fairgraph Documentation

Release 0.6.0

Andrew P. Davison, Onur Ates, Yann Zerlaut, Nico Feld, Glynis Ma

Nov 23, 2021

Contents:

fairgraph is an experimental Python library for working with metadata in the HBP/EBRAINS Knowledge Graph, with a particular focus on data reuse, although it is also useful in metadata registration/curation. The API is not stable, and is subject to change.

CHAPTER 1

Installation

To get the latest release:

```
pip install fairgraph
```

To get the development version:

```
git clone https://github.com/HumanBrainProject/fairgraph.git
pip install -r ./fairgraph/requirements.txt
pip install -U ./fairgraph
```

CHAPTER 2

About the Knowledge Graph

The Human Brain Project/EBRAINS Knowledge Graph is a metadata store for neuroscience.

When sharing neuroscience data, it is essential to also share all of the context and background information needed to interpret and understand the data: the age of the subject, the sampling rate of the recording system, etc. For the HBP/EBRAINS data sharing platform, the actual data files are stored at the Swiss National Supercomputing Center, CSCS. All of the metadata associated with these files (including the precise file locations) is stored in the Knowledge Graph.

There are many ways to access the contents of the Knowledge Graph: through a graphical search interface, with an anatomical search through the EBRAINS brain atlases, through web services, and through Python clients.

fairgraph is an experimental, high-level Python client for the Knowledge Graph, which aims to be convenient, powerful and easy-to-use. Alternative ways to access the Knowledge Graph programmatically are summarized in the section "Alternatives" below.

2.1 Structure

The HBP/EBRAINS Knowledge Graph is a semantic graph database (in the sense of graph theory). It consists of "nodes", each of which contains metadata about a specific aspect of a neuroscience experiment. These nodes are connected to each other, and the connections represent the relationships between the different pieces of metadata (for example, a node representing a slice of rat hippocampus will be connected to other nodes representing each of the neurons in that slice that was recorded from with an electrode, or reconstructed from microscopy images. The connections between nodes are of many different types, so that we can represent precisely the meaning of the connection, the type of the relationship (this is why we call it a _semantic_ graph). This graph structure gives great flexibility and ease of evolution compared to a traditional database.

Todo: insert a figure here showing a part of the graph

fairgraph maps the Knowledge Graph onto connected Python objects. For example, a node in the graph containing metadata about a neuron whose activity was recorded using patch-clamp electrophysiology is represented by a

Python object `PatchedCell` whose attributes correspond to the metadata stored in that node _and_ to the semantic connections to other nodes.

2.1.1 Alternatives

Todo: write about KG Query API, pyxus, KG Query Python pip

CHAPTER 3

Querying the Knowledge Graph

3.1 Setting up a connection

Communication between fairgraph metadata objects and the Knowledge Graph web service is through a client object, for which an access token associated with an HBP Identity account is needed. To obtain an HBP Identity account, please see https://services.humanbrainproject.eu/oidc/account/request.

If you are working in an HBP Collaboratory Jupyter notebook, you have already logged in with your user name and password, so you can get an access token as follows:

```
token = clb_oauth.get_token()
```

If working outside the Collaboratory, we recommend you obtain a token from whichever authentication endpoint is available to you, and save it as an environment variable so the client can find it, e.g. at a shell prompt:

```
export KG_AUTH_TOKEN=eyJhbGci...nPq
```

You can then create the client object:

```
>>> from fairgraph.client_v3 import KGv3Client as KGClient
>>> client = KGClient()
```

You can also pass the token explicitly to the client:

```
>>> client = KGClient(token)
```

3.2 Listing the available metadata types

Each type of metadata node in the Knowledge Graph is represented by a Python class. These classes are organized into modules according to the openMINDS schemas. For a full list of modules, see *Metadata domains*.

To get a list of classes in a given module, import the module and then run `list_kg_classes()`, e.g.:

```
>>> import fairgraph.openminds.core as omcore

>>> omcore.list_kg_classes()
[fairgraph.openminds.core.research.behavioral_protocol.BehavioralProtocol,
fairgraph.openminds.core.actors.contact_information.ContactInformation,
fairgraph.openminds.core.data.content_type.ContentType,
fairgraph.openminds.core.miscellaneous.doi.DOI,
fairgraph.openminds.core.products.dataset.Dataset,
fairgraph.openminds.core.products.dataset_version.DatasetVersion,
fairgraph.openminds.core.data.file.File,
fairgraph.openminds.core.data.file_bundle.FileBundle,
fairgraph.openminds.core.data.file_repository.FileRepository,
fairgraph.openminds.core.data.file_repository_structure.FileRepositoryStructure,
fairgraph.openminds.core.miscellaneous.funding.Funding,
fairgraph.openminds.core.miscellaneous.gridid.GRIDID,
fairgraph.openminds.core.miscellaneous.isbn.ISBN,
fairgraph.openminds.core.data.license.License,
fairgraph.openminds.core.products.meta_data_model.MetaDataModel,
fairgraph.openminds.core.products.meta_data_model_version.MetaDataModelVersion,
fairgraph.openminds.core.products.model.Model,
fairgraph.openminds.core.products.model_version.ModelVersion,
fairgraph.openminds.core.miscellaneous.orcid.ORCID,
fairgraph.openminds.core.actors.organization.Organization,
fairgraph.openminds.core.actors.person.Person,
fairgraph.openminds.core.products.project.Project,
fairgraph.openminds.core.research.protocol.Protocol,
fairgraph.openminds.core.research.protocol_execution.ProtocolExecution,
fairgraph.openminds.core.miscellaneous.rorid.RORID,
fairgraph.openminds.core.miscellaneous.swhid.SWHID,
fairgraph.openminds.core.data.service_link.ServiceLink,
fairgraph.openminds.core.products.software.Software,
fairgraph.openminds.core.products.software_version.SoftwareVersion,
fairgraph.openminds.core.research.stimulation.Stimulation,
fairgraph.openminds.core.research.subject.Subject,
fairgraph.openminds.core.research.subject_group.SubjectGroup,
fairgraph.openminds.core.research.subject_group_state.SubjectGroupState,
fairgraph.openminds.core.research.subject_state.SubjectState,
fairgraph.openminds.core.research.tissue_sample.TissueSample,
fairgraph.openminds.core.research.tissue_sample_collection.TissueSampleCollection,
fairgraph.openminds.core.research.tissue_sample_collection_state.
↪TissueSampleCollectionState,
fairgraph.openminds.core.research.tissue_sample_state.TissueSampleState,
fairgraph.openminds.core.miscellaneous.url.URL]
```

3.3 Listing all metadata nodes of a given type

To obtain a list of all the metadata nodes of a given type, import the associated class and use the `list()` method, passing the *client* object you created previously, e.g. to get a list of patched cells:

```
from fairgraph.openminds.core import License

licenses = License.list(client)
```

By default, this gives you the first 100 results. You can change the number of results retrieved and the starting point, e.g.

```
licenses = License.list(client, from_index=15, size=10)
```

This returns 10 nodes starting with the 15th. To see how many nodes there are in total:

```
License.count(client)
```

Note: if you consistently retrieve an empty list, it is probably because you do not yet have the necessary permissions. See *Access permissions* for more information.

3.4 Filtering/searching

To obtain only metadata nodes that have certain properties, you can filter the list of nodes. For example, to see only datasets whose name contain the phrase 'patch-clamp':

```
from fairgraph.openminds.core import DatasetVersion

datasets = DatasetVersion.list(client, name="patch-clamp")
```

Warning: the filtering system is currently primitive, and unaware of hierarchies, e.g. filtering by "hippocampus" **will not** return cells with the brain region set to "hippocampus CA1". This is on our list of things to fix soon! To see a list of possible search terms, use the `fields()` attribute, e.g. `DatasetVersion.fields`.

3.5 Retrieving a specific node based on its name or id

If you know the name or unique id of a node in the KnowledgeGraph, you can retrieve it directly:

```
dataset_of_interest = DatasetVersion.by_name('Whole cell patch-clamp recordings of␣
↪cerebellar Golgi cells', client)
dataset_of_interest = DatasetVersion.from_id('17196b79-04db-4ea4-bb69-d20aab6f1d62',␣
↪client)
```

3.6 Viewing metadata and connections

Once you have retrieved a node of interest, the associated metadata are available as attributes of the Python object, e.g.:

```
>>> dataset_of_interest.id
'https://kg.ebrains.eu/api/instances/17196b79-04db-4ea4-bb69-d20aab6f1d62'

>>> dataset_of_interest.uuid
'17196b79-04db-4ea4-bb69-d20aab6f1d62'

>>> dataset_of_interest.description[:100] + "..."
'The Golgi cells, together with granule cells and mossy fibers, form a neuronal␣
↪microcircuit regulati...'
```

Connections between graph nodes are also available as attributes:

```
>>> dataset_of_interest.license
KGProxyV3([<class 'fairgraph.openminds.core.data.license.License'>], 'https://kg.
↪ebrains.eu/api/instances/6ebce971-7f99-4fbc-9621-eeae47a70d85')
```

By default, for performance reasons, connections are not followed, and instead you will see either a `KGQuery` or `KGProxy` object. In both these cases, follow the connection using the `resolve()` method, e.g.:

```
>>> license = dataset_of_interest.license.resolve(client)

>>> license.name
'Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International'
```

Note: It is rather cumbersome to have to follow all these connections manually. In the near future, you will be able to ask fairgraph to resolve the connections for you, although with the risk of poor performance if your node of interest is indirectly connected to many other nodes in the graph.

