The Journal of Open Source Software

# SampleDB: A sample and measurement metadata database

**Florian Rhiem**[1]

**1** PGI/JCNS-TA, Forschungszentrum Jülich

**DOI:** 10.21105/joss.02107

**Software**

- Review
- Repository
- Archive

**Editor:** Arfon Smith

**Reviewers:**

- @stuartcampbell
- @dvanic

**Submitted:** 10 December 2019
**Published:** 10 February 2021



## 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 findable, accessible, interoperable and re-usable by other researchers is crucial for reproducibility and can also yield new findings (Wilkinson et al., 2016). 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.

Using a schema system for process-specific metadata allows `SampleDB` to support a wide variety of processes and presents an advantage over alternative applications, such as the

JuliaBase project (Bronger, 2015), which define instruments and processes as part of the application code. That approach inherently limits the group of users that can define or alter processes to those users with administrative access. Keeping application logic and process-specific metadata separate has the disadvantage that SampleDB cannot provide built-in data analysis tools the way other applications such as the JuliaBase project or BikaLIMS (Bika LIMS Collective, 2020) / Senaite (RIDING BYTES GmbH & NARALABS S.L., 2020) can, however the support for Jupyter notebook templates can be used as an alternative to these data analysis tools. Using pre-defined calculations as they are offered by BikaLIMS / Senaite could also be explored for SampleDB in the future.

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. Information on setting up a SampleDB instance can be found on the GitHub project site and a more in-depth guide can be found in the project documentation.

## 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*. https://pypi.org/project/SQLAlchemy/

Bika LIMS Collective. (2020). *The bika open source LIMS project*. https://www.bikalims.org/

Bronger, T. (2015). *The JuliaBase project*. https://juliabase.org/project.html

German Research Foundation. (2019). *Handling of research data*. https://www.dfg.de/en/research_funding/proposal_review_decision/applicants/research_data/index.html

RIDING BYTES GmbH, & NARALABS S.L. (2020). *SENAITE enterprise open source laboratory system*. https://www.senaite.com/

Ronacher, A. (2019). *Flask*. https://pypi.org/project/Flask/

The PostgreSQL Global Development Group. (2019). *PostgreSQL*. https://www.postgresql.org/

Wilkinson, M. D., Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., Silva Santos, L. B. da, Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., … Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, *3*(160018). https://doi.org/10.1038/sdata.2016.18