scientific data

OPEN

ARTICLE

# OPTIMADE, an API for exchanging materials data

Casper W. Andersen *et al.*#

**The Open Databases Integration for Materials Design (OPTIMADE) consortium has designed a universal application programming interface (API) to make materials databases accessible and interoperable. We outline the first stable release of the specification, v1.0, which is already supported by many leading databases and several software packages. We illustrate the advantages of the OPTIMADE API through worked examples on each of the public materials databases that support the full API specification.**

## Introduction

Data has become a crucial resource in many scientific fields, and materials science is no exception. Experimental data has long been meticulously curated in handbooks and databases, with the first edition of Landolt-Börnstein[1] being published in 1883. Nowadays, various commercial and non-commercial experimental databases, such as the Inorganic Crystal Structure Database (ICSD)[2], are widely used throughout the field.

High-throughput electronic structure calculations, themselves enabled by algorithmic improvements and growing computational resources, have significantly increased the availability of useful data from computational simulations of materials. Since the pioneering work of Ceder *et al.*[3], a large number of high-throughput first-principles studies have been reported in the literature (for a review, see ref. [4]), with results typically collated in databases. This explosion in the amount of available data has kick-started a new paradigm of data-driven materials science[5], creating opportunities for concurrent, automated materials design, boosted by databases that can be queried by humans and machines via an application programming interface (API)[6–9].

As materials databases differ in fidelity and focus across material classes and properties, it is extremely beneficial to be able to liberate and unify data from multiple sources. However, retrieving data from multiple databases is difficult as each database has its own specialized, and sometimes esoteric, API that governs data access patterns, querying and the representation of the underlying data. Moreover, as the APIs of individual databases inevitably evolve, existing clients must also evolve; a significant maintenance effort is required to translate the responses from the new API to the representation of the client.

Motivated by these considerations, providers of several materials databases united to design and implement an API specification that enables seamless access and interoperability across materials databases. The effort started at the workshop "Open Databases Integration for Materials Design", held at the Lorentz Center in Leiden, Netherlands in October 2016, and continued at followup workshops held at CECAM in Lausanne, Switzerland in June 2018, June 2019, and June 2020. The result is the OPTIMADE specification (v1.0)[10]; OPTIMADE defines a RESTful API that is queried with URLs, with responses adhering to the JSON:API specification[11]. Specification development adheres to Semantic Versioning[12] to avoid surprises and enable backwards-compatibility where possible, without impeding further development. By extracting the technical and scientific commonalities from existing APIs, the OPTIMADE API has been designed so that it can be implemented across a broad range of materials domains, database back-ends and sizes.

In this paper, we first review the query format of existing databases to motivate the design and construction of the OPTIMADE API specification. We then illustrate the use of the API with a set of worked examples; databases that already fully support the OPTIMADE API are enumerated alongside their results for representative queries in Table 1. We further highlight libraries that could accelerate uptake and assist materials data curators to support the OPTIMADE API format. Finally, we discuss future prospects and ongoing development of the OPTIMADE API.

#A full list of authors and their affiliations appears at the end of the paper.

| Provider | $N_1$ | $N_2$ | $N_3$ |
|---|---|---|---|
| AFLOW[26,27] | 700,192 | 62,293 | 382,554 |
| Crystallography Open Database (COD)[28,29] | 416,314 | 3,896 | 32,420 |
| Theoretical Crystallography Open Database (TCOD)[19] | 2,631 | 296 | 660 |
| Materials Cloud[9,16,17] | 886,518 | 801,382 | 103,075 |
| Materials Project[21,24,30,31] | 27,309 | 3,545 | 10,501 |
| Novel Materials Discovery Laboratory (NOMAD)[22,32] | 3,359,594 | 532,123 | 1,611,302 |
| Open Database of Xtals (odbx)[33] | 55 | 54 | 0 |
| Open Materials Database (*omdb*)[34] | 58,718 | 690 | 7,428 |
| Open Quantum Materials Database (OQMD)[35] | 153,113 | 11,011 | 70,252 |

**Table 1.** Materials databases with active OPTIMADE API implementations and the number of entries they return for the filters presented in this paper. The OPTIMADE website provides an up-to-date record of the implementation status of the databases (https://www.optimade.org/providers-dashboard/). AFLOW, Materials Project, odbx, *omdb*, and OQMD comprise computational materials data generated using database-specific workflows[21,27,33–35]. For the purposes of this table, Materials Cloud results were aggregated across all provided sub-databases. COD and TCOD comprise experimental and theoretical crystal structure data extracted from the literature. Materials Cloud comprises materials data from computational workflows; sub-databases group data by research project and can be contributed by users[9]. NOMAD aggregates computational data from multiple sources including from several of the repositories listed here.

## Current Generation of Materials Database APIs

Materials databases are a veritable treasure trove of information, but they only become useful once a human, or machine, can access them. In this section we review the current range of APIs used by various databases to enable access to an example compound, $SiO_2$, which serves to highlight the variation of APIs that a user must navigate in order to make use of multiple materials databases. We then demonstrate the universal nature of the OPTIMADE API that permits seamless access to all materials databases that support it.

