Human Brain Project

EBRAINS

fairgraph

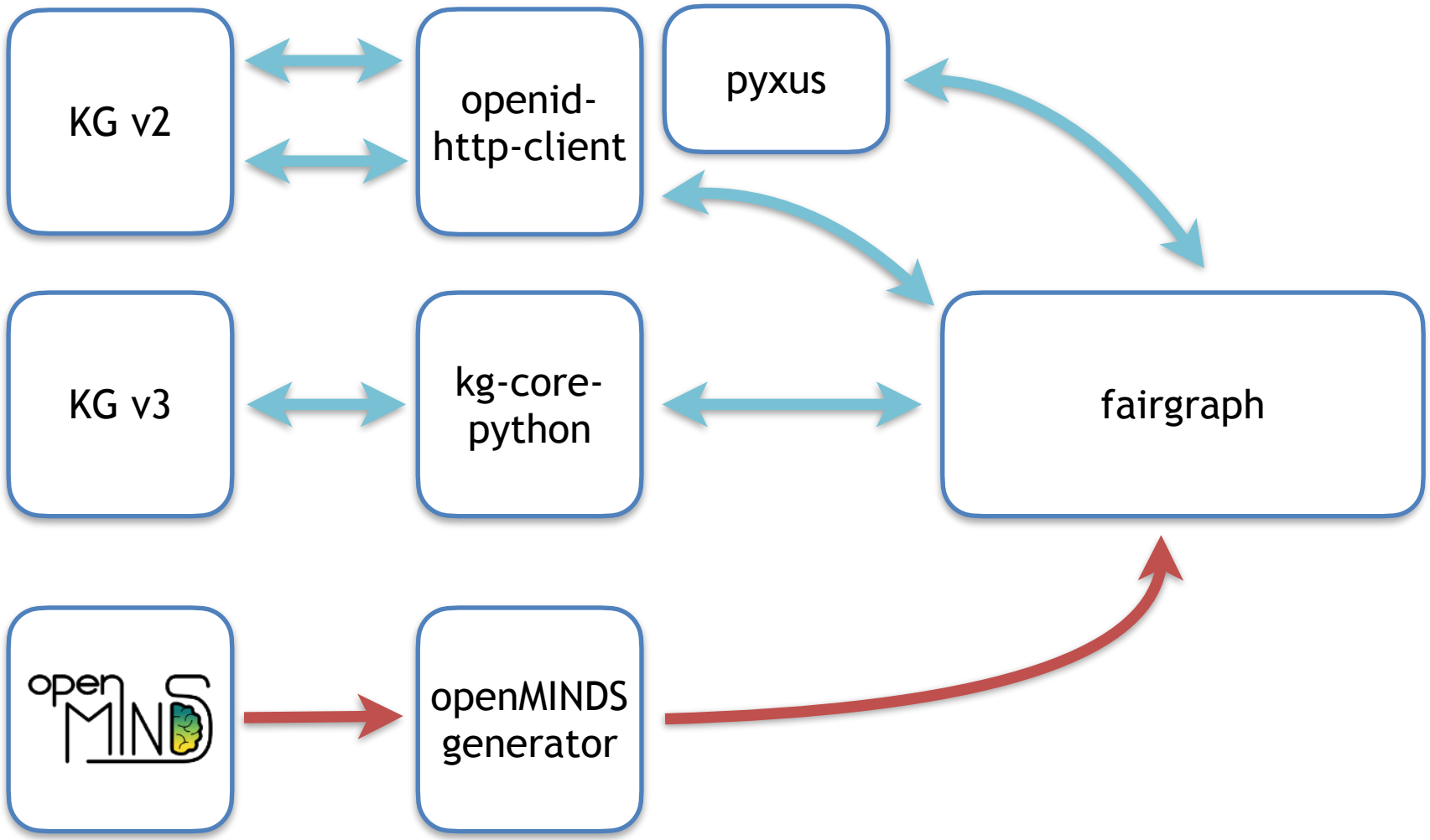
Andrew Davison

CNRS / Université Paris-Saclay

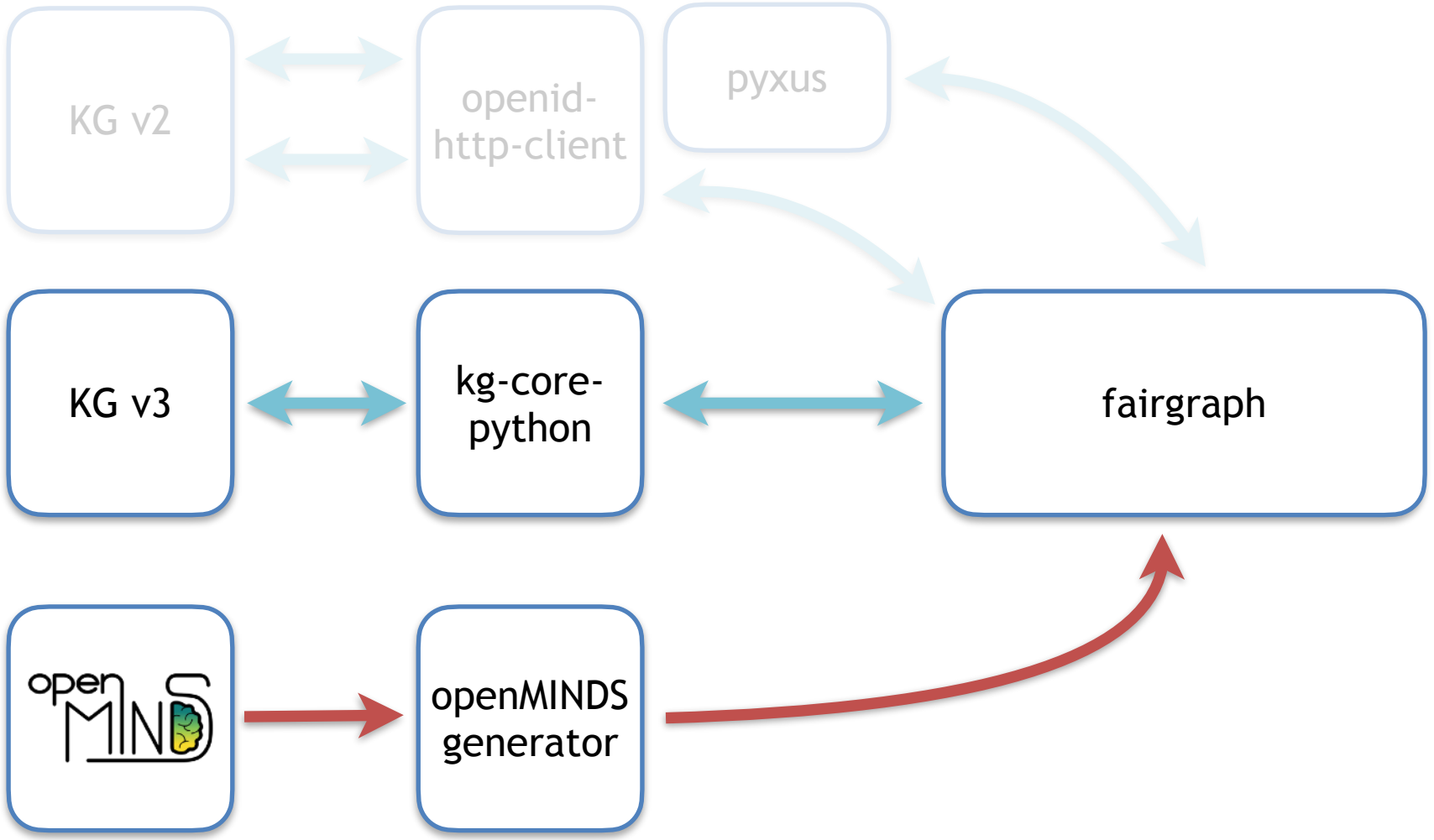
CodeJam #12

23rd November 2021

A Python client for the EBRAINS Knowledge Graph



A Python client for the EBRAINS Knowledge Graph



Getting started

Installation

- `pip install https://github.com/HumanBrainProject/kg-core-python/archive/refs/heads/master.zip`
- `pip install fairgraph`

Example

```
In [1]: from fairgraph.client_v3 import KGv3Client as KGClient
In [2]: from fairgraph.openminds.core import DatasetVersion

In [3]: client = KGClient(host="core.kg-ppd.ebrains.eu")
In [4]: DatasetVersion.set_strict_mode(False)
In [5]: datasets = DatasetVersion.list(client, name="patch-clamp",
   ...:                                scope="in progress")
In [6]: for dataset in datasets:
   ...:     print(dataset.name)
   ...:
Patch-clamp electrophysiological characterization of neurons in human dentate gyrus
Whole cell patch-clamp recordings of cerebellar basket cells
Whole cell patch-clamp recordings of cerebellar Golgi cells
Whole cell patch-clamp recordings of cerebellar granule cells
Whole cell patch-clamp recordings of cerebellar stellate cells
```

