

---

# **fairgraph Documentation**

***Release 0.5.1***

**Andrew P. Davison, Onur Ates, Yann Zerlaut, Glynis Mattheisen**

**Nov 26, 2019**



---

## Contents:

---

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>About the Knowledge Graph</b>	<b>5</b>
<b>3</b>	<b>Querying the Knowledge Graph</b>	<b>7</b>
<b>4</b>	<b>Creating and updating metadata nodes</b>	<b>11</b>
<b>5</b>	<b>Metadata domains</b>	<b>13</b>
<b>6</b>	<b>Access permissions</b>	<b>49</b>
<b>7</b>	<b>Contributing to fairgraph</b>	<b>51</b>
<b>8</b>	<b>Getting help</b>	<b>53</b>
<b>9</b>	<b>Authors / contributors</b>	<b>55</b>
<b>10</b>	<b>Quickstart</b>	<b>57</b>
	<b>Python Module Index</b>	<b>61</b>
	<b>Index</b>	<b>63</b>



**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
```





---

### 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

---

---

## 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:

```
from jupyter_collab_storage import oauth_token_handler
token = oauth_token_handler.get_token()
```

If working outside the Collaboratory, you can obtain a token from <https://nexus-iam.humanbrainproject.org/v0/oauth2/authorize>. We suggest you then save it as an environment variable, e.g. at a shell prompt:

```
export HBP_AUTH_TOKEN=eyJhbGciOi...
```

and then in Python:

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

Once you have a token:

```
from fairgraph import KGClient

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 domain, e.g. “electrophysiology” or “brainsimulation”. For a full list of domains, see

*Metadata domains.*

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

```
>>> from fairgraph import electrophysiology

>>> electrophysiology.list_kg_classes()
[fairgraph.electrophysiology.BrainSlicingActivity,
 fairgraph.electrophysiology.IntraCellularSharpElectrodeExperiment,
 fairgraph.electrophysiology.IntraCellularSharpElectrodeRecordedCell,
 fairgraph.electrophysiology.IntraCellularSharpElectrodeRecordedCellCollection,
 fairgraph.electrophysiology.IntraCellularSharpElectrodeRecordedSlice,
 fairgraph.electrophysiology.IntraCellularSharpElectrodeRecording,
 fairgraph.electrophysiology.MultiChannelMultiTrialRecording,
 fairgraph.electrophysiology.PatchClampActivity,
 fairgraph.electrophysiology.PatchClampExperiment,
 fairgraph.electrophysiology.PatchedCell,
 fairgraph.electrophysiology.PatchedCellCollection,
 fairgraph.electrophysiology.PatchedSlice,
 fairgraph.electrophysiology.QualifiedMultiTraceGeneration,
 fairgraph.electrophysiology.QualifiedTraceGeneration,
 fairgraph.electrophysiology.Slice,
 fairgraph.electrophysiology.StepCurrentStimulus,
 fairgraph.electrophysiology.Trace]
```

### 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.electrophysiology import PatchedCell

cells = PatchedCell.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.

```
cells = PatchedCell.list(client, from_index=50, size=50)
```

This returns 50 nodes starting with the 50th. To see how many nodes there are in total:

```
PatchedCell.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 patched cells from the CA1 region of the hippocampus in the mouse:

```
from fairgraph.common import BrainRegion, Species

hippocampus_cells = PatchedCell.list(client,
                                     brain_region=BrainRegion("hippocampus CA1"),
                                     species=Species("Mus musculus"))
```

**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 `terms()` method, e.g. `BrainRegion.terms()`, `Species.terms()`

## 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:

```
cell_of_interest = PatchedCell.by_name('hbp00011_Sub3_Samp2_ExpE10', client)
cell_of_interest = PatchedCell.from_id("8512c3a3-eee3-4c64-acbf-850ab0bd42ee", 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.:

```
>>> cell_of_interest.id
'https://nexus.humanbrainproject.org/v0/data/neuralactivity/experiment/patchedcell/v0.1.0/8512c3a3-eee3-4c64-acbf-850ab0bd42ee'

>>> cell_of_interest.uuid
'8512c3a3-eee3-4c64-acbf-850ab0bd42ee'

>>> cell_of_interest.brain_location
BrainRegion('hippocampus CA1', 'http://purl.obolibrary.org/obo/UBERON_0003881')

>>> cell_of_interest.cell_type
CellType('hippocampus CA1 pyramidal cell', 'http://uri.neuinfo.org/nif/nifstd/sao830368389')
```