3.7 Strict mode

fairgraph is quite strict about which metadata attributes and data types are expected, somewhat stricter than the Knowledge Graph itself. If you find that certain queries produce errors, you can relax this strict checking for a given node type as follows:

```
DatasetVersion.set_strict_mode(False)
```

CHAPTER 4

Creating and updating metadata nodes

To create a new metadata node, create an instance of the appropriate Python class, then use the `save()` method, e.g.:

```
from fairgraph.openminds.core import SoftwareVersion, URL

# By default, fairgraph strictly enforces required fields.
# For demonstration purposes we turn this enforcement off here.
SoftwareVersion.set_strict_mode(False)

sv = SoftwareVersion(
    name="numpy",
    alias="numpy",
    version_identifier="1.14.9"
)
sv.save(client, space="myspace")
```

To update a node, edit the attributes of the corresponding Python object, then `save()` again:

```
from fairgraph.base_v3 import IRI

sv.homepage = URL(url=IRI("https://numpy.org"))
sv.save(client)
```

(Note that for updating existing objects you don't need to specify the space.)

4.1 How does fairgraph distinguish between creating a new node and modifying an existing one?

If a previously-created node has been retrieved from the Knowledge Graph, it will have a unique ID, and therefore calling `save()` will update the node with this ID.

If a new Python object is created with the same or similar metadata, **fairgraph** queries for a node with matching metadata for a *subset* of the fields. If you want to know which fields are included in the match, examine the

`existence_query_fields` attribute, e.g.:

```
>>> SoftwareVersion.existence_query_fields
('alias', 'version_identifier')
```

4.2 Permissions

If you get an error message when trying to create or update a node, it may be because you do not have the necessary permissions. See *Access permissions* for more information.

CHAPTER 5

Metadata domains

5.1 KG version 3

5.1.1 openminds.core

fairgraph.openminds.core.**list_kg_classes**()
List all KG classes defined in this module

5.1.2 openminds.controlledterms

fairgraph.openminds.controlledterms.**list_kg_classes**()
List all KG classes defined in this module

5.1.3 openminds.sands

fairgraph.openminds.sands.**list_kg_classes**()
List all KG classes defined in this module

5.1.4 openminds.computation

fairgraph.openminds.computation.**list_kg_classes**()
List all KG classes defined in this module

5.2 KG version 2

5.2.1 minds

"Minimal Information for Neuroscience DataSets" - metadata common to all neuroscience datasets independent of the type of investigation

class fairgraph.minds.**Person**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

A person associated with research data or models, for example as an experimentalist, or a data analyst.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –
- **shortname** (*str*) –

class fairgraph.minds.**Activity**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

A research activity.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –
- **ethics_approval** (EthicsApproval) –
- **ethics_authority** (EthicsAuthority) –
- **methods** (Method) –
- **preparation** (Preparation) –
- **protocols** (Protocol) –

class fairgraph.minds.**AgeCategory**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

An age category, e.g. "adult", "juvenile"

Parameters

- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.minds.**EthicsApproval**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

Record of an ethics approval.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –
- **generated_by** (EthicsAuthority) –

class fairgraph.minds.**EthicsAuthority**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

A entity legally authorised to approve or deny permission to conduct an experiment on ethical grounds.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.minds.**Dataset**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

A collection of related data files.

Parameters

- **activity** (Activity) –
- **container_url_as_ZIP** (*bool*) –
- **container_url** (*str*) –
- **datalink** (*str*) –
- **dataset_doi** (*str*) –
- **description** (*str*) –
- **external_datalink** (*str*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **release_date** (*datetime*) –
- **component** (PLAComponent) –
- **contributors** (Person) –
- **doireference** (*str*) –
- **embargo_status** (EmbargoStatus) –
- **formats** (Format) –
- **license** (License) –
- **modality** (Modality) –
- **owners** (Person) –
- **parcellation_atlas** (ParcellationAtlas) –
- **parcellation_region** (ParcellationRegion) –
- **part_of** (*str*) –
- **publications** (Publication) –
- **reference_space** (ReferenceSpace) –
- **specimen_group** (SpecimenGroup) –

methods(*client*, *api='query'*, *scope='released'*)
Return a list of experimental methods associated with the dataset

classmethod list(*client*, *size=100*, *from_index=0*, *api='query'*, *scope='released'*, *resolved=False*, ***filters*)
List all objects of this type in the Knowledge Graph

class fairgraph.minds.**EmbargoStatus**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

Information about the embargo period during which a given dataset cannot be publicly shared.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.minds.**File**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

Metadata about a single file.

Parameters

- **absolute_path** (*str*) –
- **byte_size** (*int*) –
- **content_type** (*str*) –
- **hash** (*str*) –
- **identifier** (*str*) –
- **last_modified** (*datetime*) –
- **name** (*str*) –
- **relative_path** (*str*) –

class fairgraph.minds.**FileAssociation**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

A link between a file and a dataset.

Parameters

- **from** (File) –
- **identifier** (*str*) –
- **name** (*str*) –
- **to** (Dataset) –

class fairgraph.minds.**Format**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

A file/data format.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.minds.**License**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

A license governing sharing of a dataset.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.minds.**Method**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

An experimental method.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.minds.**Modality**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

A recording modality.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.minds.**ParcellationAtlas**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

A brain atlas in which the brain of a given species of animal is divided into regions.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.minds.**ParcellationRegion**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

A brain region as defined by a brain atlas.

Parameters

- **alias** (*str*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **url** (*str*) –
- **species** (Species) –

class fairgraph.minds.**PLAComponent**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

A data or software component, as defined in the HBP "project lifecycle" application.

Parameters

- **description** (*str*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **component** (*str*) –

class fairgraph.minds.**Preparation**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

An experimental preparation.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.minds.**Protocol**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

An experimental procotol.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.minds.**Publication**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

A scientific publication.

Parameters

- **cite** (*str*) –
- **doi** (*str*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **authors** (Person) –

class fairgraph.minds.**ReferenceSpace**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

A reference space for a brain atlas.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.minds.**Role**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

The role of a person within an experiment.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.minds.**Sample**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

A sample of neural tissue.

Parameters

- **container_url** (*str*) –

- **identifier** (*str*) –
- **name** (*str*) –
- **weight_post_fixation** (*str*) –
- **weight_pre_fixation** (*str*) –
- **methods** (Method) –
- **parcellation_atlas** (ParcellationAtlas) –
- **parcellation_region** (ParcellationRegion) –
- **reference** (*str*) –

class fairgraph.minds.**Sex**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

The sex of an animal or person from whom/which data were obtained.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.minds.**SoftwareAgent**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

Software that performed a given activity.

Parameters

- **description** (*str*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.minds.**Species**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

The species of an experimental subject, expressed with the binomial nomenclature.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.minds.**SpecimenGroup**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

A group of experimental subjects.

Parameters

- **identifier** (*str*) –
- **name** (*str*) –
- **subjects** (Subject) –

class fairgraph.minds.**Subject**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

The organism that is the subject of an experimental investigation.

Parameters

- **cause_of_death** (*str*) –
- **genotype** (*str*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **strain** (*str*) –
- **strains** (*str*) –
- **weight** (*str*) –
- **age** (*str*) –
- **age_category** (AgeCategory) –
- **samples** (Sample) –
- **sex** (Sex) –
- **species** (Species) –

fairgraph.minds.**list_kg_classes**()
List all KG classes defined in this module

fairgraph.minds.**Project**
alias of *fairgraph.minds.PLAComponent*

5.2.2 uniminds

An updated version of MINDS

class fairgraph.uniminds.**UnimindsObject**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

class fairgraph.uniminds.**UnimindsOption**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.minds.MINDSObject

class fairgraph.uniminds.**Person**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsObject*

A person associated with research data or models, for example as an experimentalist, or a data analyst.

Parameters

- **alternatives** (*KGObject*) –
- **email** (*str*) –
- **family_name** (*str*) –
- **given_name** (*str*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **orcid** (*str*) –

class fairgraph.uniminds.**AbstractionLevel**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

Level of abstraction for a neuroscience model, e.g.rate neurons, spiking neurons

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**AgeCategory**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

An age category, e.g. "adult", "juvenile"

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**BrainStructure**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

A sub-structure or region with the brain.