We first compare and contrast the APIs that must be used to request records on an exemplar system, $SiO_2$, from three different databases: AFLOW, the Materials Project, and the Crystallographic Open Database (COD). All three queried databases support requests using a representational state transfer through a web service (RESTful), at the following URLs:

AFLOW http://aflow.org/API/aflux/?species(Si,O),nspecies(2)
Materials Project https://www.materialsproject.org/rest/v2/materials/SiO2/vasp/structure
COD https://www.crystallography.net/cod/result.php?formula=O2%20Si

Note that the Materials Project requires the user to supply an API key (http://materialsproject.org/open) preferably specified in the `X-API-KEY` HTTP header.

The three APIs vary syntactically (in format), taxonomically (having different names for terms), and semantically (in the conflicting definitions of chemical formula as an intensive or extensive property). AFLOW returns all structures with both Si and O present, whereas both the Materials Project and COD deliver any structure with a formula unit of $SiO_2$. The wide range of query formats that will deliver non-overlapping structures significantly complicates access to all available data for $SiO_2$, without even considering the differing representations of the structures returned.

The inconsistent format of the query is further complicated by the difficulty of accessing other structures with the $SiO_2$ formula. Focusing on just AFLOW, two possible queries that users more familiar with the other APIs might attempt are

http://aflow.org/API/aflux/?compound(SiO2)
which returns no response;
http://aflow.org/API/aflux/?compound(O2Si1)
now lists the elements in alphabetical order as required by AFLOW, and includes the "1" after element symbols, so that "SiO2" becomes "O2Si1". This returns entries where the unit cell is $SiO_2$, but does not return $Si_2O_4$ or simulation cells containing more formula units.

The exemplar http://aflow.org/API/aflux/?species(Si,O),nspecies(2) returns all entries with at least one Si and an O, so while the response includes the $SiO_2$ phases of interest, it may also contain other stoichiometries.

The distinctions between the request format for each database require the user to become an expert in many different APIs. This again emphasises the need for a single well-designed and standardized API to access all materials databases, which is the aim of the OPTIMADE API.

## The OPTIMADE API

The OPTIMADE API provides a holistic standard for serving and accessing the information in compatible materials databases. To retrieve information about materials from a particular database, the user submits a request via a URL. Each database provider will have published a base URL that serves the OPTIMADE API, for example https://example.com/optimade/. The same URL path, across different OPTIMADE API implementations, allows uniform access to the underlying databases. Both human-readable and machine-readable versions of the OPTIMADE API specification are available online with releases archived at Zenodo[10]. The specification is also registered as an API standard on FAIRsharing.org[13].

### Design philosophy.

The OPTIMADE specification strives to enable materials information to be filtered and retrieved in a straightforward and intuitive manner. The three queries from the previous section can each be performed on a standardised, versioned endpoint (`/v1/structures`) that enables access to a structures entry resource type that consists of many well-defined attributes. The specification then defines a grammar for filtering entries against these attributes, allowing the previous $SiO_2$ filter example to be expressed in a common way (`?filter=chemical_formula_reduced="O2Si"`). Altogether, the universal OPTIMADE URL, where only the implementation URL changes, becomes:

<optimade_implementation_url>/v1/structures?filter=chemical_formula_reduced="O2Si"

The OPTIMADE specification also aims to be flexible to many different underlying data representations, and thus there are very few properties that are mandatory. Instead of enforcing an exhaustive set of property definitions, individual OPTIMADE implementations can describe the data they serve via `/info` endpoints for each entry type. These introspective endpoints allow clients to adapt to the particular implementation for an underlying database, and allow providers to disseminate properties beyond the simple structural and chemical information standardized by the specification. To avoid naming collisions, each provider-specific property name must be prefixed by a provider-specific token, itself bookended by underscores (_). The property `custom_property` from an example provider with assigned prefix `exmpl` would be expressed as `_exmpl_custom_property`: e.g., `_tcod_a` for the lattice constant, *a*, in TCOD, or `_aflow_spacegroup_relax` for the space group of the relaxed structure in AFLOW.

### Implementation discovery.

The list of implementations confirmed and tested in this paper to support the OPTIMADE API is shown in Table 1. They are all publicly accessible, providing users with open access to large international repositories of computational and experimental materials science data.

The OPTIMADE consortium provides an open, federated list of implementations (https://providers.optimade.org). It is considered to be a catalogue of currently available and/or known public OPTIMADE implementations. New implementations are welcome and can register themselves via a pull request on GitHub (https://github.com/Materials-Consortia/providers).

The requirements for appearing in the above providers list are very loose. Some databases listed in the catalogue are signalling the intent of future implementations, while others only have partial implementations of the OPTIMADE API, including JARVIS (https://jarvis.nist.gov/optimade)[14] and MatCloud (https://matcloud.com.cn)[15]. Some software frameworks, such as AiiDA[16–18], also enable users to access their personal data through an OPTIMADE API, and therefore have a dedicated provider-specific ID, but no single official OPTIMADE implementation base URL.

The OPTIMADE API also specifies an endpoint for semi-automated cross-provider discovery. The `/links` endpoint serves `links` resources that may refer to either provider internal (`child, root`) or external (`external, providers`) resources based on the `link_type` attribute. To avoid being overly restrictive, it is at the provider's discretion whether they serve a list of known providers; however, this provides a mechanism for scalable and decentralised discovery of new implementations beyond the federated provider list.

## Worked Example

To illustrate the effective use of the OPTIMADE API we now provide a worked example of querying structures. We explore materials containing Group 14 elements (the carbon family), starting with a general search before drilling down to specific materials. The Group 14 elements are of particular interest as their atomic orbitals regularly hybridise, enabling a variety of bonding with differing geometries. The hybridised orbitals enable these elements to form the backbone of a wide range of compounds, both inorganic and organic, that underpin plastics, drugs, and semiconductors. Group 14 therefore forms both a diverse and important family of compounds that heavily populates databases, so are an ideal case study to demonstrate the OPTIMADE API.