Connections between graph nodes are also available as attributes:

```
>>> cell_of_interest.collection
KGQuery([<class 'fairgraph.electrophysiology.PatchedCellCollection'>], {'path':
  'prov:hadMember', 'op': 'eq', 'value': 'https://nexus.humanbrainproject.org/v0/data/neuralactivity/experiment/patchedcell/v0.1.0/8512c3a3-eee3-4c64-acbf-850ab0bd42ee'})
```

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.:

```
>>> cell_collection = cell_of_interest.collection.resolve(client)

>>> patched_slice = cell_collection.slice.resolve(client)
```

(continues on next page)

(continued from previous page)

```
>>> original_slice = patched_slice.slice.resolve(client)

>>> subject = original_slice.subject.resolve(client)

>>> subject.name
'hbp00011_Sub3'

>>> subject.species
Species('Mus musculus', 'http://purl.obolibrary.org/obo/NCBITaxon_10090')

>>> subject.sex
Sex('female', 'schema:Female')

>>> subject.age
Age(QuantitativeValue(3.0 'months'), 'Post-natal')
```

This could be chained together in a single line!

```
>>> subject = cell_of_interest.collection.resolve(client).slice.resolve(client).slice.
↳ resolve(client).subject.resolve(client)
```

---

**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:

```
PatchedCell.set_strict_mode(False)
```

---

## 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.modelvalidation import AnalysisResult

result = AnalysisResult(
    name="inter-spike-interval histograms from subject #f2009a33, white-noise_
↪stimulation",
    result_file="isi_f2009a33_wn.txt"
)
result.save(client)
```

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

```
result.description = "ISIs from 32 neurons, first column is bin left edges, remaining_
↪columns one per neuron"
result.save(client)
```

### 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. In the case of `AnalysisResult`, above, those fields are *name* and *timestamp*.

---

**Note:** at present, the only way to know which subset of fields are used in this query is to view the sourcecode, and inspect the `_existence_query()` method.

---

## 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.



## 5.1 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

**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 protocol.

#### 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 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

- **alternatives** (*KGObject*) –
- **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** (*HBPCComponent*) –
- **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*) –

**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*) –
- **modelformat** (*ModelFormat*) –
- **modelscope** (*ModelScope*) –
- **publication** (*Publication*) –
- **study\_target** (*StudyTarget*) –

**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
```

## 5.3 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.Trace(name, data_location, generated_by, generation_metadata, channel, data_unit, time_step, part_of=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*) –

```
class fairgraph.electrophysiology.MultiChannelMultiTrialRecording(name, data_location, generated_by, generation_metadata, channel_names, data_unit, time_step, 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 ch 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*) –
- **generation\_metadata** (*QualifiedMultiTraceGeneration*) –
- **channel\_names** (*str*) –
- **data\_unit** (*str*) –
- **time\_step** (*QuantitativeValue*) –
- **part\_of** (*Dataset*) –

```
class fairgraph.electrophysiology.PatchedCell(name, brain_location, collection=None,
                                              cell_type=None, experiments=None,
                                              pipette_id=None, seal_resistance=None,
                                              pipette_resistance=None, liquid_
                                              uid_junction_potential=None, label-
                                              ing_compound=None, reversal_
                                              potential_cl=None, id=None, in-
                                              stance=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*) –
- **labeling\_compound** (*str*) –
- **reversal\_potential\_cl** (*QuantitativeValue*) –

```
classmethod list(client, size=100, from_index=0, api='query', scope='released', re-
                  solved=False, **filters)
```

List all objects of this type in the Knowledge Graph

```
class fairgraph.electrophysiology.Slice(id=None, instance=None, **properties)
```

Bases: fairgraph.base.KGObject

A brain slice.