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**CellularTarget**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

The type of neuron or glial cell that is the focus of the study.

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**Country**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

A geographical country.

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**Dataset**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsObject*

A collection of related data files.

Parameters

- **description** (*str*) –
- **identifier** (*str*) –

- **intended_release_date** (*datetime*) –
- **name** (*str*) –
- **brain_structure** (BrainStructure) –
- **cellular_target** (CellularTarget) –
- **contributor** (Person) –
- **custodian** (Person) –
- **doi** (Doi) –
- **embargo_status** (EmbargoStatus) –
- **ethics_approval** (EthicsApproval) –
- **funding_information** (FundingInformation) –
- **hbp_component** (HBPComponent) –
- **license** (License) –
- **main_contact** (Person) –
- **main_file_bundle** (FileBundle) –
- **method** (Method) –
- **project** (Project) –
- **publication** (Publication) –
- **species** (Species) –
- **study_target** (StudyTarget) –
- **subjectgroup** (SubjectGroup) –

class fairgraph.uniminds.**Disability**(*id=None*, *instance=None*, ***properties*)

Bases: *fairgraph.uniminds.UnimindsOption*

A disability or disease.

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**Doi**(*id=None*, *instance=None*, ***properties*)

Bases: *fairgraph.uniminds.UnimindsOption*

Digital Object Identifier (https://www.doi.org)

Parameters

- **citation** (*str*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**EmbargoStatus**(*id=None*, *instance=None*, ***properties*)

Bases: *fairgraph.uniminds.UnimindsOption*

Information about the embargo period during which a given dataset cannot be publicly shared.

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**EthicsApproval**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsObject*

Record of an ethics approval.

Parameters

- **alternatives** (*KGObject*) –
- **hbpethicsapproval** (*str*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **country_of_origin** (Country) –
- **ethics_authority** (EthicsAuthority) –

class fairgraph.uniminds.**EthicsAuthority**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

A entity legally authorised to approve or deny permission to conduct an experiment on ethical grounds.

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**ExperimentalPreparation**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

An experimental preparation.

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**File**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsObject*

Metadata about a single file.

Parameters

- **alternatives** (*KGObject*) –
- **description** (*str*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **url** (*str*) –

- **mime_type** (MimeType) –

class fairgraph.uniminds.**FileAssociation**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsObject*

A link between a file and a dataset.

Parameters

- **from** (File) –
- **identifier** (*str*) –
- **name** (*str*) –
- **to** (Dataset) –

class fairgraph.uniminds.**FileBundle**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsObject*

A collection of files (e.g. in a folder or directory structure)

Parameters

- **alternatives** (*KGObject*) –
- **description** (*str*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **url** (*str*) –
- **usage_notes** (*str*) –
- **file** (File) –
- **file_bundle** (FileBundle) –
- **mime_type** (MimeType) –
- **model_instance** (ModelInstance) –

class fairgraph.uniminds.**FileBundleGroup**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsObject*

A collection of file bundles (see *FileBundle*)

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**FundingInformation**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsObject*

Information about the source of funding of a study.

Parameters

- **alternatives** (*KGObject*) –
- **grant_id** (*str*) –
- **identifier** (*str*) –

- **name** (*str*) –

class fairgraph.uniminds.**Genotype**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

Genetic makeup of a study subject, typically a reference to an inbred strain, with or without mutations.

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**Handedness**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

Preferred hand (left, right, or ambidextrous)

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**HBPComponent**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsObject*

A data or software component, as defined in the HBP "project lifecycle" application.

Parameters

- **alternatives** (*KGObject*) –
- **associated_task** (*str*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **component_owner** (Person) –

class fairgraph.uniminds.**License**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

A license governing sharing of a dataset.

Parameters

- **alternatives** (*KGObject*) –
- **fullname** (*str*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **url** (*str*) –

class fairgraph.uniminds.**Method**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsObject*

An experimental method.

Parameters

- **alternatives** (*KGObject*) –
- **description** (*str*) –
- **fullname** (*str*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **brain_structure** (BrainStructure) –
- **ethics_approval** (EthicsApproval) –
- **experimental_preparation** (ExperimentalPreparation) –
- **method_category** (MethodCategory) –
- **publication** (Publication) –
- **study_target** (StudyTarget) –
- **submethod** (Method) –

class fairgraph.uniminds.**MethodCategory**(*id=None*, *instance=None*, ***properties*)

Bases: *fairgraph.uniminds.UnimindsOption*

A category used for classifying experimental methods (see ExperimentalMethod)

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**MimeType**(*id=None*, *instance=None*, ***properties*)

Bases: *fairgraph.uniminds.UnimindsOption*

Media type of a document

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**ModelFormat**(*id=None*, *instance=None*, ***properties*)

Bases: *fairgraph.uniminds.UnimindsOption*

Programming or markup language used to describe or create a model

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**ModelInstance**(*id=None*, *instance=None*, ***properties*)

Bases: *fairgraph.uniminds.UnimindsObject*

A specific version/parameterization of a neuroscience model.

Parameters

- **alternatives** (*KGObject*) –
- **description** (*str*) –
- **identifier** (*str*) –
- **license** (License) –
- **name** (*str*) –
- **version** (*str*) –
- **abstraction_level** (AbstractionLevel) –
- **brain_structure** (BrainStructure) –
- **cellular_target** (CellularTarget) –
- **contributor** (Person) –
- **custodian** (Person) –
- **main_contact** (Person) –
- **used_dataset** (*KGObject*) –
- **produced_dataset** (Dataset) –
- **modelformat** (ModelFormat) –
- **modelscope** (ModelScope) –
- **publication** (Publication) –
- **study_target** (StudyTarget) –
- **embargo_status** (EmbargoStatus) –
- **alternate_of** (ModelInstance, MEModel) –

class fairgraph.uniminds.**ModelScope**(*id=None*, *instance=None*, ***properties*)

Bases: *fairgraph.uniminds.UnimindsOption*

'What is being modelled': a protein, a single cell, the entire brain, etc.

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**Organization**(*id=None*, *instance=None*, ***properties*)

Bases: *fairgraph.uniminds.UnimindsObject*

An organization associated with research data or models, e.g. a university, lab or department.

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **created_as** (*str*) –

class fairgraph.uniminds.**Project**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsObject*

A research project, which may have generated one or more datasets (see *Dataset*)

Parameters

- **alternatives** (*KGObject*) –
- **description** (*str*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **coordinator** (Person) –

class fairgraph.uniminds.**Publication**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsObject*

A scientific publication.

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **url** (*str*) –
- **brain_structure** (BrainStructure) –
- **project** (Project) –
- **publication_id** (PublicationId) –
- **study_target** (StudyTarget) –
- **subjectgroup** (SubjectGroup) –

class fairgraph.uniminds.**PublicationId**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

Identifier for a publication (e.g. a DOI, a PubMed ID)

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **publication** (Publication) –
- **publication_id_type** (PublicationIdType) –

class fairgraph.uniminds.**PublicationIdType**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

A type of publication identifier (e.g. ISBN, DOI)

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –

- **name** (*str*) –

class fairgraph.uniminds.**Sex**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

The sex of an animal or person from whom/which data were obtained.

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**Species**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

The species of an experimental subject, expressed with the binomial nomenclature.

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**Strain**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

An inbred sub-population within a species.

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**StudyTarget**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsObject*

The focus of an experimental or modelling study.

Parameters

- **alternatives** (*KGObject*) –
- **fullname** (*str*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **study_target_source** (StudyTargetSource) –
- **study_target_type** (StudyTargetType) –

class fairgraph.uniminds.**StudyTargetSource**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

Context of a study target, e.g. if the target is a brain region, the source might be an atlas.

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –

- **name** (*str*) –

class fairgraph.uniminds.**StudyTargetType**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsOption*

Category of study target (see *StudyTarget*)

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –

class fairgraph.uniminds.**Subject**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsObject*

The organism that is the subject of an experimental investigation.

Parameters

- **age** (*str, float*) –
- **age_range_max** (*str, float*) –
- **age_range_min** (*str, float*) –
- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **age_category** (AgeCategory) –
- **brain_structure** (BrainStructure) –
- **cellular_target** (CellularTarget) –
- **disability** (Disability) –
- **genotype** (Genotype) –
- **handedness** (Handedness) –
- **publication** (Publication) –
- **sex** (Sex) –
- **species** (Species) –
- **strain** (Strain) –
- **study_target** (StudyTarget) –

class fairgraph.uniminds.**SubjectGroup**(*id=None*, *instance=None*, ***properties*)
Bases: *fairgraph.uniminds.UnimindsObject*

A group of experimental subjects.