### Common features of the response.

Whilst our previous exploration of the Group 14 compounds considered only $SiO_2$, the flexibility of the OPTIMADE API allows us to start with a search over all materials in Group 14, comprising carbon (C), silicon (Si), germanium (Ge), tin (Sn), and lead (Pb). We start with a simple API call that searches for all materials that contain at least one element in Group 14:

/v1/structures?filter=elements HAS ANY "C", "Si", "Ge", "Sn", "Pb"

This string can be appended to the base URL of any of the available implementations, to gather results in a standardised form. The base URL can be found on the providers dashboard (https://www.optimade.org/providers-dashboard).

As an example, this query is run through the Theoretical Crystallography Open Database (TCOD)[19] with the following URL:

https://www.crystallography.net/tcod/optimade/v1/structures?filter=elements+HAS+ANY+"C","Si","Ge","Sn","Pb"

The JSON response is summarized in Boxes 1 through 5, where some lines have been omitted for brevity; the full response is given in Supplementary File 1.

The first tranche of the JSON response comprises the "`data`" field that contains a list of entries returned for the query; a truncated version of this field is shown in Box 1, displaying a few salient properties of just one of the ten entries from the full response. The response for a particular material entry comprises multiple sections:

**Box 1** An excerpt of the JSON response showing the material attributes for one of the returned `structures` entries.

```
{
  "data": [
    {
      "attributes": {
```

```
        "dimension_types": [1, 1, 1],
        "elements": ["O", "Sn", "Ta"],
```

```
        "lattice_vectors": [
          [4.02510400822, 0, 0],
          [0, 4.02510400822, 0],
          [0, 0, 4.02510400822]
        ],
```

```
        "_tcod_a": 4.02510400822,
        "_tcod_alpha": 90,
        "_tcod_b": 4.02510400822,
        "_tcod_beta": 90,
        "_tcod_c": 4.02510400822,
```

`attributes`. Box 1 shows the physical properties of the material comprising both mandatory information such as `elements` and `lattice_vectors`, as well as optional, additional database-specific information prefixed with the database name (e.g., `_tcod_`, here used to provide lattice parameters). This ensures that all databases return the most important and common information in a standardized format, as well as allowing them to include additional database-specific data. Importantly, the OPTIMADE specification provides a standardized way for database implementations to be self-documenting, via introspective `/info` endpoints. We see in the `elements` section that here we have returned a material comprising the element of interest, Sn, as well as O and Ta.

**Box 2** An excerpt of the JSON response showing the top-level entry ID and a self-link.

```
        "id": "10000003",
        "links": {
          "self": "http://www.crystallography.net/tcod/10000003.cif"
        },
```

`id and links`. Box 2 shows the unique ID for the entry within the database, and a self-link to the database-specific representation/rendering of the entry (in this case, a link to a Crystallographic Information File[20]).

`relationships`. The relationships section in Box 3 links the user to other entries in the database and beyond, here the bibliographic references.

The additional nine materials not shown here all comprised of compounds containing either C, Si, Ge, Sn, or Pb, supplemented by a variety of other elements. The foot of the response contains information about the underlying database, comprised of three sections:

`links`. The response returned the first ten (i.e., the default page limit) entries that matched the query, however, more materials are available within the database. Box 4 shows the JSON:API-compliant pagination links to the current and next page of results for this query, as well as relevant external links.

**Box 3** An excerpt JSON data response for links to other entries related to this structure in the database (here, a bibliographic references entry).

```
        "relationships": {
          "references": {
            "data": {
              "id": "10000003",
              "type": "references"
            }
          }
        }
```

meta. Box 5 provides metadata associated with the request, such as number of results, the details of the database provider, the implementation and the representation and timestamp of the submitted query.

**Box 4** An excerpt of the JSON response for navigable pagination links to additional conformant materials that match the query.

```
  "links": {
    "base_url": "http://www.crystallography.net/tcod/optimade/v1.0.0/",
    "first": {
      "href": "http://www.crystallography.net/tcod/optimade/v1.0.0⌋
      ↪  /structures?page_limit=10&filter=elements HAS ANY
      ↪  \"C\",\"Si\",\"Ge\",\"Sn\",\"Pb\""
    },
    "next": {
      "href": "http://www.crystallography.net/tcod/optimade/v1.0.0⌋
      ↪  /structures?page_limit=10&page_offset=10&filter=elements HAS ANY
      ↪  \"C\",\"Si\",\"Ge\",\"Sn\",\"Pb\""
    }
  },
```

A benefit of the OPTIMADE API is that the structure of the response is common to all materials databases. The responses differ only in the optional and database-specific information prefixed with the database name (here _tcod_). Table 1 lists several materials databases that have active OPTIMADE API implementations, and the large number of results ($\mathbf{N_1}$) that they return for this particular filter.

**Database filtering.** Requesting the example filter above from the TCOD database returns 2,631 materials entries, but the same filter could return millions (see Table 1). For some requests the volume of the materials data could become unmanageable so the specification allows for the use of several pagination methods laid out by JSON:API[11]. These approaches all provide a link to the "next" page of data:

/v1/structures?page_limit=10&page_offset=10&filter=elements HAS ANY "C", "Si", "Ge", "Sn", "Pb"

with parameter page_offset = 10 to allow the user to select which page to enter, and the parameter page_limit = 10 to control the number of materials returned per individual request.