#### Parameters

- **name** (*str*) –
- **subject** (*Subject*) –

- **brain\_slicing\_activity** ([BrainSlicingActivity](#)) –

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

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.BrainSlicingActivity (subject, slices,
                                                    brain_location=None,
                                                    slicing_plane=None,
                                                    slicing_angle=None,
                                                    cutting_solution=None,
                                                    cutting_thickness=None,
                                                    start_time=None, people=None,
                                                    id=None, instance=None)
```

Bases: `fairgraph.base.KGObject`

The activity of cutting brain tissue into slices.

#### Parameters

- **subject** ([Subject](#)) –
- **slices** ([Slice](#)) –
- **brain\_location** ([BrainRegion](#)) –
- **slicing\_plane** (*str*) –
- **slicing\_angle** (*float*) –
- **cutting\_solution** (*str*) –
- **cutting\_thickness** ([QuantitativeValue](#)) –
- **start\_time** (*datetime*) –
- **people** ([Person](#)) –

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

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.PatchedSlice (name, slice, recorded_cells, recording_activity=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](#)) –

```
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.PatchClampActivity(name, slice, recorded_slice,
                                                    protocol=None, people=None,
                                                    id=None, instance=None)
```

Bases: `fairgraph.base.KGObject`

A patch clamp recording session.

#### Parameters

- **name** (`str`) –
- **slice** (`Slice`) –
- **recorded\_slice** (`PatchedSlice`) –
- **protocol** (`str`) –
- **people** (`Person`) –

```
class fairgraph.electrophysiology.PatchClampExperiment(name, recorded_cell, stimulus=None, traces=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`) –
- **stimulus** (`StimulusType`) –
- **traces** (`Trace`, `MultiChannelMultiTrialRecording`) –

```
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, holding_potential=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`) –

- **holding\_potential** (*QuantitativeValue*) –

```
class fairgraph.electrophysiology.ImplantedBrainTissue (id=None, instance=None,  
                                                    **properties)
```

Bases: *fairgraph.base.KGObject*

docstring

#### Parameters

- **name** (*str*) –
- **subject** (*Subject*) –

**resolve** (*client*)

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.ElectrodeImplantationActivity (subject, im-  
                                                                planted_brain_tissues,  
                                                                brain_location,  
                                                                start_time=None,  
                                                                end_time=None,  
                                                                people=None,  
                                                                id=None, in-  
                                                                stance=None)
```

Bases: *fairgraph.base.KGObject*

docstring

#### Parameters

- **subject** (*Subject*) –
- **implanted\_brain\_tissues** (*ImplantedBrainTissue*) –
- **brain\_location** (*BrainRegion*) –
- **start\_time** (*datetime*) –
- **end\_time** (*datetime*) –
- **people** (*Person*) –

**resolve** (*client*)

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.ExtracellularElectrodeExperiment (name,  
                                                                recorded_cell,  
                                                                stimu-  
                                                                lus=None,  
                                                                traces=None,  
                                                                id=None,  
                                                                in-  
                                                                stance=None)
```

Bases: *fairgraph.electrophysiology.PatchClampExperiment*

docstring

#### Parameters

- **name** (*str*) –
- **recorded\_cell** (*ImplantedBrainTissue*) –
- **stimulus** (*StimulusType*) –

```
    • traces (Trace, MultiChannelMultiTrialRecording) –  
classmethod list (client, size=100, api='nexus', **filters)  
    List all objects of this type in the Knowledge Graph  
class fairgraph.electrophysiology.IntraCellularSharpElectrodeRecordedCell (name,  
                                                                           brain_location,  
                                                                           col-  
                                                                           lec-  
                                                                           tion=None,  
                                                                           cell_type=None,  
                                                                           ex-  
                                                                           per-  
                                                                           i-  
                                                                           ments=None,  
                                                                           pipette_id=None,  
                                                                           seal_resistance=None,  
                                                                           pipette_resistance=None,  
                                                                           liq-  
                                                                           uid_junction_potential=None,  
                                                                           la-  
                                                                           bel-  
                                                                           ing_compound=None,  
                                                                           re-  
                                                                           ver-  
                                                                           sal_potential_cl=None,  
                                                                           id=None,  
                                                                           in-  
                                                                           stance=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,  
                                                                    slice,  
                                                                    recorded_slice,  
                                                                    pro-  
                                                                    to-  
                                                                    col=None,  
                                                                    peo-  
                                                                    ple=None,  
                                                                    id=None,  
                                                                    in-  
                                                                    stance=None)
```

Bases: *fairgraph.electrophysiology.PatchClampActivity*

A sharp-electrode recording session.

#### Parameters

- **name** (*str*) –
- **slice** (*Slice*) –
- **recorded\_slice** (*IntraCellularSharpElectrodeRecordedSlice*) –
- **protocol** (*str*) –
- **people** (*Person*) –