Key concepts: scopes

released

- metadata have been curated
- metadata appear in public KG Search (no authentication required)

in progress

- metadata are uncurated, or curation in progress
- metadata are access controlled

Key concepts: spaces

- The KG is divided into *spaces*, which are used for access control and for structure:

  "public" spaces

  - write-access restricted to curators
  - read-access to released metadata

Space	Metadata about…
atlas	spatial locations in the brain
common	people, organisations, funding
computation	workflows
controlled	vocabularies
dataset	datasets
files	files
model	models
restricted	GDPR-sensitive metadata, e.g. e-mail addresses
software	software

Key concepts: spaces

- The KG is divided into *spaces*, which are used for access control and for structure:

  private spaces

  - only for in progress metadata
  - “myspace”
    - read- and write-access for single individual
    - every EBRAINS account holder has their own space
  - “collab” spaces
    - read- and write-access based on collab teams
    - by request from kg@ebrains.eu

openMINDS mapping

Person

https://humanbrainproject.github.io/openMINDS/

https://openminds.ebrains.eu/core/Person

part of **core** in version **v4** - view as **JSON Schema**

Required properties are highlighted in **bold**

PROPERTY		VALUE
NAME*	**DESCRIPTION**	**TYPE**
givenName	Name given to a person, including all potential middle names, but excluding the family name.	string
affiliation	Declaration of a person being closely associated to an organization.	embedded object array (1 - n) of type: Affiliation
contactInformation	Any available way used to contact a person or business (e.g., address, phone number, email address, etc.).	linked object ContactInformation
digitalIdentifier	Digital handle to identify objects or legal persons.	linked object array (1 - n) of type: ORCID
familyName	Name borne in common by members of a family.	string

```
class Person(KGObjectV3):
    """
    Structured information on a person.
    """
    default_space = "common"
    type = ["https://openminds.ebrains.eu/core/Person"]
    context = {
        "vocab": "https://openminds.ebrains.eu/vocab/"
    }
    fields = [
        Field("digital_identifiers",
              "openminds.core.ORCID",
              "vocab:digitalIdentifier",
              multiple=True, required=False,
              doc="Digital handle to identify objects or legal persons."),
        Field("contact_information",
              "openminds.core.ContactInformation",
              "vocab:contactInformation",
              multiple=False, required=False,
              doc="Any available way used to contact a person or business (e
        Field("family_name", str, "vocab:familyName",
              multiple=False, required=False,
              doc="Name borne in common by members of a family."),
        Field("given_name", str, "vocab:givenName",
              multiple=False, required=True,
              doc="Name given to a person, including all potential middle na
        Field("affiliations",
              "openminds.core.Affiliation",
              "vocab:affiliation",
              multiple=True, required=False,
              doc="Declaration of a person being closely associated to an or
    ]
    existence_query_fields = ('given_name', 'family_name')
```

Following links in the graph

```
>>> from fairgraph.openminds.core import SoftwareVersion

>>> SoftwareVersion.set_strict_mode(False)

>>> nest220 = SoftwareVersion.list(client, alias="NEST",
...                                version_identifier="v2.20.0",
...                                scope="in progress")[0]
```

Following links in the graph

```
>>> nest220.show()
id                    https://kg.ebrains.eu/api/instances/
                      fbfc57ca-0cdf-4085-9598-0828342c5
digital_identifier    None
features              [KGProxyV3([<class 'fairgraph.openminds.controlledterms.
                       software_feature.SoftwareFeature'>], 'https://kg.ebrains.eu/api/
                       instances/f77da1d8-c072-4dfd-9ce0-0988134b7dd1'), …]
requirements          ['Python 3', 'GNU Libtool', 'GNU Scientific Library',
                       'for parallel simulation: MPI, OpenMP']
licenses              KGProxyV3([<class 'fairgraph.openminds.core.data.license.
                       License'>],
                       'https://kg.ebrains.eu/api/instances/21e41848-a595-49ed-848b-
                        e810e9c1a8fb')
operating_systems     [KGProxyV3([<class 'fairgraph.openminds.controlledterms.
                       operating_system.OperatingSystem'>], 'https://kg.ebrains.eu/api/
                       instances/a77d9acb-979d-46f2-961e-2a9d5c28521f')…]
```