The most useful way to explore an OPTIMADE database is to apply a filter; the specification mandates that several relevant properties must be queryable. For example, we can perform a more focused search for materials comprising at least one element in Group 14, and a maximum of two elements (a binary material), with the filter

/v1/structures?filter=elements HAS ANY "C", "Si", "Ge", "Sn", "Pb" AND nelements=2

**Box 5** An excerpt of the JSON response showing the metadata returned for this request.

```
  "meta": {
    "api_version": "1.0.0",
    "data_available": 2906,
    "data_returned": 2631,
    "implementation": {
      "maintainer": {
        "email": "cod-bugs@ibt.lt"
      },
      "name": "Theoretical Crystallography Open Database",
      "source_url": "svn://www.crystallography.net/cod/tags/tcod/v0.2.1⌋
      ↪  /cgi-bin/optimade.pl@261229",
      "version": "v0.2.1"
    },
    "last_id": "10000031",
    "more_data_available": true,
    "provider": {
      "description": "Theoretical Crystallography Open Database",
      "name": "Theoretical Crystallography Open Database",
      "prefix": "tcod"
    },
    "query": {
      "representation": "/structures?filter=elements HAS ANY
      ↪  \"C\",\"Si\",\"Ge\",\"Sn\",\"Pb\""
    },
    "time_stamp": "2021-02-24T16:36:05Z"
  }
}
```

This query returns 296 materials from the TCOD database, with the response summarized in Box 6, where some lines have been omitted for brevity and the full response is given in the Supplementary File 2. The number of matching entries ($\mathbf{N_2}$) across all implementations for this filter are shown in Table 1.

**Box 6** The truncated JSON response for the more focused search, showing the elements in the first material returned.

```
{
   "data" : [
      {
         "attributes" : {

            "elements" : [
               "Ge",
               "O"
            ],

         },

      },

   ],

}
```

We can now see that the first structure, and indeed all structures, returned are comprised of at least one element in Group 14, here Ge, and a maximum of one other element (a binary material), here O. Additional filters can be chained to further refine the materials returned, or to construct more complex queries. For example, ternary structures that contain at least one of the elements C, Si, Ge, or Sn, but do not contain Pb (e.g., for applications where Pb toxicity would be a concern), can be retrieved using the filter

/v1/structures?filter=elements HAS ANY "C", "Si", "Ge", "Sn" AND NOT elements HAS "Pb" AND elements LENGTH 3

The number of entries matching this filter are denoted as $\mathbf{N_3}$ in Table 1.

These simple examples demonstrate how useful chemical queries are expressible with the OPTIMADE API, allowing users to refine their queries and to suit their specific application. Further functionality of the OPTIMADE API can be found in the specification[10].

## Related Libraries

The wider usage of the OPTIMADE API is a key goal for the consortium; to this end, several open source libraries have been developed to help users of the OPTIMADE API (either implementation developers, or client end-users), of which three are introduced below. The first two libraries offer tools that aid the implementation of the API for materials database developers, with the first also containing tools to construct and validate queries, while the third library is intended for end users of OPTIMADE-compliant APIs.

**optimade-python-tools.** optimade-python-tools is an open source Python package available on GitHub (https://github.com/Materials-Consortia/optimade-python-tools). The package contains a complete set of tools for implementing an OPTIMADE-compliant API, as well as several utilities that can be used by client code. The package is listed on the Python Package Index (PyPI) as optimade (https://pypi.org/project/optimade). Current (v0.14) functionality of the package includes:

- pydantic (https://github.com/samuelcolvin/pydantic) data and validation models for all objects defined in the OPTIMADE specification that can be used in server or client code;
- an extensible reference server implementation leveraging pydantic and FastAPI (https://github.com/tiangolo/fastapi). This reference server forms the basis of the OPTIMADE implementations for the Materials Project[21], NOMAD[22], Materials Cloud[9] and odbx (https://odbx.science) providers. These tools are also used to generate a machine-readable OpenAPI version[23] of the OPTIMADE specification;
- a validator for OPTIMADE implementations, available standalone with a command line interface (CLI) or as a GitHub action (https://github.com/Materials-Consortia/optimade-validator-action). This validator is run against the federated list of OPTIMADE providers every day, with a live dashboard indicating compliance with the specification at https://www.optimade.org/providers-dashboard/, the source code for which can be found on GitHub (https://github.com/Materials-Consortia/providers-dashboard);
- a Python parser for the OPTIMADE filter language written with the Lark parsing library (https://github.com/lark-parser/lark);

- filter transformers from the abstract syntax tree to queries to popular database back-ends. Currently, the supported back-ends are MongoDB (via pymongo) (https://github.com/mongodb) and Elasticsearch (https://github.com/elastic/elasticsearch);
- adapter classes to interface with other popular libraries, namely pymatgen[24], ASE[25], AiiDA[16–18] and JARVIS[14] as well as converting OPTIMADE structures to the CIF[20], XYZ, and more domain standardised file formats.

## OPTIMADE::Filter

OPTIMADE::Filter is a Perl library for the syntactical analysis of the OPTIMADE filter language. Apart from the construction of abstract syntax trees, the library can translate simple filter strings to SQL queries. The Git repository with the source code is publicly available on GitHub (https://github.com/Materials-Consortia/OPTIMADE-Filter).