```
class fairgraph.electrophysiology.IntraCellularSharpElectrodeRecordedCellCollection(name,  
                                                                                   cells,  
                                                                                   slice=None,  
                                                                                   id=None,  
                                                                                   in-  
                                                                                   stance=None)
```

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,  
                                                                              record-  
                                                                              ing_activity=None,  
                                                                              id=None,  
                                                                              in-  
                                                                              stance=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,  
                                                                           stim-  
                                                                           u-  
                                                                           lus=None,  
                                                                           traces=None,  
                                                                           id=None,  
                                                                           in-  
                                                                           stance=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*) –
- **stimulus** (*StimulusType*) –
- **traces** (*Trace*) –

```
classmethod list (client, size=100, from_index=0, api='query', scope='released', re-  
                  solved=False, **filters)
```

List all objects of this type in the Knowledge Graph

```
class fairgraph.electrophysiology.QualifiedMultiTraceGeneration (name, stimu-  
                                                                    lus_experiment,  
                                                                    sweeps, hold-  
                                                                    ing_potential=None,  
                                                                    id=None, in-  
                                                                    stance=None)
```

Bases: *fairgraph.base.KGObject*

#### Parameters

- **name** (*str*) –
- **stimulus\_experiment** (*ExtracellularElectrodeExperiment, IntraCellularSharpElectrodeExperiment, PatchClampExperiment*) –
- **sweeps** (*int*) –
- **holding\_potential** (*QuantitativeValue*) –

```
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.

## 5.4 brainsimulation

Metadata for model building, simulation and validation.

```
class fairgraph.brainsimulation.ModelProject(name, owners, authors, descrip-
tion, date_created, private, col-
lab_id=None, alias=None, organiza-
tion=None, pla_components=None,
brain_region=None, species=None, cell-
type=None, abstraction_level=None,
model_of=None, old_uuid=None,
parents=None, instances=None, im-
ages=None, id=None, instance=None)
```

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

Representation of a neuroscience model or modelling project.

We distinguish a model in an abstract sense (this class), which may have multiple parameteriza-  
tions 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** (*int*) –
- **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, times-
tamp=None, brain_region=None,
species=None, model_of=None, re-
lease=None, part_of=None, de-
scription=None, parameters=None,
old_uuid=None, id=None, in-
stance=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*) –

```
class fairgraph.brainsimulation.MEModel (name, e_model, morphology, main_script, ver-  
sion, timestamp=None, brain_region=None,  
species=None, model_of=None, release=None,  
part_of=None, description=None, parame-  
ters=None, old_uuid=None, id=None, in-  
stance=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*) –
- **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.AnalysisResult (name, result_file=None, times-  
                                         tamp=None, derived_from=None,  
                                         attributed_to=None, description=None,  
                                         id=None, instance=None)
```

Bases: fairgraph.base.KGObject

The result of a data analysis.

For example a graph, a histogram, etc. The result is expected to be stored either in a local file or in a web-accessible location with a direct URL.

Note that local results files smaller than 1 MB in size will be uploaded and stored within the Knowledge Graph. Larger files must be stored elsewhere.

#### Parameters

- **name** (*str*) –
- **result\_file** (*Distribution*, *str*) –
- **timestamp** (*datetime*) –
- **derived\_from** (*KGObject*) –
- **attributed\_to** (*Person*) –
- **description** (*str*) –

```
save (client)  
      docstring
```

```
class fairgraph.brainsimulation.ValidationTestDefinition (id=None, in-  
                                                         stance=None, **proper-  
                                                         ties)
```

Bases: fairgraph.base.KGObject, fairgraph.brainsimulation.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\_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** (*int*, *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*) –

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.

## 5.5 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.Software(name, version, summary=None, description=None,
                                identifier=None, citation=None, license=None, re-
                                lease_date=None, previous_version=None, contrib-
                                utors=None, project=None, image=None, down-
                                load_url=None, access_url=None, categories=None,
                                subcategories=None, operating_system=None, re-
                                lease_notes=None, requirements=None, copyright=None,
                                components=None, part_of=None, funding=None,
                                languages=None, features=None, keywords=None,
                                is_free=None, homepage=None, documentation=None,
                                help=None, id=None, instance=None)
```

Bases: fairgraph.base.KGObject

```
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.