Parameters

- **age_range_max** (*str, float*) –
- **age_range_min** (*str, float*) –
- **alternatives** (*KGObject*) –
- **description** (*str*) –

- **identifier** (*str*) –
- **name** (*str*) –
- **num_of_subjects** (*int*) –
- **age_category** (AgeCategory) –
- **cellular_target** (CellularTarget) –
- **brain_structure** (BrainStructure) –
- **disability** (Disability) –
- **genotype** (Genotype) –
- **handedness** (Handedness) –
- **publication** (Publication) –
- **sex** (Sex) –
- **species** (Species) –
- **strain** (Strain) –
- **study_target** (StudyTarget) –
- **subjects** (Subject) –

class fairgraph.uniminds.**TissueSample**(*id=None*, *instance=None*, ***properties*)

Bases: *fairgraph.uniminds.UnimindsObject*

A sample of brain tissue.

Parameters

- **alternatives** (*KGObject*) –
- **identifier** (*str*) –
- **name** (*str*) –
- **subject** (Subject) –

fairgraph.uniminds.**list_kg_classes**()

List all KG classes defined in this module

5.2.3 electrophysiology

Metadata for electrophysiology experiments.

The following methods are currently supported:

- patch clamp recording in brain slices
- sharp electrode intracellular recording in brain slices

Coming soon:

- patch clamp recordings in cultured neurons
- extracellular electrode recording, including tetrodes and multi-electrode arrays

class fairgraph.electrophysiology.**Sensor**(*name*, *coordinate_system=None*, *coordinate_units=None*, *description=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

Object specific to sensors used in electrode array experiments

Parameters

- **name** (*str*) –
- **coordinate_system** (*Distribution*) –
- **coordinate_units** (*str*) –
- **description** (*str*) –

class fairgraph.electrophysiology.**Trace**(*name*, *data_location*, *generated_by*, *generation_metadata*, *channel*, *data_unit*, *time_step*, *part_of=None*, *retrieval_date=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

Single time series recorded during an experiment or simulation.

Trace represents a single recording from a single channel. If you have a file containing recordings from multiple channels, or multiple recordings from a single channel, use *MultiChannelMultiTrialRecording*.

Parameters

- **name** (*str*) –
- **data_location** (*Distribution*) –
- **generated_by** (PatchClampExperiment) –
- **generation_metadata** (QualifiedTraceGeneration) –
- **channel** (*int*) –
- **data_unit** (*str*) –
- **time_step** (QuantitativeValue) –
- **part_of** (Dataset) –
- **retrieval_date** (*datetime*) –

class fairgraph.electrophysiology.**MultiChannelMultiTrialRecording**(*name*, *data_location*, *generated_by*, *generation_metadata*, *channel_names*, *data_unit*, *time_step*, *channel_type=None*, *part_of=None*, *id=None*, *instance=None*)

Bases: *fairgraph.electrophysiology.Trace*

Multiple time series recorded during an experiment or simulation. Time series may be recorded from multiple channels. If you have a file containing only a single recording from a single channel, you may instead use *Trace*.

Parameters

- **name** (*str*) –
- **data_location** (*Distribution*) –
- **generated_by** (PatchClampExperiment, ExtracellularElectrodeExperiment, ElectrodeArrayExperiment, EEGExperiment, ECoGExperiment) –
- **generation_metadata** (QualifiedMultiTraceGeneration) –
- **channel_names** (*str*) –
- **data_unit** (*str*) –
- **time_step** (QuantitativeValue) –
- **channel_type** (ChannelType) –
- **part_of** (Dataset) –

class fairgraph.electrophysiology.**PatchedCell**(*name*, *brain_location*, *collection=None*, *putative_cell_type=None*, *cell_type=None*, *morphology_type=None*, *experiments=None*, *pipette_id=None*, *seal_resistance=None*, *pipette_resistance=None*, *start_membrane_potential=None*, *end_membrane_potential=None*, *pipette_solution=None*, *liquid_junction_potential=None*, *labeling_compound=None*, *reversal_potential_cl=None*, *description=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

A cell recorded in patch clamp.

Parameters

- **name** (*str*) –
- **brain_location** (BrainRegion) –
- **collection** (PatchedCellCollection) –
- **cell_type** (CellType) –
- **experiments** (PatchClampExperiment) –
- **pipette_id** (*str, int*) –
- **seal_resistance** (QuantitativeValue) –
- **pipette_resistance** (QuantitativeValue) –
- **liquid_junction_potential** (QuantitativeValue) –
- **start_membrane_potential** (QuantitativeValue) –
- **end_membrane_potential** (QuantitativeValue) –
- **pipette_solution** (*str*) –
- **labeling_compound** (*str*) –
- **reversal_potential_cl** (QuantitativeValue) –
- **description** (*str*) –

classmethod list(*client*, *size=100*, *from_index=0*, *api='query'*, *scope='released'*, *resolved=False*, ***filters*)

List all objects of this type in the Knowledge Graph

class fairgraph.electrophysiology.**PatchedSlice**(*name*, *slice*, *recorded_cells*, *recording_activity=None*, *brain_location=None*, *bath_solution=None*, *description=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

A slice that has been recorded from using patch clamp.

Parameters

- **name** (*str*) –
- **slice** (*Slice*) –
- **recorded_cells** (PatchedCellCollection) –
- **recording_activity** (PatchClampActivity) –
- **brain_location** (BrainRegion) –
- **bath_solution** (QuantitativeValue) –
- **description** (*str*) –

class fairgraph.electrophysiology.**PatchedCellCollection**(*name*, *cells*, *slice=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

A collection of patched cells.

Parameters

- **name** (*str*) –

- **cells** (PatchedCell) –
- **slice** (PatchedSlice) –

class fairgraph.electrophysiology.**CellCulture**(*name*, *subject*, *cells*, *culturing_activity=None*, *experiment=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

A cell culture.

Parameters

- **name** (*str*) –
- **subject** (Subject) –
- **cells** (PatchedCell) –
- **culturing_activity** (CellCulturingActivity) –
- **experiment** (PatchClampActivity) –

class fairgraph.electrophysiology.**PatchClampActivity**(*name*, *recorded_tissue*, *recorded_slice=None*, *protocol=None*, *people=None*, *start_time=None*, *end_time=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

A patch clamp recording session.

Parameters

- **name** (*str*) –
- **recorded_tissue** (CellCulture, *Slice*, *CranialWindow*) –
- **recorded_slice** (PatchedSlice) –
- **protocol** (*str*) –
- **people** (Person) –
- **start_time** (*datetime*) –
- **end_time** (*datetime*) –

class fairgraph.electrophysiology.**PatchClampExperiment**(*name*, *recorded_cell*, *acquisition_device=None*, *stimulation=None*, *traces=None*, *start_time=None*, *end_time=None*, *people=None*, *protocol=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

Stimulation of the neural tissue and recording of the responses during a patch clamp recording session.

Parameters

- **name** (*str*) –
- **recorded_cell** (PatchedCell) –
- **acquisition_device** (*Device*) –

- **stimulation** (*VisualStimulation, BehavioralStimulation, ElectrophysiologicalStimulation*) –
- **traces** (Trace, MultiChannelMultiTrialRecording) –
- **start_time** (*datetime*) –
- **end_time** (*datetime*) –
- **people** (Person) –
- **protocol** (Protocol) –

classmethod from_kg_instance(*instance*, *client*, *resolved=False*)
docstring

classmethod list(*client*, *size=100*, *from_index=0*, *api='query'*, *scope='released'*, *resolved=False*, ***filters*)
List all objects of this type in the Knowledge Graph

class fairgraph.electrophysiology.**QualifiedTraceGeneration**(*name*, *stimulus_experiment*, *sweep*, *repetition=None*, *at_time=None*, *provider_experiment_id=None*, *provider_experiment_name=None*, *holding_potential=None*, *measured_holding_potential=None*, *input_resistance=None*, *series_resistance=None*, *compensation_current=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

Additional information about the generation of a single-channel electrophysiology trace.

Parameters

- **name** (*str*) –
- **stimulus_experiment** (PatchClampExperiment, IntraCellularSharpElectrodeExperiment) –
- **sweep** (*int*) –
- **repetition** (*int*) –
- **at_time** (*datetime*) –
- **provider_experiment_id** (*str*) –
- **provider_experiment_name** (*str*) –
- **holding_potential** (QuantitativeValue) –
- **measured_holding_potential** (QuantitativeValue) –

- **input_resistance** (QuantitativeValue) –
- **series_resistance** (QuantitativeValue) –
- **compensation_current** (QuantitativeValue) –

class fairgraph.electrophysiology.**ImplantedBrainTissue**(*name*, *subject*, *implantation_activity=None*, *experiment=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

Brain tissue in which extracellular electrode was implanted.