**pymatgen optimade module.** pymatgen[24] (https://pymatgen.org) is a Python library for materials science. A user-friendly `OptimadeRester` client has been added to a new OPTIMADE module within pymatgen to provide a way to query OPTIMADE structure resources in a way familiar to existing users of pymatgen and the Materials Project API. The Git repository with the source code is publicly available on GitHub (https://github.com/materialsproject/pymatgen).

## Summary

The latest OPTIMADE API specification v1.0[10] offers holistic access to many leading crystal structure databases, namely: AFLOW, COD, TCOD, Materials Cloud, Materials Project, NOMAD, odbx, Open Materials Database (*omdb*), and OQMD. Open client implementations are also available (https://optimade.science, https://materialscloud.org/optimadeclient) that enable aggregated searches over many databases as well as user-friendly graphical widgets that can create an OPTIMADE filter to empower the user with even easier access to data. OPTIMADE provides researchers easy access to over 10000000 results for different materials, providing benchmarking opportunities and offering a huge opportunity for high-throughput screening and machine learning studies. The ability of the OPTIMADE API to search databases, expose links between databases, and deliver standardized results makes it well-positioned to significantly enhance the impact and permeability of pre-existing data silos. This should empower researchers to scan through new and unexpected material families, and train models from all available data that can understand deep correlations.

The OPTIMADE API is flexible and will be extended to more use cases going forward. The development and adoption of the OPTIMADE API relies on the involvement of a large number of scientists, so contributions from the community are strongly encouraged, and questions on development, registration of a provider, or usage can be directed to the web forum (https://matsci.org/optimade) or mailing list (dev@optimade.org). Proposed developments include the standardization of more filterable materials properties, the integration of molecular dynamics simulations and of experimental results, and extensions beyond electronic-structure calculations. The future development of APIs, including OPTIMADE, should herald an era of effective use of big, open data in materials science.

## Data availability

The OPTIMADE specification is developed openly on GitHub with releases archived on Zenodo[10]. Version 1.0.0 was released on 1 July 2020 and is licensed under Creative Commons Attribution 4.0 International (CC-BY 4.0).

## Code availability

All associated code is hosted under the Materials-Consortia organisation on GitHub (https://github.com/Materials-Consortia).

Received: 2 March 2021; Accepted: 23 June 2021;
Published online: 12 August 2021

## References

1. Landolt, H. & Bornstein, R. *Physikalisch-Chemische Tabellen* (Julius Springer, Berlin, 1883).
2. Zagorac, D., Müller, H., Ruehl, S., Zagorac, J. & Rehme, S. Recent developments in the inorganic crystal structure database: theoretical crystal structure data and related features. *J. Appl. Crystallogr.* **52**, 918–925, https://doi.org/10.1107/S160057671900997X (2019).
3. Ceder, G. *et al.* Identification of cathode materials for lithium batteries guided by first-principles calculations. *Nature* **392**, 694–696, https://doi.org/10.1038/33647 (1998).
4. Curtarolo, S. *et al.* The high-throughput highway to computational materials design. *Nat. Mater.* **12**, 191-201, https://doi.org/10.1038/nmat3568 (2013).
5. Himanen, L., Geurts, A., Foster, A. S. & Rinke, P. Data-Driven Materials Science: Status, Challenges, and Perspectives. *Adv. Sci.* **6**, 1900808, https://doi.org/10.1002/advs.201900808 (2019).
6. Taylor, R. H. *et al.* A RESTful API for exchanging materials data in the AFLOWLIB.org consortium. *Comput. Mater. Sci.* **93**, 178–192, https://doi.org/10.1016/j.commatsci.2014.05.014 (2014).
7. Rose, F. *et al.* AFLUX: The LUX materials search API for the AFLOW data repositories. *Comput. Mater. Sci.* **137**, 362–370, https://doi.org/10.1016/j.commatsci.2017.04.036 (2017).
8. Ong, S. P. *et al.* The Materials Application Programming Interface (API): A simple, flexible and efficient API for materials data based on REpresentational State Transfer (REST) principles. *Comput. Mater. Sci.* **97**, 209–215, https://doi.org/10.1016/j.commatsci.2014.10.037 (2015).
9. Talirz, L. *et al.* Materials Cloud, a platform for open computational science. *Sci. Data* **7**, 299, https://doi.org/10.1038/s41597-020-00637-5 (2020).
10. Andersen, C. W. *et al.* The OPTIMADE Specification (Version 1.0.0). *Zenodo* https://doi.org/10.5281/zenodo.4195051 (2020).
11. The JSON:API Specification (Version 1.0). https://jsonapi.org/format (2015).