Following links in the graph

```
>>> nest220.licenses
KGProxyV3([<class 'fairgraph.openminds.core.data.license.License'>],
'https://kg.ebrains.eu/api/instances/21e41848-a595-49ed-848b-e810e9c1a8fb')

>>> license = nest220.licenses.resolve(client, scope="in progress")

>>> license.show()
id          https://kg.ebrains.eu/api/instances/21e41848-a595-49ed-848b-
            e810e9c1a8fb
name        GNU General Public License v2.0 or later
legal_code  https://www.gnu.org/licenses/old-licenses/gpl-2.0-
            standalone.html
alias       GPL-2.0-or-later
webpages    https://spdx.org/licenses/GPL-2.0-or-later.html
```

Convenience methods

```
>>> me = Person.me(client)
>>> me.full_name
'Andrew P. Davison'
```

```
>>> dataset.download(client, local_directory)
```

Saving metadata to the KG

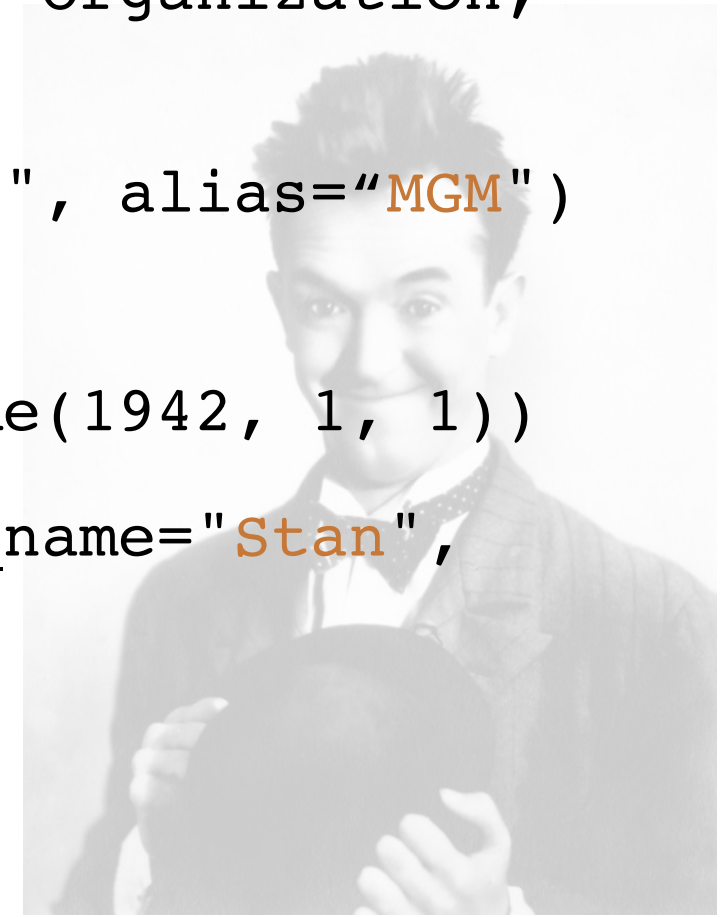
```
from datetime import datetime
from fairgraph.openminds.core import Person, Organization,
Affiliation

mgm = Organization(name="Metro-Goldwyn-Mayer", alias="MGM")

affiliation = Affiliation(organization=mgm,
                          start_date=datetime(1942, 1, 1))

author = Person(family_name="Laurel", given_name="Stan",
                affiliations=affiliation)

author.save(client, space="myspace")
```



Future plans

1. Removal of access to KG v2
2. More “helper” methods and functions
3. Optional automatic following of links
4. Simplified API (explicit “client” not needed)

Bug reports

- fairgraph is beta software
- please address bug reports to https://github.com/HumanBrainProject/fairgraph/issues
- for support requests, visit https://ebrains.eu/support/

Thanks

fairgraph development has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements 800858 (ICEI), 720270 (HBP-SGA1), 785907 (HBP-SGA2) & 945539 (HBP-SGA3).

Glynis Mattheisen
Yann Zerlaut
Onur Ates
Nico Feld

Oliver Schmid & the KG team
Lyuba Zehl
Ulrike Schlegel
Stefan Köhnen