The Journal of Open Source Software

`optimade-python-tools`: a Python library for serving and consuming materials data via OPTIMADE APIs

Matthew L. Evans[*][1, 2], **Casper W. Andersen**[†][3], **Shyam Dwaraknath**[4], **Markus Scheidgen**[5, 6], **Ádám Fekete**[1, 8, 9], **and Donald Winston**[4, 7]

1 Institut de la Matière Condensée et des Nanosciences, Université catholique de Louvain, Chemin des Étoiles 8, Louvain-la-Neuve 1348, Belgium **2** Theory of Condensed Matter Group, Cavendish Laboratory, University of Cambridge, J. J. Thomson Avenue, Cambridge, CB3 0HE, United Kingdom **3** Theory and Simulation of Materials (THEOS), Faculté des Sciences et Techniques de l'Ingénieur, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland **4** Lawrence Berkeley National Laboratory, Berkeley, CA, USA **5** Fritz-Haber-Institut der Max-Planck-Gesellschaft, Faradayweg 4-6, 14195, Berlin, Germany **6** Humboldt-Universität zu Berlin, Institut für Physik and IRIS Adlershof, 12489 Berlin, Germany **7** Polyneme LLC, New York, NY, USA **8** Department of Physics, King's College London, Strand, London WC2R 2LS, United Kingdom **9** Department of Physics and Namur Institute of Structured Materials, University of Namur, Rue de Bruxelles 51, 5000 Namur, Belgium

DOI: 10.21105/joss.03458

Software

- Review
- Repository
- Archive

Editor: Jeff Gostick

Reviewers:

- @hungpham2017
- @jamesrhester

Submitted: 14 June 2021
Published: 22 September 2021

Summary

In recent decades, improvements in algorithms, hardware, and theory have enabled crystalline materials to be studied computationally at the atomistic level with great accuracy and speed. To enable dissemination, reproducibility, and reuse, many digital crystal structure databases have been created and curated, ready for comparison with existing infrastructure that stores structural characterizations (e.g., diffraction) of real crystals. Each database will typically have a bespoke, stateless, web-based Application Programming Interface (API); users can submit a query via specially-crafted URLs. Such esoteric and specialized APIs incur maintenance and usability costs upon both the data providers and consumers, who may not be software specialists.

The OPTIMADE API specification (Andersen et al., 2020, 2021), released in July 2020, aimed to reduce these costs by designing a common API for use across a consortium of collaborating materials databases and beyond. Whilst based on the robust JSON:API standard (Katz et al., 2015), the OPTIMADE API specification presents several domain-specific features and requirements that can be tricky to implement for non-specialist teams. The repository presented here, `optimade-python-tools`, provides a modular reference server implementation and a set of associated tools to accelerate the development process for data providers, toolmakers and end-users.

Statement of need

In order to accommodate existing materials database APIs, the OPTIMADE API specification allows for flexibility in the specific data served, but enforces a simple yet domain-specific filter language on well-defined resources. However, this flexibility could be daunting to database providers, likely acting to increase the barrier to hosting an OPTIMADE API.

*corresponding author
†co-first author

optimade-python-tools aims to catalyse the creation of APIs from existing and new data sources by providing a configurable and modular reference server implementation for hosting materials data in an OPTIMADE-compliant way. The repository hosts the optimade Python package, which leverages the modern Python libraries pydantic (Colvin & others, 2021) and FastAPI (Ramírez & others, 2021) to specify the data models and API routes defined in the OPTIMADE API specification, additionally providing a schema following the OpenAPI format (Miller et al., 2021). As this package was developed concomitantly with the OPTIMADE specification itself, the present authors are not aware of any other generic packages with similar functionality. Two storage back-ends are supported out of the box, with full filter support for databases that employ the popular MongoDB or Elasticsearch frameworks.

Functionality

The modular functionality of optimade can be broken down by the different stages of a user query to the reference server. Consider the following query URL to an OPTIMADE API, which should filter for any crystal structures in the database with a composition that consists of any three elements in a 1:1:1 ratio:

```
https://example.org/v1/structures?filter=chemical_formula_anonymous="ABC"
```

1. After routing the query to the appropriate `/structures/` endpoint adhering to version `v1` of the specification, the filter string `chemical_formula_anonymous="ABC"` is tokenized and parsed into an abstract tree by a `FilterParser` object using the Lark parsing library (Shinan & others, 2021) against the formal grammar defined by the specification.
2. The abstract tree is then transformed by a `FilterTransformer` object into a database query specific to the configured back-end for the server. This transformation can include aliasing and custom transformations such that the underlying database format can be accommodated.
3. The results from the database query are then de-serialized by `EntryResourceMapper` objects into the OPTIMADE-defined data models and finally re-serialized into JSON before being served to the user over HTTP.