Parameters

- **name** (*str*) –
- **subject** (Subject) –
- **implantation_activity** (ElectrodeImplantationActivity) –
- **experiment** (ExtracellularElectrodeExperiment) –

resolve(*client*, *api='query'*, *use_cache=True*)

To avoid having to check if a child attribute is a proxy or a real object, a real object resolves to itself.

class fairgraph.electrophysiology.**ElectrodeArrayExperiment**(*name*, *device=None*, *implanted_brain_tissues=None*, *stimulation=None*, *sensors=None*, *digitized_head_points_coordinates=None*, *head_localization_coils_coordinates=None*, *digitized_head_points=False*, *digitized_landmarks=False*, *start_time=None*, *end_time=None*, *people=None*, *protocol=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

Electrode array experiment (EEG, ECoG, MEG, ERP).

Parameters

- **name** (*str*) –
- **device** (*Device*) –
- **implanted_brain_tissues** (ImplantedBrainTissue) –
- **stimulation** (*VisualStimulation, BehavioralStimulation, ElectrophysiologicalStimulation*) –
- **sensors** (Sensor) –
- **digitized_head_points_coordinates** (Sensor) –
- **head_localization_coils_coordinates** (Sensor) –
- **digitized_head_points** (*bool*) –

- **digitized_landmarks** (*bool*) –
- **start_time** (*datetime*) –
- **end_time** (*datetime*) –
- **people** (Person) –
- **protocol** (Protocol) –

classmethod list(*client*, *size=100*, *from_index=0*, *api='query'*, *scope='released'*, *resolved=False*, ***filters*)

List all objects of this type in the Knowledge Graph

class fairgraph.electrophysiology.**ECoGExperiment**(*name*, *device=None*, *implanted_brain_tissues=None*, *stimulation=None*, *sensors=None*, *digitized_head_points_coordinates=None*, *head_localization_coils_coordinates=None*, *digitized_head_points=False*, *digitized_landmarks=False*, *start_time=None*, *end_time=None*, *people=None*, *protocol=None*, *id=None*, *instance=None*)

Bases: *fairgraph.electrophysiology.ElectrodeArrayExperiment*

Electrocorticography experiment.

Parameters

- **name** (*str*) –
- **device** (*Device*) –
- **implanted_brain_tissues** (ImplantedBrainTissue) –
- **stimulation** (*VisualStimulation, BehavioralStimulation, ElectrophysiologicalStimulation*) –
- **sensors** (Sensor) –
- **digitized_head_points_coordinates** (Sensor) –
- **head_localization_coils_coordinates** (Sensor) –
- **digitized_head_points** (*bool*) –
- **digitized_landmarks** (*bool*) –
- **start_time** (*datetime*) –
- **end_time** (*datetime*) –
- **people** (Person) –
- **protocol** (Protocol) –

class fairgraph.electrophysiology.**EEGExperiment**(*name*, *device=None*, *implanted_brain_tissues=None*, *stimulation=None*, *sensors=None*, *digitized_head_points_coordinates=None*, *head_localization_coils_coordinates=None*, *digitized_head_points=False*, *digitized_landmarks=False*, *start_time=None*, *end_time=None*, *people=None*, *protocol=None*, *id=None*, *instance=None*)

Bases: *fairgraph.electrophysiology.ElectrodeArrayExperiment*

Electroencephalography experiment.

Parameters

- **name** (*str*) –
- **device** (*Device*) –
- **implanted_brain_tissues** (ImplantedBrainTissue) –
- **stimulation** (*VisualStimulation, BehavioralStimulation, ElectrophysiologicalStimulation*) –
- **sensors** (Sensor) –
- **digitized_head_points_coordinates** (Sensor) –
- **head_localization_coils_coordinates** (Sensor) –
- **digitized_head_points** (*bool*) –
- **digitized_landmarks** (*bool*) –
- **start_time** (*datetime*) –
- **end_time** (*datetime*) –
- **people** (Person) –
- **protocol** (Protocol) –

class fairgraph.electrophysiology.**ElectrodePlacementActivity**(*name*, *subject*, *brain_location*, *device=None*, *protocol=None*, *people=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

docstring

Parameters

- **name** (*str*) –
- **subject** (Subject) –
- **brain_location** (BrainRegion) –
- **device** (*Device*) –
- **protocol** (Protocol) –

- **people** (Person) –

class fairgraph.electrophysiology.**ElectrodeImplantationActivity**(*name*, *subject*, *brain_location*, *implanted_brain_tissues=None*, *device=None*, *cranial_window=None*, *protocol=None*, *anesthesia=None*, *start_time=None*, *end_time=None*, *people=None*, *id=None*, *instance=None*)

Bases: *fairgraph.electrophysiology.ElectrodePlacementActivity*

docstring

Parameters

- **name** (*str*) –
- **subject** (Subject) –
- **brain_location** (BrainRegion) –
- **implanted_brain_tissues** (ImplantedBrainTissue) –
- **device** (*Device*) –
- **cranial_window** (*CranialWindow*) –
- **protocol** (Protocol) –
- **anesthesia** (*str*) –
- **start_time** (*datetime*) –
- **end_time** (*datetime*) –
- **people** (Person) –

class fairgraph.electrophysiology.**ExtracellularElectrodeExperiment**(*name*, *stimulation=None*, *recorded_cell=None*, *traces=None*, *id=None*, *instance=None*)

Bases: *fairgraph.electrophysiology.PatchClampExperiment*

Stimulation of the neural tissue and recording of the responses with an extracellular electrode.

Parameters

- **name** (*str*) –

- **stimulation** (*VisualStimulation, BehavioralStimulation, ElectrophysiologicalStimulation*) –
- **recorded_cell** (ImplantedBrainTissue) –
- **traces** (Trace) –

class fairgraph.electrophysiology.**IntraCellularSharpElectrodeRecordedCell**(*name*, *brain_location*, *collection=None*, *putative_cell_type=None*, *cell_type=None*, *morphology_type=None*, *experiments=None*, *pipette_id=None*, *seal_resistance=None*, *pipette_resistance=None*, *start_membrane_potenti*, *end_membrane_potentia*, *pipette_solution=None*, *liquid_junction_potential=*, *labeling_compound=None*, *reversal_potential_cl=None*, *description=None*, *id=None*, *instance=None*)

Bases: *fairgraph.electrophysiology.PatchedCell*

A cell recorded intracellularly with a sharp electrode.

Parameters

- **name** (*str*) –
- **brain_location** (BrainRegion) –
- **collection** (IntraCellularSharpElectrodeRecordedCellCollection) –
- **cell_type** (CellType) –
- **experiments** (IntraCellularSharpElectrodeExperiment) –

- **pipette_id** (*str, int*) –
- **seal_resistance** (QuantitativeValue) –
- **pipette_resistance** (QuantitativeValue) –
- **liquid_junction_potential** (QuantitativeValue) –
- **labeling_compound** (*str*) –
- **reversal_potential_cl** (QuantitativeValue) –

class fairgraph.electrophysiology.**IntraCellularSharpElectrodeRecording**(*name*, *recorded_tissue*, *recorded_slice=None*, *protocol=None*, *people=None*, *start_time=None*, *end_time=None*, *id=None*, *instance=None*)

Bases: *fairgraph.electrophysiology.PatchClampActivity*

A sharp-electrode recording session.

Parameters

- **name** (*str*) –
- **recorded_tissue** (CellCulture, *Slice, CranialWindow*) –
- **recorded_slice** (IntraCellularSharpElectrodeRecordedSlice) –
- **protocol** (*str*) –
- **people** (Person) –

class fairgraph.electrophysiology.**IntraCellularSharpElectrodeRecordedCellCollection**(*name*, *cells*, *slice=No*, *id=None*, *instance=N*

Bases: *fairgraph.electrophysiology.PatchedCellCollection*

A collection of cells recorded with a sharp electrode.

Parameters

- **name** (*str*) –
- **cells** (IntraCellularSharpElectrodeRecordedCell) –
- **slice** (IntraCellularSharpElectrodeRecordedSlice) –

class fairgraph.electrophysiology.**IntraCellularSharpElectrodeRecordedSlice**(*name*, *slice*, *recorded_cells*, *recording_activity=None*, *brain_location=None*, *bath_solution=None*, *description=None*, *id=None*, *instance=None*)

Bases: *fairgraph.electrophysiology.PatchedSlice*

A slice that has been recorded from using a sharp electrode.

Parameters

- **name** (*str*) –
- **slice** (*Slice*) –
- **recorded_cells** (IntraCellularSharpElectrodeRecordedCellCollection) –
- **recording_activity** (IntraCellularSharpElectrodeRecording) –

class fairgraph.electrophysiology.**IntraCellularSharpElectrodeExperiment**(*name*, *recorded_cell*, *acquisition_device=None*, *stimulation=None*, *traces=None*, *start_time=None*, *end_time=None*, *people=None*, *protocol=None*, *id=None*, *instance=None*)

Bases: *fairgraph.electrophysiology.PatchClampExperiment*

Stimulation of the neural tissue and recording of the responses with a sharp intracellular electrode.

