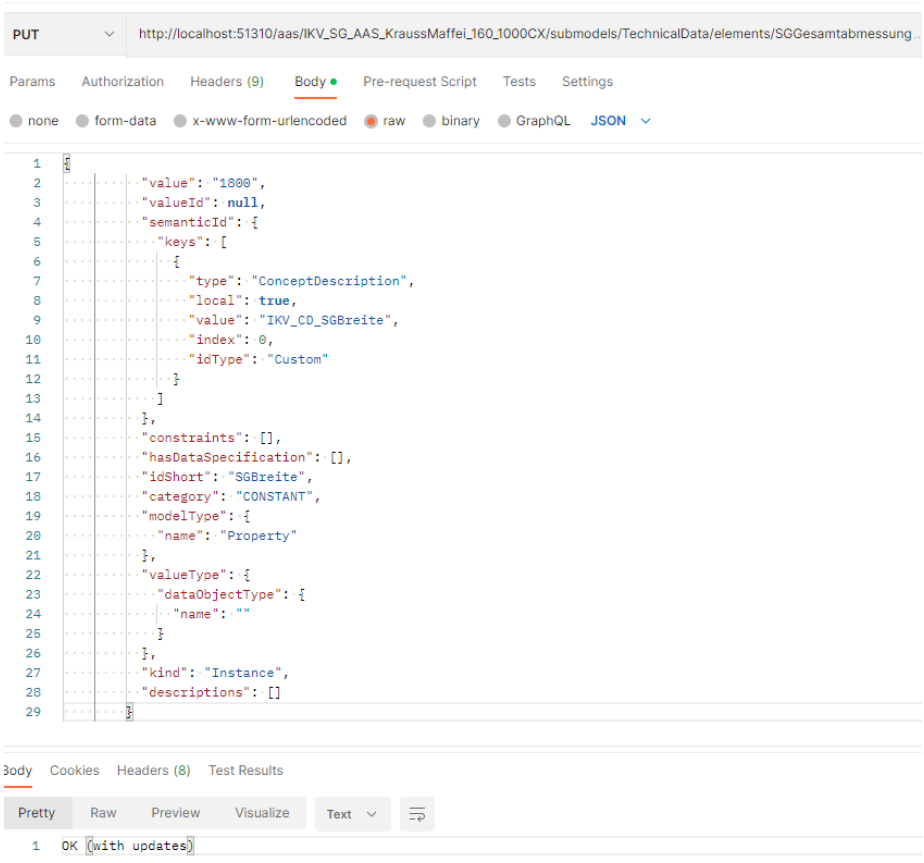


Figure 1: Payload

# 2 Neo4j Desktop DB Configuration

**Reading from a file**

The configuration setting

```
}apoc.import.file.enabled=true
```

is used in Neo4j to enable the import of data from external files using the APOC library. By default, this setting is disabled (false), and enabling it allows you to use APOC procedures to import data from files. It can be enabled by following the steps below:

1. Install APOC Library
   - Once you have a Neo4j database in your project, click on the database to select it. In the database details panel, click on *Manage*.
   - In the *Manage* section, you will see a list of available plugins. Look for *APOC* and click on the "Install" button next to it.
   - Neo4j Desktop will automatically download and install the *APOC* library for your selected database.
2. Launch Neo4j Desktop and open the project containing the Neo4j database you want to configure
3. In the project view, click on the database you want to configure. This will open the database details page
4. On the database details page, click on the *Manage* button for the database (three vertical dots 2)
5. From the dropdown menu, select *Open Folder* (it will open the file explorer at the location where the database files are stored)
6. In the opened file explorer, locate the *Configuration* folder (it contains the configuration files for the Neo4j database)
7. Open the *conf* folder, and you should find a file named *apoc.conf* [1]
8. Create the apoc.conf folder if it doesn't exist and set the following property in it:

   ```
   apoc.import.file.enabled=true
   ```

[1] There is no apoc.conf file in conf folder in version 5.6.0

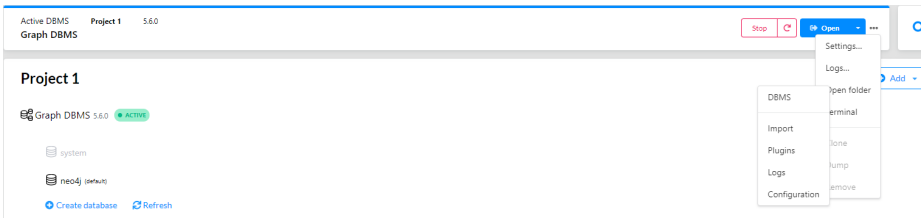


Figure 2: Configuration

Cypher commands:

```
1. Delete all nodes / specific node:
MATCH (n)
DETACH DELETE n

MATCH (n)
WHERE id(n) = 1
DELETE n
```

# 3 DB Graphical Representation

The whole query for building and executing the neo4j graphical database is saved to `db.cypher` file.

