

# SampleDB: A sample and measurement metadata database

Florian Rhiem<sup>1</sup>

<sup>1</sup> PGI/JCNS-TA, Forschungszentrum Jülich

DOI: [10.21105/joss.01949](https://doi.org/10.21105/joss.01949)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Pending Editor](#) ↗

Submitted: 10 December 2019

Published: 12 December 2019

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

## Summary

One of the key aspects of good scientific practice is the handling of research data (German Research Foundation, 2019). Archiving research data and making it accessible to other researchers is crucial for reproducibility and can also yield new findings. This is not only true for research data resulting from experiments or simulations, but particularly for information on how a sample was created, how a measurement was performed and which parameters were used for a simulation.

SampleDB is a web-based sample and measurement metadata database developed at Jülich Centre for Neutron Science (JCNS) and Peter Grünberg Institute (PGI). Researchers can use SampleDB to store and retrieve information on samples, measurements and simulations, analyze them using Jupyter notebooks, track sample storage locations and responsibilities and view sample life cycles.

The application was designed to support the wide variety of instruments and processes found at PGI, JCNS and other institutes by allowing users to define the metadata of new processes using a graphical editor or a JSON-based schema language. These schemas can contain common datatypes such as booleans, texts and arrays, but also more specialized datatypes such as physical quantities and references to existing samples. Using these schemas, the application can generate forms for entering the information and then validate the information, including support for using regular expressions as constraints for text input. By storing the information using the JSONB datatype of PostgreSQL (The PostgreSQL Global Development Group, 2019), the built-in search function can support complex expressions and search for quantities with differing units but equal dimensionality.

In addition to the process-specific metadata, users can also store other information, including the location of a sample, auxiliary files, publications and additional comments. To provide a complete history of the metadata on an object, all previous versions are stored and can be accessed by users.

A Web API allows automated data entry using already existing information, e.g. by integrating it into an instrument control system or by monitoring log files, thereby archiving the information without additional overhead for the scientists. The Web API can also be used for automated data retrieval, e.g. for accessing measurement parameters during data analysis.

The latter is further simplified if SampleDB is used in conjunction with a JupyterHub server. Instrument scientists can provide Jupyter notebook templates for analyzing data from their instruments, along with a list of schema entries containing the necessary metadata. Users can then use these templates to create Jupyter notebooks from within SampleDB, which will copy the required metadata from its SampleDB entry into the notebook, preparing a process-specific data analysis.

SampleDB was developed using Python 3, the Flask web framework (Ronacher, 2019) and the SQLAlchemy package (Bayer, 2019). To run SampleDB, we recommend using the provided

[Docker images](#) along with a PostgreSQL container, though it can also be run directly from source. A guide on setting up a SampleDB instance can be found on the [GitHub project site](#).

## Acknowledgements

We thank Dorothea Henkel for her contributions, Daniel Kaiser for code review, and Paul Zakalek and Jörg Perßon for their feedback during the development of SampleDB.

## References

- Bayer, M. (2019). SQLAlchemy. Retrieved December 10, 2019, from <https://pypi.org/project/SQLAlchemy/>
- German Research Foundation. (2019). Handling of research data. Retrieved December 10, 2019, from [https://www.dfg.de/en/research\\_funding/proposal\\_review\\_decision/applicants/research\\_data/index.html](https://www.dfg.de/en/research_funding/proposal_review_decision/applicants/research_data/index.html)
- Ronacher, A. (2019). Flask. Retrieved December 10, 2019, from <https://pypi.org/project/Flask/>
- The PostgreSQL Global Development Group. (2019). PostgreSQL. Retrieved December 10, 2019, from <https://www.postgresql.org/>