Parameters

- **name** (*str*) –
- **recorded_cell** (IntraCellularSharpElectrodeRecordedCell) –

- **stimulation** (*VisualStimulation, BehavioralStimulation, ElectrophysiologicalStimulation*) –
- **traces** (Trace) –
- **people** (Person) –

classmethod list(*client*, *size=100*, *from_index=0*, *api='query'*, *scope='released'*, *resolved=False*, ***filters*)

List all objects of this type in the Knowledge Graph

class fairgraph.electrophysiology.**QualifiedMultiTraceGeneration**(*name*, *stimulus_experiment*, *sweeps*, *channel_type=None*, *holding_potential=None*, *sampling_frequency=None*, *power_line_frequency=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

Parameters

- **name** (*str*) –
- **stimulus_experiment** (ExtracellularElectrodeExperiment, IntraCellularSharpElectrodeExperiment, PatchClampExperiment, ElectrodeArrayExperiment) –
- **sweeps** (*int*) –
- **channel_type** (*str*) –
- **holding_potential** (QuantitativeValue) –
- **sampling_frequency** (QuantitativeValue) –
- **power_line_frequency** (QuantitativeValue) –

class fairgraph.electrophysiology.**CellCulturingActivity**(*subject*, *cell_culture*, *brain_location=None*, *culture_type=None*, *culture_age=None*, *hemisphere=None*, *culture_solution=None*, *start_time=None*, *end_time=None*, *people=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

The activity of preparing a cell culture from whole brain.

Parameters

- **subject** (Subject) –
- **cell_culture** (CellCulture) –
- **brain_location** (BrainRegion) –

- **culture_type** (CultureType) –
- **culture_age** (QuantitativeValueRange) –
- **hemisphere** (*str*) –
- **culture_solution** (*str*) –
- **start_time** (*datetime*) –
- **end_time** (*datetime*) –
- **people** (Person) –

resolve(*client*, *api='query'*, *use_cache=True*)
To avoid having to check if a child attribute is a proxy or a real object, a real object resolves to itself.

fairgraph.electrophysiology.**list_kg_classes**()
List all KG classes defined in this module

fairgraph.electrophysiology.**use_namespace**(*namespace*)
Set the namespace for all classes in this module.

5.2.4 brainsimulation

Metadata for model building, simulation and validation.

class fairgraph.brainsimulation.**ModelProject**(*name*, *owners*, *authors*, *description*, *date_created*, *private*, *collab_id=None*, *alias=None*, *organization=None*, *pla_components=None*, *brain_region=None*, *species=None*, *celltype=None*, *abstraction_level=None*, *model_of=None*, *old_uuid=None*, *parents=None*, *instances=None*, *images=None*, *id=None*, *instance=None*)
Bases: fairgraph.base.KGObject, fairgraph.base.HasAliasMixin

Representation of a neuroscience model or modelling project.

We distinguish a model in an abstract sense (this class), which may have multiple parameterizations and multiple implementations, from a specific version and parameterization of a model - see *ModelInstance* and *ModelScript*

Parameters

- **name** (*str*) –
- **owners** (Person) –
- **authors** (Person) –
- **description** (*str*) –
- **date_created** (*datetime*) –
- **private** (*bool*) –
- **collab_id** (*str*) –
- **alias** (*str*) –
- **organization** (Organization) –

- **pla_components** (*str*) –
- **brain_region** (BrainRegion) –
- **species** (Species) –
- **celltype** (CellType) –
- **abstraction_level** (AbstractionLevel) –
- **model_of** (ModelScope) –
- **old_uuid** (*str*) –
- **parents** (ModelProject) –
- **instances** (ModelInstance, MEModel) –
- **images** (*dict*) –

class fairgraph.brainsimulation.**ModelInstance**(*name*, *main_script*, *version*, *timestamp=None*, *brain_region=None*, *species=None*, *model_of=None*, *release=None*, *part_of=None*, *description=None*, *parameters=None*, *old_uuid=None*, *alternate_of=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

A specific implementation, code version and parameterization of a model.

See also: *ModelProject*, *MEModel*, *ModelScript*

Parameters

- **name** (*str*) –
- **brain_region** (BrainRegion) –
- **species** (Species) –
- **model_of** (CellType, BrainRegion) –
- **main_script** (ModelScript) –
- **release** (*str*) –
- **version** (*str*) –
- **timestamp** (*datetime*) –
- **part_of** (*KGObject*) –
- **description** (*str*) –
- **parameters** (*str*) –
- **old_uuid** (*str*) –
- **alternate_of** (*KGObject*) –

class fairgraph.brainsimulation.**MEModel**(*name*, *e_model*, *morphology*, *main_script*, *version*, *timestamp=None*, *brain_region=None*, *species=None*, *model_of=None*, *release=None*, *part_of=None*, *description=None*, *parameters=None*, *old_uuid=None*, *alternate_of=None*, *id=None*, *instance=None*)

Bases: *fairgraph.brainsimulation.ModelInstance*

A specific implementation, code version and parameterization of a single neuron model with a defined morphology (M) and electrical (E) behaviour.

This is a specialized sub-class of *ModelInstance*.

See also: *ModelProject*, *ModelScript*, *Morphology*, *EModel*

Parameters

- **name** (*str*) –
- **brain_region** (BrainRegion) –
- **species** (Species) –
- **model_of** (CellType, BrainRegion) –
- **main_script** (ModelScript) –
- **release** (*str*) –
- **version** (*str*) –
- **timestamp** (*datetime*) –
- **part_of** (*KGObject*) –
- **description** (*str*) –
- **parameters** (*str*) –
- **old_uuid** (*str*) –
- **alternate_of** (*KGObject*) –
- **morphology** (Morphology) –
- **e_model** (EModel) –

class fairgraph.brainsimulation.**Morphology**(*name*, *cell_type=None*, *morphology_file=None*, *distribution=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

The morphology of a single neuron model, typically defined as a set of cylinders or truncated cones connected in a tree structure.

Parameters

- **name** (*str*) –
- **cell_type** (CellType) –
- **distribution** (*Distribution*) –

class fairgraph.brainsimulation.**ModelScript**(*name*, *code_location=None*, *code_format=None*, *license=None*, *distribution=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

Code or markup defining all or part of a model.

See also: *ModelInstance*, *MEModel*, *EModel*

Parameters

- **name** (*str*) –
- **code_format** (*str*) –
- **license** (*str*) –
- **distribution** (*Distribution*) –

class fairgraph.brainsimulation.**EModel**(*name*, *main_script=None*, *version=None*, *timestamp=None*, *brain_region=None*, *species=None*, *model_of=None*, *release=None*, *part_of=None*, *description=None*, *parameters=None*, *old_uuid=None*, *id=None*, *instance=None*)

Bases: *fairgraph.brainsimulation.ModelInstance*

The electrical component of an *MEModel*

Parameters

- **name** (*str*) –
- **brain_region** (BrainRegion) –
- **species** (Species) –
- **model_of** (CellType, BrainRegion) –
- **main_script** (ModelScript) –
- **release** (*str*) –
- **version** (*str*) –
- **timestamp** (*datetime*) –
- **part_of** (*KGObject*) –
- **description** (*str*) –
- **parameters** (*str*) –
- **old_uuid** (*str*) –

class fairgraph.brainsimulation.**ValidationTestDefinition**(*id=None*, *instance=None*, ***properties*)

Bases: fairgraph.base.KGObject, fairgraph.base.HasAliasMixin

Definition of a model validation test.

See also: *ValidationScript*, *ValidationActivity*, *ValidationResult*

Parameters

- **name** (*str*) –

- **authors** (Person) –
- **description** (*str*) –
- **date_created** (*date, datetime*) –
- **alias** (*str*) –
- **brain_region** (BrainRegion) –
- **species** (Species) –
- **celltype** (CellType) –
- **test_type** (*str*) –
- **age** (Age) –
- **reference_data** (*KGObject*) –
- **data_type** (*str*) –
- **recording_modality** (*str*) –
- **score_type** (*str*) –
- **status** (*str*) –
- **old_uuid** (*str*) –

class fairgraph.brainsimulation.**ValidationScript**(*id=None*, *instance=None*, ***properties*)

Bases: fairgraph.base.KGObject

Code implementing a particular model validation test.

See also: ***ValidationTestDefinition***, ***ValidationActivity***, *ValidationResult*

Parameters

- **name** (*str*) –
- **date_created** (*date, datetime*) –
- **repository** (*IRI*) –
- **version** (*str*) –
- **description** (*str*) –
- **parameters** (*str*) –
- **test_class** (*str*) –
- **test_definition** (ValidationTestDefinition) –
- **old_uuid** (*str*) –

class fairgraph.brainsimulation.**ValidationResult**(*id=None*, *instance=None*, ***properties*)

Bases: fairgraph.base.KGObject

The results of running a model validation test.