Beyond this query functionality, the package also provides:

- A fuzzy implementation validator that performs HTTP queries against remote or local OPTIMADE APIs, with test queries and expected responses generated dynamically based on the data served at the introspective `/info/` endpoint of the API implementation.
- Entry "adapters" that can convert between OPTIMADE-compliant entries and the data models of popular Python libraries used widely in the materials science community: `pymatgen` (Ong et al., 2013), ASE (Larsen et al., 2017), AiiDA (Huber et al., 2020), and JARVIS (Choudhary et al., 2020).

Use cases

The package is currently used in production by three major data providers for materials science data:

- The Materials Project (Jain et al., 2013) uses `optimade-python-tools` alongside their existing API (Ong et al., 2015) and MongoDB database, providing access to highly-curated density-functional theory calculations across all known inorganic materials. `optimade-python-tools` handles filter parsing, database query generation and response validation by running the reference server implementation with minimal configuration.
- NOMAD (Ghiringhelli et al., 2017) uses `optimade-python-tools` as a library to extend its existing web app with OPTIMADE API routes. It uses the Elasticsearch implementation to filter millions of structures from published first-principles calculations provided by users and other projects. NOMAD also uses the filtering module in its own API to expose the OPTIMADE filter language in the user-centric web interface search bar. NOMAD uses a released version of `optimade-python-tools` and all necessary customization can be realized via configuration and sub-classing.
- Materials Cloud (Talirz et al., 2020) uses `optimade-python-tools` as a library to provide an OPTIMADE API entry to archived computational materials studies, created with the AiiDA (Huber et al., 2020) Python framework and published through their archive. In this case, each individual study and archive entry has its own database and separate API entry. The Python classes within the `optimade` package have been extended to make use of AiiDA and its underlying PostgreSQL storage engine.
- The `optimade.adapters` module from the `optimade-python-tools` library is used in a graphical web client hosted on Materials Cloud (Andersen, 2021). It allows users to query OPTIMADE API implementations using user-friendly widgets as well as raw filter strings. The client uses the registry of known OPTIMADE providers to allow easy switching between databases. The crystal structures returned can be inspected visually and either downloaded in formats provided by conversion functions in the `optimade.adapters` module, or used seamlessly within other Materials Cloud web tools, where the structure is automatically validated and transferred in the background, partly using the `optimade.adapters` module.

Acknowledgements

All authors acknowledge contributions and feedback from other members of the OPTIMADE consortium, with special thanks to Michael Wu, Leopold Talirz, Thomas Purcell, Abhijith Gopakumar, Andrius Merkys and Fawzi Mohamed for their direct contributions to the `optimade` package. M.L.E. would like to acknowledge the EPSRC Centre for Doctoral Training in Computational Methods for Materials Science for funding under grant number EP/L015552/1 and support from the European Union's Horizon 2020 research and innovation program under the European Union's Grant agreement No. 951786 (NOMAD CoE). C.W.A. acknowledges financial support by the MARKETPLACE project, which is funded by Horizon 2020 under H2020-NMBP-25-2017 call with Grant agreement number: 760173 as well as the National Centres of Competence in Research (NCCR) Materials' revolution: Computational Design and Discovery of Novel Materials (MARVEL) created by the Swiss National Science Foundation (SNSF). S.D. acknowledges financial support by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, Materials Sciences and Engineering Division under Contract No. DE-AC02-05-CH11231 (Materials Project program KC23MP). M.S. acknowledges support from the European Union's Horizon 2020 research and innovation program under the European Union's Grant agreement No. 676580 (NoMaD) and No. 951786 (NOMAD CoE) as well as financial support from the Max Planck research network on big-data-driven materials science (BiGmax). A.F. acknowledges support from the Communauté française de Belgique through the SURFASCOPE project (ARC 19/24-102).

Andersen, C. W. (2021). *OPTIMADE-client* (Version 2021.9.10) [Computer software]. https://materialscloud.org/optimadeclient

Andersen, C. W., Armiento, R., Blokhin, E., Conduit, G., Dwaraknath, S., Evans, M. L., Fekete, Á., Gopakumar, A., Gražulis, S., Merkys, A., Mohamed, F., Oses, C., Pizzi, G., Rignanese, G.-M., Scheidgen, M., Talirz, L., Toher, C., & Winston, D. (2020). *The OPTIMADE specification* (Version 1.0) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.4195051

Andersen, C. W., Armiento, R., Blokhin, E., Conduit, G. J., Dwaraknath, S., Evans, M. L., Fekete, Á., Gopakumar, A., Gražulis, S., Merkys, A., Mohamed, F., Oses, C., Pizzi, G., Rignanese, G.-M., Scheidgen, M., Talirz, L., Toher, C., Winston, D., Aversa, R., … Yang, X. (2021). OPTIMADE, an API for exchanging materials data. *Sci Data*, *8*, 217. https://doi.org/10.1038/s41597-021-00974-z