## 5.6 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, sex=None, strain=None, death_date=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*) –
- **sex** (*Sex*) –
- **age** (*Age*) –
- **death\_date** (*date*) –

```
class fairgraph.core.Organization(name, address=None, parent=None, id=None, in-
                                stance=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'*)

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.**Protocol** (*name*, *steps*, *materials*, *author*, *date\_published*, *identifier*,  
*id=None*, *instance=None*)

Bases: fairgraph.base.KGObject

An experimental protocol.

**classmethod from\_kg\_instance** (*instance*, *client*, *resolved=False*)

docstring

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

Bases: fairgraph.base.KGObject

**class** fairgraph.core.**Material** (*name*, *molar\_weight*, *formula*, *stock\_keeping\_unit*, *identifier*, *ven-*  
*dor*)

Bases: object

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

**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.

## 5.7 commons

**class** fairgraph.common.**Address** (*locality*, *country*)

Bases: fairgraph.base.StructuredMetadata

```

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

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

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

    An inbred sub-population within a species.

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

    A sub-structure or region with the brain.

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

    A type of neuron or glial cell.

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

    docstring

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

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

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

    docstring

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

    docstring

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

```

#### Parameters

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

**fairgraph** currently provides the following modules:

*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@humanbrainproject.eu](mailto:support@humanbrainproject.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 e-mail [support@humanbrainproject.eu](mailto: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](#).



---

### 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 Sud
2. Human-Computer Interaction, Department IV, Computer Science, Universität Trier

### 9.1 Acknowledgements

`<div></div>`

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).



### 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 HBP Identity account is needed.

If you are working in a Collaboratory Jupyter notebook:

```
from jupyter_collab_storage import oauth_token_handler
token = oauth_token_handler.get_token()
```

If working outside the Collaboratory, we recommend you obtain a token from <https://nexus-iam.humanbrainproject.org/v0/oauth2/authorize> and save it as an environment variable, e.g. at a shell prompt:

```
export HBP_AUTH_TOKEN=eyJhbGciOi...
```

and then in Python:

```
token = os.environ['HBP_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, currently *commons*, *core*, *brainsimulation*, *electrophysiology*, *software*, *minds* and *uniminds*. For example:

```
from fairgraph.common import BrainRegion
from fairgraph.electrophysiology import PatchedCell
```

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

```
cells_in_ca1 = PatchedCell.list(client, brain_region=BrainRegion("hippocampus CA1"))
```

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

```
cell_of_interest = PatchedCell.from_uuid("153ec151-b1ae-417b-96b5-4ce9950a3c56",
→client)
```

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

```
example_cell = cells_in_ca1[3]
experiment = example_cell.experiments.resolve(client)
trace = experiment.traces.resolve(client)
```

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

```
print(example_cell.cell_type)
print(example_cell.reversal_potential_cl)
print(trace.time_step)
print(trace.data_unit)
```

You can also access any associated data:

```
import requests
import numpy as np
from io import BytesIO

download_url = trace.data_location['downloadURL']
data = np.genfromtxt(BytesIO(requests.get(download_url).content))
```

## 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",
    "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.common 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](mailto: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).



### f

`fairgraph.brainsimulation`, 38  
`fairgraph.common`s, 46  
`fairgraph.core`, 45  
`fairgraph.electrophysiology`, 30  
`fairgraph.minds`, 13  
`fairgraph.software`, 44  
`fairgraph.uniminds`, 19



## A

AbstractionLevel (class in fairgraph.common), 47  
 AbstractionLevel (class in fairgraph.uniminds), 20  
 Activity (class in fairgraph.minds), 13  
 Address (class in fairgraph.common), 46  
 Age (class in fairgraph.common), 47  
 AgeCategory (class in fairgraph.minds), 13  
 AgeCategory (class in fairgraph.uniminds), 20  
 AnalysisResult (class in fairgraph.brainsimulation), 42

## B

BrainRegion (class in fairgraph.common), 47  
 BrainSlicingActivity (class in fairgraph.electrophysiology), 33  
 BrainStructure (class in fairgraph.uniminds), 20

## C

CellType (class in fairgraph.common), 47  
 CellularTarget (class in fairgraph.uniminds), 20  
 Collection (class in fairgraph.core), 46  
 Country (class in fairgraph.uniminds), 20

## D

Dataset (class in fairgraph.minds), 14  
 Dataset (class in fairgraph.uniminds), 21  
 Disability (class in fairgraph.uniminds), 21  
 Doi (class in fairgraph.uniminds), 22

## E

ElectrodeImplantationActivity (class in fairgraph.electrophysiology), 35  
 EmbargoStatus (class in fairgraph.minds), 15  
 EmbargoStatus (class in fairgraph.uniminds), 22  
 EModel (class in fairgraph.brainsimulation), 41  
 EthicsApproval (class in fairgraph.minds), 14  
 EthicsApproval (class in fairgraph.uniminds), 22  
 EthicsAuthority (class in fairgraph.minds), 14  
 EthicsAuthority (class in fairgraph.uniminds), 22