Including a numerical score, and optional additional data.

See also: *ValidationTestDefinition*, *ValidationScript*, *ValidationActivity*.

Parameters

- **name** (*str*) –
- **generated_by** (ValidationActivity) –
- **description** (*str*) –
- **score** (*float, int*) –
- **normalized_score** (*float, int*) –
- **passed** (*bool*) –
- **timestamp** (*date, datetime*) –
- **additional_data** (*KGObject*) –
- **old_uuid** (*str*) –
- **collab_id** (*str*) –
- **hash** (*str*) –

class fairgraph.brainsimulation.**ValidationActivity**(*id=None*, *instance=None*, ***properties*)

Bases: fairgraph.base.KGObject

Record of the validation of a model against experimental data.

Links a *ModelInstance*, a *ValidationTestDefinition* and a reference data set to a *ValidationResult*.

Parameters

- **model_instance** (ModelInstance, MEModel) –
- **test_script** (ValidationScript) –
- **reference_data** (Collection) –
- **timestamp** (*datetime*) –
- **result** (ValidationResult) –
- **started_by** (Person) –
- **end_timestamp** (*datetime*) –

class fairgraph.brainsimulation.**Simulation**(*id=None*, *instance=None*, ***properties*)

Bases: fairgraph.base.KGObject

Parameters

- **name** (*str*) –
- **description** (*str*) –
- **identifier** (*str*) –
- **model_instance** (ModelInstance, MEModel) –
- **config** (SimulationConfiguration) –
- **timestamp** (*datetime*) –
- **result** (SimulationOutput) –
- **started_by** (Person) –

- **end_timestamp** (*datetime*) –
- **computing_environment** (*ComputingEnvironment*) –
- **status** (*str*) –
- **resource_usage** (*float*) –
- **tags** (*str*) –
- **job_id** (*str*) –

class fairgraph.brainsimulation.**SimulationConfiguration**(*name*, *config_file=None*, *description=None*, *identifier=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

Parameters

- **name** (*str*) –
- **identifier** (*str*) –
- **description** (*str*) –
- **config_file** (*Distribution, str*) –

save(*client*)

docstring

class fairgraph.brainsimulation.**SimulationOutput**(*name*, *identifier=None*, *result_file=None*, *generated_by=None*, *derived_from=None*, *data_type=None*, *variable=None*, *target=None*, *description=None*, *timestamp=None*, *brain_region=None*, *species=None*, *celltype=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

Parameters

- **name** (*str*) –
- **description** (*str*) –
- **identifier** (*str*) –
- **result_file** (*Distribution, str*) –
- **generated_by** (Simulation) –
- **derived_from** (*KGObject*) –
- **target** (*str*) –
- **data_type** (*str*) –
- **timestamp** (*datetime*) –
- **brain_region** (BrainRegion) –
- **species** (Species) –
- **celltype** (CellType) –

save(*client*)
docstring

fairgraph.brainsimulation.**list_kg_classes**()
List all KG classes defined in this module

fairgraph.brainsimulation.**use_namespace**(*namespace*)
Set the namespace for all classes in this module.

5.2.5 software

Metadata about, or related to, software

class fairgraph.software.**SoftwareCategory**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

class fairgraph.software.**OperatingSystem**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

class fairgraph.software.**ProgrammingLanguage**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

class fairgraph.software.**SoftwareFeatureCategory**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.base.KGObject

Parameters

- **identifier** (*str*) –
- **name** (*str*) –
- **description** (*str*) –
- **parent** (SoftwareFeatureCategory) –

class fairgraph.software.**SoftwareFeature**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.base.KGObject

Parameters

- **name** (*str*) –
- **description** (*str*) –
- **category** (SoftwareFeatureCategory) –
- **identifier** (*str*) –

class fairgraph.software.**Keyword**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.base.KGObject

Parameters

- **name** (*str*) –
- **identifier** (*str*) –

class fairgraph.software.**Software**(*id=None*, *instance=None*, ***properties*)
Bases: fairgraph.base.KGObject

Parameters

- **name** (*str*) –
- **description** (*str*) –

- **citation** (*str*) –
- **release_date** (*date*) –
- **categories** (SoftwareCategory) –
- **license** (License) –
- **operating_system** (OperatingSystem) –
- **release_notes** (*IRI*) –
- **requirements** (*str*) –
- **summary** (*str*) –
- **contributors** (Person) –
- **copyright** (Person, Organization) –
- **homepage** (*IRI*) –
- **documentation** (*IRI*) –
- **help** (*IRI*) –
- **source_code** (*IRI*) –
- **programming_languages** (ProgrammingLanguage) –
- **funding** (Organization) –
- **components** (Software) –
- **is_free** (*bool*) –
- **keywords** (Keyword) –
- **version** (*str*) –
- **features** (SoftwareFeature) –

fairgraph.software.**list_kg_classes**()
List all KG classes defined in this module

fairgraph.software.**use_namespace**(*namespace*)
Set the namespace for all classes in this module.

5.2.6 core

Metadata for entities that are used in multiple contexts (e.g. in both electrophysiology and in simulation).

class fairgraph.core.**Subject**(*name*, *species*, *age=None*, *sex=None*, *handedness=None*, *strain=None*, *genotype=None*, *death_date=None*, *group=None*, *id=None*, *instance=None*)
Bases: fairgraph.base.KGObject

The individual organism that is the subject of an experimental study.

Parameters

- **name** (*str*) –
- **species** (Species) –
- **strain** (Strain) –
- **genotype** (Genotype) –

- **sex** (Sex) –
- **handedness** (Handedness) –
- **age** (Age) –
- **death_date** (*date*) –
- **group** (Group) –

class fairgraph.core.**Organization**(*name*, *address=None*, *parent=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

An organization associated with research data or models, e.g. a university, lab or department.

Parameters

- **name** (*str*) –
- **address** (Address) –
- **parent** (Organization) –

class fairgraph.core.**Person**(*family_name*, *given_name*, *email=None*, *affiliation=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

A person associated with research data or models, for example as an experimentalist, or a data analyst.

Parameters

- **family_name** (*str*) – Family name / surname
- **given_name** (*str*) – Given name
- **email** (*str*) – e-mail address
- **affiliation** (Organization) – Organization to which person belongs

classmethod list(*client*, *size=100*, *api='query'*, *scope='released'*, *resolved=False*, ***filters*)

List all objects of this type in the Knowledge Graph

resolve(*client*, *api='query'*, *use_cache=True*)

To avoid having to check if a child attribute is a proxy or a real object, a real object resolves to itself.

classmethod me(*client*, *api='query'*, *allow_multiple=False*)

Return the Person who is currently logged-in.

(the user associated with the token stored in the client).

If the Person node does not exist in the KG, it will be created.

class fairgraph.core.**Identifier**(*id=None*, *instance=None*, ***properties*)

Bases: fairgraph.base.KGObject

class fairgraph.core.**Material**(*name*, *molar_weight=None*, *formula=None*, *stock_keeping_unit=None*, *reagent_distribution=None*, *vendor=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

Metadata about a chemical product or other material used in an experimental protocol.

Parameters

- **name** (*str*) –
- **molar_weight** (QuantitativeValue) –

- **formula** (*str*) –
- **stock_keeping_unit** (*str*) –
- **reagent_distribution** (*Distribution*) –
- **vendor** (Organization) –

class fairgraph.core.**Step**(*name*, *previous_step_name=None*, *sequence_number=None*, *identifier=None*, *version=None*, *distribution=None*, *description=None*, *materials=None*, *author=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

A step in an experimental protocol.

Parameters

- **name** (*str, int*) –
- **previous_step_name** (*str, int*) –
- **sequence_number** (*int*) –
- **identifier** (*str*) –
- **version** (*str, int*) –
- **distribution** (*Distribution*) –
- **description** (*str*) –
- **materials** (Material) –
- **author** (Person) –

class fairgraph.core.**Protocol**(*name*, *version=None*, *identifier=None*, *doi=None*, *distribution=None*, *number_of_steps=None*, *steps=None*, *materials=None*, *author=None*, *date_published=None*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

An experimental protocol.

Parameters

- **name** (*str*) –
- **version** (*str, int*) –
- **identifier** (*str*) –
- **distribution** (*Distribution*) –
- **number_of_steps** (*int*) –
- **steps** (Step) –
- **materials** (Material) –
- **author** (Person) –
- **date_published** (*date*) –

class fairgraph.core.**Collection**(*name*, *members*, *id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

A collection of other metadata objects

Parameters

- **name** (*str*) –
- **members** (*KGObject*) –

fairgraph.core.**list_kg_classes**()
List all KG classes defined in this module

fairgraph.core.**use_namespace**(*namespace*)
Set the namespace for all classes in this module.