12. Semantic Versioning (Version 2.0.0). https://semver.org/spec/v2.0.0.html (2013).
13. FAIRsharing.org: OPTIMADE; Open Databases Integration for Materials Design. https://doi.org/10.25504/FAIRsharing.xvfqAC (2020).
14. Choudhary, K. *et al.* 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 (2020).
15. Yang, X. *et al.* Matcloud: A high-throughput computational infrastructure for integrated management of materials simulation, data and resources. *Comput. Mater. Sci.* **146**, 319–333, https://doi.org/10.1016/j.commatsci.2018.01.039 (2018).
16. Pizzi, G., Cepellotti, A., Sabatini, R., Marzari, N. & Kozinsky, B. AiiDA: automated interactive infrastructure and database for computational science. *Comput. Mater. Sci.* **111**, 218–230, https://doi.org/10.1016/j.commatsci.2015.09.013 (2016).
17. Huber, S. P. *et al.* 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 (2020).
18. Uhrin, M., Huber, S. P., Yu, J., Marzari, N. & Pizzi, G. Workflows in AiiDA: Engineering a high-throughput, event-based engine for robust and modular computational workflows. *Comput. Mater. Sci.* **187**, 110086, https://doi.org/10.1016/j.commatsci.2020.110086 (2021).
19. Merkys, A. *et al.* A posteriori metadata from automated provenance tracking: integration of AiiDA and TCOD. *J. Cheminform.* **9**, 56, https://doi.org/10.1186/s13321-017-0242-y (2017).
20. Bernstein, H. J. *et al.* Specification of the crystallographic information file format, version 2.0. *J. Appl. Crystallogr.* **49**, 277–284, https://doi.org/10.1107/S1600576715021871 (2016).
21. Jain, A. *et al.* Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL Mater.* **1**, 011002, https://doi.org/10.1063/1.4812323 (2013).
22. Draxl, C. & Scheffler, M. NOMAD: The FAIR concept for big data-driven materials science. *MRS Bull.* **43**, 676–682, https://doi.org/10.1557/mrs.2018.208 (2018).
23. The OpenAPI Specification (Version 3.1.0). https://spec.openapis.org/oas/v3.1.0.html (2021).
24. Ong, S. P. *et al.* Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. *Comput. Mater. Sci.* **68**, 314–319, https://doi.org/10.1016/j.commatsci.2012.10.028 (2013).
25. Larsen, A. H. *et al.* The atomic simulation environment—a python library for working with atoms. *J. Phys. Condens. Matter* **29**, 273002, https://doi.org/10.1088/1361-648x/aa680e (2017).
26. Curtarolo, S. *et al.* AFLOWLIB.ORG: A distributed materials properties repository from high-throughput *ab initio* calculations. *Comput. Mater. Sci* **58**, 227–235, https://doi.org/10.1016/j.commatsci.2012.02.002 (2012).
27. Toher, C. *et al.* The AFLOW fleet for materials discovery. In Andreoni, W. & Yip, S. (eds.) *Handbook of Materials Modeling*, 1–28, https://doi.org/10.1007/978-3-319-42913-7_63-1 (Springer International Publishing, Cham, Switzerland, 2018).
28. Gražulis, S. *et al.* Crystallography Open Database - an open-access collection of crystal structures. *J. Appl. Crystallogr* **42**, 726–729, https://doi.org/10.1107/S0021889809016690 (2009).
29. Gražulis, S. *et al.* Crystallography Open Database (COD): an open-access collection of crystal structures and platform for world-wide collaboration. *Nucleic Acids Res.* **40**, D420–D427, https://doi.org/10.1093/nar/gkr900 (2012).
30. Jain, A. *et al.* A high-throughput infrastructure for density functional theory calculations. *Comput. Mater. Sci.* **50**, 2295–2310, https://doi.org/10.1016/j.commatsci.2011.02.023 (2011).
31. Mathew, K. *et al.* Atomate: A high-level interface to generate, execute, and analyze computational materials science workflows. *Comput. Mater. Sci.* **139**, 140–152, https://doi.org/10.1016/j.commatsci.2017.07.030 (2017).
32. Ghiringhelli, L. M. *et al.* Towards efficient data exchange and sharing for big-data driven materials science: metadata and data formats. *npj Computational Materials* **3**, 46, https://doi.org/10.1038/s41524-017-0048-5 (2017).
33. Evans, M. L. & Morris, A. J. matador: a Python library for analysing, curating and performing high-throughput density-functional theory calculations. *J. Open Source Softw.* **5**, 2563, https://doi.org/10.21105/joss.02563 (2020).
34. Armiento, R. Database-Driven High-Throughput Calculations and Machine Learning Models for Materials Design. In Schütt, K. T. *et al.* (eds.) *Machine Learning Meets Quantum Physics*, Lecture Notes in Physics, 377–395, https://doi.org/10.1007/978-3-030-40245-7_17 (Springer International Publishing, Cham, 2020).
35. Saal, J. E., Kirklin, S., Aykol, M., Meredig, B. & Wolverton, C. Materials design and discovery with high-throughput density functional theory: The Open Quantum Materials Database (OQMD). *JOM* **65**, 1501–1509, https://doi.org/10.1007/s11837-013-0755-4 (2013).