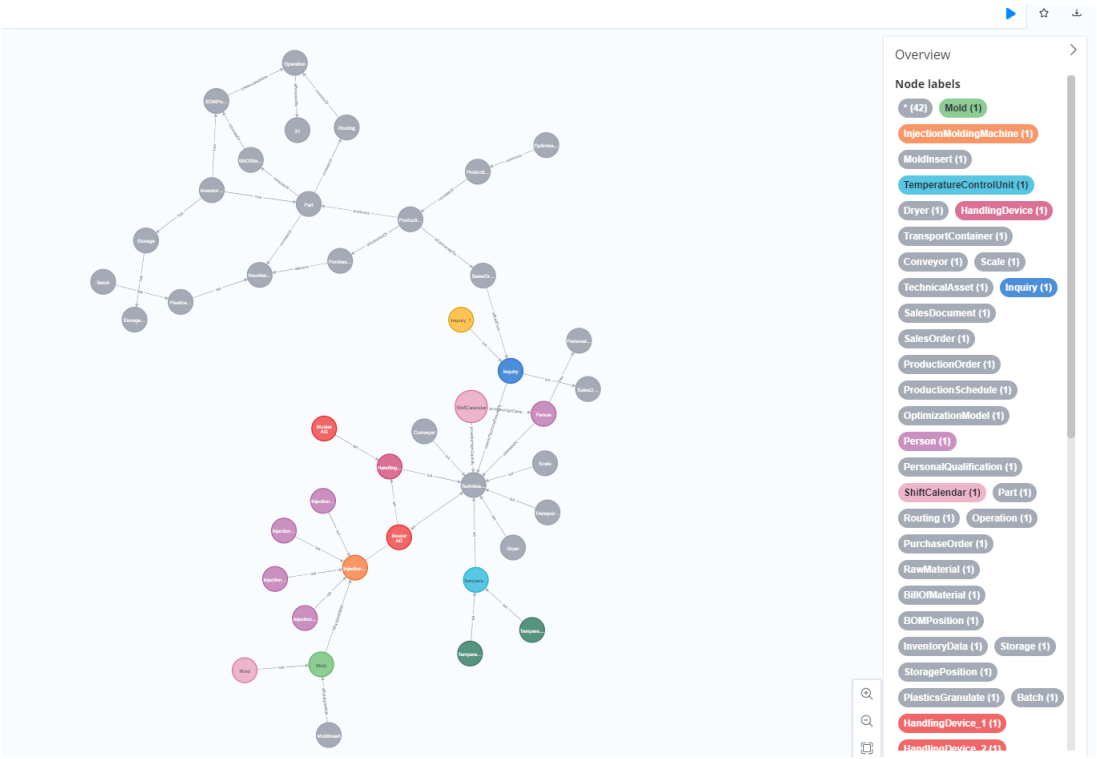


Figure 3: DB graphical representation

# 4 Running the GUI

1. Create a new directory for your project:

```
mkdir YourProjectName
cd YourProjectName
```

2. Create a virtual environment:

```
pthon -m venv venv
```

3. Activate the virtual environment: - On Windows:

```
.\venv\Scripts\activate
```

4. Clone the Git repository:

```
git clone https://github.com/psapel/AAS-neo4j-Database.git
```

5. Install packages within the virtual environment:

```
pip install package_name; flask, neo4j, python-dot-env, py2neo //
pip install -r requirements.txt
```

6. Do

```
set FLASK_APP=app.py \\
flask run --port=5001
```

# 5 MinIO

Initial Setup:

- Download the Minio server from the Minio website, i.e. `https://min.io/docs/minio/windows/index.html`
- Extract the downloaded Minio archive to a folder of your choice

Start Minio Server:

- Open a terminal in the directory where Minio is extracted and start the Minio Server with the default credentials as seen in Figure 4.

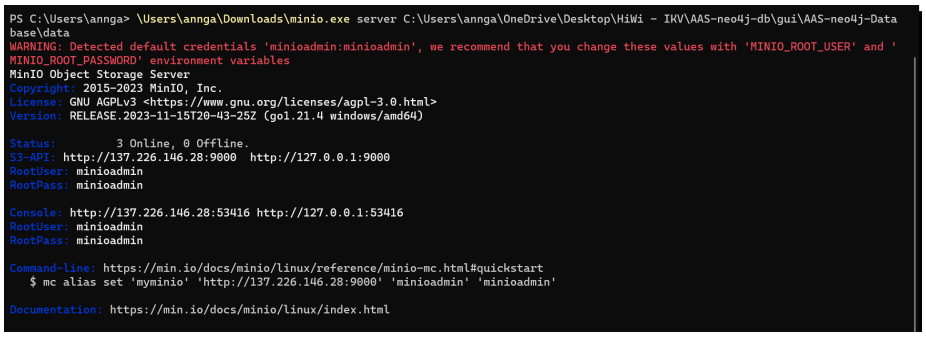


Figure 4: Minio Object Store

[2]

Using Minio Web Console:

- Open the web browser and navigate to the *MinIO* console. The default address is `http://127.0.0.1:9000`
- Log in using the access key and secret key. The default credentials are *minioadmin:minioadmin*
- Once logged in, you should see the MinIO web console
- In the console, find the *Create Bucket* option and enter the bucket name (i.e., query-bucket) and click the *Create* button
- Manually upload the query folder, containing the python files, to the bucket.

Alternatively,

- Store the queries to MinIO by running the script `file_uploader.py`
- Retrieve the queries from MinIO by running the script `file_downloader.py`

[2]data: The directory where Minio will store data