5.2.7 commons

class fairgraph.commons.**Address**(*locality*, *country*)
Bases: fairgraph.base.StructuredMetadata

class fairgraph.commons.**Group**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

The subject group

class fairgraph.commons.**CultureType**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

The type of cell culture used

class fairgraph.commons.**Species**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

The species of an experimental subject, expressed with the binomial nomenclature.

class fairgraph.commons.**Shape**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

Shape of a region of interest (ROI).

class fairgraph.commons.**MorphologyType**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

The morphology of the cell used for recording.

class fairgraph.commons.**SomaType**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

The type of soma of a reconstructed cell.

class fairgraph.commons.**ObjectiveType**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

The type of objective used for microscopy.

class fairgraph.commons.**Strain**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

An inbred sub-population within a species.

class fairgraph.commons.**Genotype**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

Transgenic modification of the strain.

class fairgraph.commons.**Sex**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

The sex of an animal or person from whom/which data were obtained.

class fairgraph.commons.**Handedness**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

The handedness of an animal or person from whom/which data were obtained.

class fairgraph.commons.**ChannelType**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

The recording method used.

class fairgraph.commons.**BrainRegion**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

A sub-structure or region with the brain.

class fairgraph.commons.**CellType**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

A type of neuron or glial cell.

class fairgraph.commons.**AbstractionLevel**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

Level of abstraction for a neuroscience model, e.g.rate neurons, spiking neurons

class fairgraph.commons.**ModelScope**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

docstring

class fairgraph.commons.**License**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

class fairgraph.commons.**StimulusType**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

class fairgraph.commons.**Origin**(*label*, *iri=None*, *strict=False*)
Bases: fairgraph.base.OntologyTerm

class fairgraph.commons.**QuantitativeValue**(*value*, *unit_text*, *unit_code=None*)
Bases: fairgraph.base.StructuredMetadata

docstring

class fairgraph.commons.**QuantitativeValueRange**(*min*, *max*, *unit_text*, *unit_code=None*)
Bases: fairgraph.base.StructuredMetadata

docstring

class fairgraph.commons.**Age**(*value*, *period*)
Bases: fairgraph.base.StructuredMetadata

Parameters

- **value** (*str*) –
- **period** (*str*) –

fairgraph currently provides the following modules for working with KG v3:

minds "Minimal Information for Neuroscience DataSets" - metadata common to all neuroscience datasets independent of the type of investigation

uniminds an updated version of MINDS

electrophysiology metadata relating to patch clamp and sharp electrode intracellular recordings *in vitro*. Support for extracellular recording, tetrodes, multi-electrode arrays and *in vivo* recordings coming soon.

brainsimulation metadata relating to modelling, simulation and validation

software metadata relating to software used in neuroscience (for simulation, data analysis, stimulus presentation, etc.)

core metadata for entities that are used in multiple contexts (e.g. in both electrophysiology and in simulation).

commons metadata that are not specific to EBRAINS, typically these refer to URIs in standard ontologies, outside the Knowledge Graph.

Additional modules are planned, e.g. for fMRI, functional optical imaging. In addition, the base, commons, and utility modules provide additional tools for structuring metadata and for working with fairgraph objects.

CHAPTER 6

Access permissions

Before accessing the Human Brain Project/EBRAINS Knowledge Graph through fairgraph, you must read and accept the Terms of Use, and then e-mail support@ebrains.eu to request access.

CHAPTER 7

Contributing to fairgraph

Todo: add information about creating tickets, sending feedback, and a developers' guide.

CHAPTER 8

Getting help

In case of questions about **fairgraph**, please contact us via https://ebrains.eu/support/. If you find a bug or would like to suggest an enhancement or new feature, please open a ticket in the issue tracker.

CHAPTER 9

Authors / contributors

The following people have contributed to fairgraph. Their affiliations at the time of the contributions are shown below.

- Andrew Davison [1]
- Onur Ates [1]
- Yann Zerlaut [1]
- Nico Feld [2]
- Glynis Mattheisen[1]

1. Department of Integrative and Computational Neuroscience, Paris-Saclay Institute of Neuroscience, CNRS/Université Paris Saclay
2. Human-Computer Interaction, Department IV, Computer Science, Universität Trier

9.1 Acknowledgements

** <img src="**https://www.braincouncil.eu/wp-content/uploads/2018/11/wsi-imageoptim-EU-Logo.jpg**"** alt="EU Logo" height="23%" width="15%" align="right" style="margin-left: 10px">

This open source software code was developed in part or in whole in the Human Brain Project, funded from the European Union's Horizon 2020 Framework Programme for Research and Innovation under Specific Grant Agreements No. 720270, No. 785907 and No. 945539 (Human Brain Project SGA1, SGA2 and SGA3).

CHAPTER 10

Quickstart

10.1 Installation

To get the latest release:

```
pip install fairgraph
```

To get the development version:

```
git clone https://github.com/HumanBrainProject/fairgraph.git
pip install -r ./fairgraph/requirements.txt
pip install -U ./fairgraph
```

10.2 Basic setup

The basic idea of the library is to represent metadata nodes from the Knowledge Graph as Python objects. Communication with the Knowledge Graph service is through a client object, for which an access token associated with an EBRAINS account is needed.

If you are working in a Collaboratory Jupyter notebook, the client will take its access token from the notebook automatically:

```
from fairgraph import KGClient

client = KGClient()
```

If working outside the Collaboratory, you will need to obtain a token (for example from the KG Editor if you are a curator, or using *clb_oauth.get_token()* in a Collaboratory Jupyter notebook) and save it as an environment variable, e.g. at a shell prompt:

```
export KG_AUTH_TOKEN=eyJhbGci...nPq
```

and then in Python:

```
token = os.environ['KG_AUTH_TOKEN']
```

Once you have a token:

```
from fairgraph import KGClient

client = KGClient(token)
```

10.3 Retrieving metadata from the Knowledge Graph

The different metadata/data types available in the Knowledge Graph are grouped into modules within the *openminds* module. For example:

```
from fairgraph.openminds.core import DatasetVersion
```

Using these classes, it is possible to list all metadata matching a particular criterion, e.g.:

```
datasets = DatasetVersion.list(client, from_index=10, size=10)
```

If you know the unique identifier of an object, you can retrieve it directly:

```
dataset_of_interest = Dataset.from_id("153ec151-b1ae-417b-96b5-4ce9950a3c56", client)
dataset_of_interest.show()
```

Links between metadata in the Knowledge Graph are not followed automatically, to avoid unnecessary network traffic, but can be followed with the *resolve()* method:

```
dataset_license = dataset_of_interest.license.resolve(client)
```

The associated metadata are accessible as attributes of the Python objects, e.g.:

```
print(dataset_of_interest.description)
```

You can also access any associated data:

```
print(dataset.files)
dataset.download(dataset.files[0])
```

10.4 Advanced queries

While certain filters and queries are built in (such as the filter by brain region, above), more complex queries are possible using the Nexus query API.

```
from fairgraph.base import KGQuery
from fairgraph.minds import Dataset

query = {
   "path": "minds:specimen_group / minds:subjects / minds:samples / minds:methods /␣
↪schema:name",
   "op": "in",
```

(continues on next page)

(continued from previous page)

```
    "value": ["Electrophysiology recording",
              "Voltage clamp recording",
              "Single electrode recording",
              "functional magnetic resonance imaging"]
}
context = {
              "schema": "http://schema.org/",
              "minds": "https://schema.hbp.eu/minds/"
}

activity_datasets = KGQuery(Dataset, query, context).resolve(client)
for dataset in activity_datasets:
    print("* " + dataset.name)
```

10.5 Storing and editing metadata

For those users who have the necessary permissions to store and edit metadata in the Knowledge Graph, **fairgraph** objects can be created or edited in Python, and then saved back to the Knowledge Graph, e.g.:

```
from fairgraph.core import Person, Organization, use_namespace
from fairgraph.commons import Address

use_namespace("neuralactivity")

mgm = Organization("Metro-Goldwyn-Mayer")
mgm.save(client)
author = Person("Laurel", "Stan", "laurel@example.com", affiliation=mgm)
author.save(client)
```

```
mgm.address = Address(locality='Hollywood', country='United States')
mgm.save(client)
```

10.6 Getting help

In case of questions about **fairgraph**, please e-mail support@humanbrainproject.eu. If you find a bug or would like to suggest an enhancement or new feature, please open a ticket in the issue tracker.

10.7 Acknowledgements

This open source software code was developed in part or in whole in the Human Brain Project, funded from the European Union's Horizon 2020 Framework Programme for Research and Innovation under Specific Grant Agreements No. 720270 and No. 785907 (Human Brain Project SGA1 and SGA2).

Python Module Index

f

Index

Q

R

S

T

U

V