OPTIMADE serialization format

Meeting 12 April 2023

Participants:

- Matthew Evans
- Kristjan Eimre
- Jusong Yu

Notes

- Use cases:
  - Start:
    - 1) AiiDA database
    - 2) archive of structures (e.g. a zip of cif files; uploaded to MC Archive.
  - Starting point -> [serialization format] -> start an performant optimade server
- Json lines:
  - 1) metadata
  - 2) structures (which format?)

You can serialize an OPTIMADE API to disk in JSONL format:

1. First entry must be an empty dictionary with the key “OPTIMADE”
2. Then comes metadata
3. Then corresponding info endpoint data for the types in the file
4. Then each line contains an entry from the corresponding endpoints

optimade_example.jsonl:

```
{“x-optimade”: {...}}
{“meta”: {“api_version”: “1.1.0”}}\n
{“info”: {“properties”: [“_aiida_cell_volume”: {“...”}}}\n
{“info/structures”: {“properties”: [“_aiida_cell_volume”: {“type”:
number}}}\n
{“info/references”: {...}}\n
{“data”: {“type”: “structures”, “id”: “1234”, “attributes”: {...}}\n
{“data”: {“type”: “structures”, “id”: “1234”, “attributes”: {...}}\n
{“data”: {“type”: “references”, “id”: “sfdas”, “attributes”: {...}}
```

Potential challenges:

- Need to annotate info response as e.g., info/structures
- Need to decide how much is mandatory for valid format vs necessary for re-use

Follow-up:

- Matthew to write script that takes optimade.odbx.science and creates such a file
- Matthew to find time to write SQL implementation to optimade-python-tools before June