## Acknowledgements

The authors acknowledge support from CECAM in Lausanne (Switzerland) and the Lorentz Center in Leiden (Netherlands) for hosting OPTIMADE workshops. C.W.A., S.K., N.M., G.Pi., M.U., and L.T. acknowledge financial support by the MARVEL National Centre of Competence in Research funded by the Swiss National Science Foundation (grant agreement ID 51NF40-182892), the European Centre of Excellence MaX "Materials design at the Exascale" (grant no. 824143) and by the the "MARKETPLACE" H2020 project (grant agreement ID 760173). R.Ar. acknowledges financial support from the Swedish e-Science Research Centre (SeRC) and the Swedish Research Council (VR) project no. 2016-04810 and 2020-05402. R.Av. has received funding from the European Union's Horizon 2020 Research and Innovation Program under grant agreement No. 654360 NFFA-EUROPE. G.J.C. would like to acknowledge financial support from the Royal Society. S.D., D.Wi., M.Wu., M.H., P.H., and K.P. acknowledge 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). C.D., M.L.E., G.H., G.-M.R., M.Schef., M.Schei., and D.Wa. acknowledge support from the European Union's Horizon 2020 research and innovation program under the European Union's Grant agreement No. 951786 (NOMAD CoE). M.L.E. acknowledges the EPSRC CDT in Computational Methods for Materials Science for funding under grant number EP/L015552/1. A.G., M.L., V.H., and C.W. acknowledges support by Toyota Research Institute, and the award 70NANB14H012 from U.S. Department of Commerce, National Institute of Standards and Technology as part of the Center for Hierarchical Materials Design (CHiMaD). M.Go. acknowledges support by MICCoM, as part of the Computational Materials Sciences Program funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, Materials Sciences and Engineering Division. S.G., A.Me., and A.V. were supported by funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 689868 (SOLSA). A.J.M. acknowledges networking support via the EPSRC Collaborative Computational Projects, CCP9 (EP/M022595/1) and CCP-NC (EP/T026642/1) and the EPSRC High-End Computing Consortium, UKCP (EP/P022561/1). G.-M.R. acknowledges support from the F.R.S.-FNRS. MSchei acknowledges support from the European Union's Horizon 2020 research and innovation program under the European Union's Grant agreement No. 676580 (NoMaD) and financial support from the Max Planck research network on big-data-driven materials science (BiGmax). C.O., C.T., P.C., M.Es., F.Ro., and S.C. acknowledge support by DOD-ONR (N00014-17-1-2090, N00014-17-1-2876) and by the National Science Foundation under DMREF Grant No. DMR-1921909. X.Y. acknowledges support from the National Key Research and Development Program of China (Grants No. 2017YFB0701702 and 2016YFB0700501).

## Author contributions

All authors participated in discussions that lead to the OPTIMADE specification version 1.0, in particular, by taking part to the OPTIMADE workshops that took place at the Lorentz Center in Leiden (Netherlands) in 2016, and at the CECAM in Lausanne (Switzerland) in 2018, 2019, and 2020. They also revised and approved the final version of the paper. C.W.A., R.Ar., E.B., G.J.C., S.D., M.L.E., A.F., A.G., S.G., A.Me., F.M., C.O., G.Pi., G.-M.R., M.Schei., L.T., C.T., and D.Wi. provided the key ideas for the design of the API, took part to the discussions about the issues opened on the Materials Consortia GitHub repositories, writing its specifications, and prepared the paper. C.W.A. and M.L.E., with help from A.F., S.D., and M.Schei., write and maintain the optimade-python-tools library, which is based on initial work by D.Wi. and M.Wu., with contributions from R.Ar., K.C., A.G., A.Me., F.M., L.T., and T.P. This library forms the basis of the Materials Cloud, Materials Project, NOMAD and odbx providers as well as the client hosted on Materials Cloud. C.D., G.H., N.M., G.-M.R. and M.Schef. started the OPTIMADE workshop series, N.M. suggesting to dedicate these to the creation of a common API for databases of materials properties. R.Ar., G.J.C., C.D., S.G., N.M., G.H., G.Pi., G.-M.R., M.Schef., M.Schei., and C.T. were on the organizing committee. C.W.A. implemented the OPTIMADE server for the AiiDA framework and the Materials Cloud databases with contributions from G.Pi., L.T., and S.K. R.Ar. has contributed very actively to the OPTIMADE specification GitHub repository. He also implemented the OPTIMADE server in the high-throughput toolkit (*httk*) open source framework used for the *omdb* database. S.D. implemented the OPTIMADE server for the Materials Project provider, with assistance from M.K.H. and P.H.. M.K.H. wrote the OPTIMADE client in *pymatgen*. M.L.E. implemented the OPTIMADE server for the odbx provider, with assistance from A.J.M.. A.G. and M.L. implemented OPTIMADE server for OQMD provider with assistance from V.H., S.G., A.Me., and L.T. contributed to the implementation of the OPTIMADE::Filter Perl library for the syntactical analysis of the OPTIMADE filter language. M.Schei. implemented the OPTIMADE server for the NOMAD provider with contributions from F.M. C.T., C.O., P.C., M.E., F.Ro., and S.C. implemented the OPTIMADE server for the AFLOW provider. A.V. contributed to the long-term maintenance of the COD, to make it suitable for presentation via the OPTIMADE API.

## Competing interests

G.J.C. is a shareholder and Director of Intellegens Ltd. G.H., G.Pe., G.-M.R. and D.Wa. are shareholders and Directors of Matgenix SRL. The other authors declare no competing commercial interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at https://doi.org/10.1038/s41597-021-00974-z.