Choudhary, K., Garrity, K. F., Reid, A. C. E., DeCost, B., Biacchi, A. J., Hight Walker, A. R., Trautt, Z., Hattrick-Simpers, J., Kusne, A. G., Centrone, A., Davydov, A., Jiang, J., Pachter, R., Cheon, G., Reed, E., Agrawal, A., Qian, X., Sharma, V., Zhuang, H., … Tavazza, F. (2020). The joint automated repository for various integrated simulations (JARVIS) for data-driven materials design. *Npj Comput Mater*, *6*, 173. https://doi.org/10.1038/s41524-020-00440-1

Colvin, S., & others. (2021). *Pydantic* (Version 1.8.2) [Computer software]. https://github.com/samuelcolvin/pydantic

Ghiringhelli, L. M., Carbogno, C., Levchenko, S., Mohamed, F., Huhs, G., Lüders, M., Oliveira, M., & Scheffler, M. (2017). Towards efficient data exchange and sharing for big-data driven materials science: Metadata and data formats. *Npj Comput Mater*, *3*(1), 1–9. https://doi.org/10.1038/s41524-017-0048-5

Huber, S. P., Zoupanos, S., Uhrin, M., Talirz, L., Kahle, L., Häuselmann, R., Gresch, D., Müller, T., Yakutovich, A. V., Andersen, C. W., Ramirez, F. F., Adorf, C. S., Gargiulo, F., Kumbhar, S., Passaro, E., Johnston, C., Merkys, A., Cepellotti, A., Mounet, N., … Pizzi, G. (2020). AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance. *Sci Data*, *7*, 300. https://doi.org/10.1038/s41597-020-00638-4

Jain, A., Ong, S. P., Hautier, G., Chen, W., Richards, W. D., Dacek, S., Cholia, S., Gunter, D., Skinner, D., Ceder, G., & Persson, K. A. (2013). The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials*, *1*(1), 011002. https://doi.org/10.1063/1.4812323

Katz, Y., Gebhardt, D., Sullice, G., Kellen, T., Klabnik, Steve, & Resnick, E. (2015). *The JSON:API specification* (Version 1.0). https://jsonapi.org/format/1.0/

Larsen, A. H., Mortensen, J. J., Blomqvist, J., Castelli, I. E., Christensen, R., Dułak, M., Friis, J., Groves, M. N., Hammer, B., Hargus, C., Hermes, E. D., Jennings, P. C., Jensen, P. B., Kermode, J., Kitchin, J. R., Kolsbjerg, E. L., Kubal, J., Kaasbjerg, K., Lysgaard, S., … Jacobsen, K. W. (2017). The atomic simulation environment—a Python library for working with atoms. *J. Phys.: Condens. Matter*, *29*(27), 273002. https://doi.org/10.1088/1361-648x/aa680e

Miller, D., Whitlock, J., Gardiner, M., Ralphson, M., Ratovsky, R., & Sarid, U. (Eds.). (2021). *The OpenAPI specification* (Version 3.1). https://spec.openapis.org/oas/v3.1.0

Ong, S. P., Cholia, S., Jain, A., Brafman, M., Gunter, D., Ceder, G., & Persson, K. A. (2015). The materials application programming interface (API): A simple, flexible and efficient API for materials data based on REpresentational state transfer (REST) principles. *Comp Mater Sci*, *97*, 209–215. https://doi.org/10.1016/j.commatsci.2014.10.037

Ong, S. P., Richards, W. D., Jain, A., Hautier, G., Kocher, M., Cholia, S., Gunter, D., Chevrier, V. L., Persson, K. A., & Ceder, G. (2013). Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. *Comp Mater Sci*, *68*, 314–319. https://doi.org/10.1016/j.commatsci.2012.10.028

Ramírez, S., & others. (2021). *FastAPI* (Version 0.65.1) [Computer software]. https://github.com/tiangolo/fastapi

Shinan, E., & others. (2021). *Lark - a parsing toolkit for python* (Version 0.11.3) [Computer software]. https://github.com/lark-parser/lark

Talirz, L., Kumbhar, S., Passaro, E., Yakutovich, A. V., Granata, V., Gargiulo, F., Borelli, M., Uhrin, M., Huber, S. P., Zoupanos, S., Adorf, C. S., Andersen, C. W., Schütt, O., Pignedoli, C. A., Passerone, D., VandeVondele, J., Schulthess, T. C., Smit, B., Pizzi, G., & Marzari, N. (2020). Materials cloud, a platform for open computational science. *Sci Data*, *7*, 299. https://doi.org/10.1038/s41597-020-00637-5