**Correspondence** and requests for materials should be addressed to G.-M.R.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Casper W. Andersen[1,34], Rickard Armiento[2,34], Evgeny Blokhin[3,4,34], Gareth J. Conduit[5,34], Shyam Dwaraknath[6,34], Matthew L. Evans[5,7,34], Ádám Fekete[7,8,34], Abhijith Gopakumar[9,34], Saulius Gražulis[10,11,34], Andrius Merkys[10,34], Fawzi Mohamed[12,34], Corey Oses[13,14,34], Giovanni Pizzi[1,34], Gian-Marco Rignanese[7,34] ✉, Markus Scheidgen[12,15,34], Leopold Talirz[1,16,34], Cormac Toher[13,14,34], Donald Winston[6,34], Rossella Aversa[17,18], Kamal Choudhary[19], Pauline Colinet[13,14], Stefano Curtarolo[13,14], Davide Di Stefano[20], Claudia Draxl[15], Suleyman Er[21], Marco Esters[13,14], Marco Fornari[13,22], Matteo Giantomassi[7], Marco Govoni[23], Geoffroy Hautier[7,24], Vinay Hegde[9], Matthew K. Horton[6], Patrick Huck[6], Georg Huhs[15], Jens Hummelshøj[25], Ankit Kariryaa[26],

**Boris Kozinsky[27,28], Snehal Kumbhar[1], Mohan Liu[9], Nicola Marzari[1], Andrew J. Morris[29], Arash A. Mostofi[30], Kristin A. Persson[6,31], Guido Petretto[7], Thomas Purcell[12], Francesco Ricci[7], Frisco Rose[13,14], Matthias Scheffler[12], Daniel Speckhard[12,15], Martin Uhrin[1], Antanas Vaitkus[10], Pierre Villars[4], David Waroquiers[7], Chris Wolverton[9], Michael Wu[6] & Xiaoyu Yang[32,33]**

[1]Theory and Simulation of Materials (THEOS), and National Centre for Computational Design and Discovery of Novel Materials (MARVEL), École Polytechnique Fédérale de Lausanne, 1015, Lausanne, Switzerland. [2]Materials Design and Informatics unit, Department of Physics, Chemistry and Biology, Linköping University, Linköping, Sweden. [3]Tilde Materials Informatics, Straßmannstraße 25, 10249, Berlin, Germany. [4]Materials Platform for Data Science, Sepapaja 6, 15551, Tallinn, Estonia. [5]Theory of Condensed Matter Group, Cavendish Laboratory, 19 J.J. Thomson Avenue, Cambridge, CB3 0HE, United Kingdom. [6]Lawrence Berkeley National Laboratory, 1 Cyclotron Rd, Berkeley, CA 94720, USA. [7]UCLouvain, Institut de la Matière Condensée et des Nanosciences (IMCN), Chemin des Étoiles 8, Louvain-la-Neuve, 1348, Belgium. [8]Department of Physics, King's College London, London, United Kingdom. [9]Department of Materials Science and Engineering, Northwestern University, Evanston, IL, 60208, USA. [10]Institute of Biotechnology, Life Science Center, Vilnius University, Saulėtekio av. 7, LT-10257, Vilnius, Lithuania. [11]Institute of Computer Science, Faculty of Mathematics and Informatics, Vilnius University, Naugarduko g. 24, LT-03225, Vilnius, Lithuania. [12]Fritz-Haber-Institut der Max-Planck-Gesellschaft, Faradayweg 4-6, 14195, Berlin, Germany. [13]Center for Autonomous Materials Design, Duke University, Durham, NC, 27708, USA. [14]Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC, 27708, USA. [15]Humboldt-Universität zu Berlin, Institut für Physik and IRIS Adlershof, 12489, Berlin, Germany. [16]Laboratory of Molecular Simulation (LSMO), École Polytechnique Fédérale de Lausanne, 1951, Sion, Switzerland. [17]National Research Council-Istituto Officina dei Materiali (CNR-IOM), 34136, Trieste, Italy. [18]Karlsruhe Institute of Technology (KIT), Hermann-von-Helmholtz-Platz 1, 76344, Eggenstein-Leopoldshafen, Germany. [19]Materials Measurement Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, 20899, USA. [20]Ansys, 300 Rustat House, 62 Clifton Rd, Cambridge, CB1 7EG, United Kingdom. [21]Dutch Institute for Fundamental Energy Research (DIFFER), De Zaale 20, 5612 AJ, Eindhoven, The Netherlands. [22]Department of Physics and Science of Advanced Materials Program, Central Michigan University, Mount Pleasant, Michigan, 48859, USA. [23]Argonne National Laboratory, Lemont, IL, 60439, USA. [24]Thayer School of Engineering, Dartmouth College, Hanover, NH, 03755, USA. [25]Toyota Research Institute (TRI), Los Altos, California, 94022, USA. [26]Department of Computer Science, University of Copenhagen, Universitetsparken 1, 2100, Copenhagen, Denmark. [27]John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts, 02138, USA. [28]Robert Bosch LLC, Research and Technology Center North America, 255 Main St, Cambridge, Massachusetts, 02142, USA. [29]School of Metallurgy and Materials, University of Birmingham, Edgbaston, Birmingham, B15 2TT, United Kingdom. [30]Departments of Materials and Physics, and the Thomas Young Centre, Imperial College London, Exhibition Road, London, SW7 2AZ, United Kingdom. [31]Department of Material Science and Engineering, Hearst Mining Memorial Building, UC Berkeley, Berkeley, CA, 94720, USA. [32]Computer Network Information Center, Chinese Academy of Sciences, Beijing, China. [33]University of Chinese Academy of Sciences, Beijing, China. [34]These authors contributed equally: Casper W. Andersen, Rickard Armiento, Evgeny Blokhin, Gareth J. Conduit, Shyam Dwaraknath, Matthew L. Evans, Ádám Fekete, Abhijith Gopakumar, Saulius Gražulis, Andrius Merkys, Fawzi Mohamed, Corey Oses, Giovanni Pizzi, Gian-Marco Rignanese, Markus Scheidgen, Leopold Talirz, Cormac Toher, Donald Winston. ✉e-mail: gian-marco.rignanese@uclouvain.be