ExperimentalPreparation (class in fairgraph.uniminds), 22  
 ExtracellularElectrodeExperiment (class in fairgraph.electrophysiology), 35

## F

fairgraph.brainsimulation (module), 38  
 fairgraph.common (module), 46  
 fairgraph.core (module), 45  
 fairgraph.electrophysiology (module), 30  
 fairgraph.minds (module), 13  
 fairgraph.software (module), 44  
 fairgraph.uniminds (module), 19  
 File (class in fairgraph.minds), 15  
 File (class in fairgraph.uniminds), 23  
 FileAssociation (class in fairgraph.minds), 15  
 FileAssociation (class in fairgraph.uniminds), 23  
 FileBundle (class in fairgraph.uniminds), 23  
 FileBundleGroup (class in fairgraph.uniminds), 23  
 Format (class in fairgraph.minds), 16  
 from\_kg\_instance() (fairgraph.core.Protocol class method), 46  
 from\_kg\_instance() (fairgraph.electrophysiology.PatchClampExperiment class method), 34  
 FundingInformation (class in fairgraph.uniminds), 24

## G

Genotype (class in fairgraph.uniminds), 24

## H

Handedness (class in fairgraph.uniminds), 24  
 HBPCComponent (class in fairgraph.uniminds), 24

## I

Identifier (class in fairgraph.core), 46  
 ImplantedBrainTissue (class in fairgraph.electrophysiology), 35

IntraCellularSharpElectrodeExperiment  
(class in fairgraph.electrophysiology), 38

IntraCellularSharpElectrodeRecordedCell  
(class in fairgraph.electrophysiology), 36

IntraCellularSharpElectrodeRecordedCellCollection  
(class in fairgraph.electrophysiology), 37

IntraCellularSharpElectrodeRecordedSlice  
(class in fairgraph.electrophysiology), 37

IntraCellularSharpElectrodeRecording  
(class in fairgraph.electrophysiology), 36

## L

License (class in fairgraph.common), 47

License (class in fairgraph.minds), 16

License (class in fairgraph.uniminds), 25

list() (fairgraph.core.Person class method), 46

list() (fairgraph.electrophysiology.ExtracellularElectrodeExperiment  
class method), 36

list() (fairgraph.electrophysiology.IntraCellularSharpElectrodeExperiment  
class method), 38

list() (fairgraph.electrophysiology.PatchClampExperiment  
class method), 34

list() (fairgraph.electrophysiology.PatchedCell class  
method), 32

list\_kg\_classes() (in module fair-  
graph.brainsimulation), 44

list\_kg\_classes() (in module fairgraph.core), 46

list\_kg\_classes() (in module fair-  
graph.electrophysiology), 38

list\_kg\_classes() (in module fairgraph.minds),  
19

list\_kg\_classes() (in module fairgraph.software),  
45

list\_kg\_classes() (in module fair-  
graph.uniminds), 30

## M

Material (class in fairgraph.core), 46

me() (fairgraph.core.Person class method), 46

MEModel (class in fairgraph.brainsimulation), 40

Method (class in fairgraph.minds), 16

Method (class in fairgraph.uniminds), 25

MethodCategory (class in fairgraph.uniminds), 25

methods() (fairgraph.minds.Dataset method), 15

MimeType (class in fairgraph.uniminds), 25

Modality (class in fairgraph.minds), 16

ModelFormat (class in fairgraph.uniminds), 26

ModelInstance (class in fairgraph.brainsimulation),  
39

ModelInstance (class in fairgraph.uniminds), 26

ModelProject (class in fairgraph.brainsimulation),  
38

ModelScope (class in fairgraph.common), 47

ModelScope (class in fairgraph.uniminds), 26

ModelScript (class in fairgraph.brainsimulation), 41

Morphology (class in fairgraph.brainsimulation), 41

MultiChannelMultiTrialRecording (class in  
fairgraph.electrophysiology), 31

## O

OperatingSystem (class in fairgraph.software), 44

Organization (class in fairgraph.core), 45

Organization (class in fairgraph.uniminds), 26

## P

ParcellationAtlas (class in fairgraph.minds), 16

ParcellationRegion (class in fairgraph.minds), 16

PatchClampActivity (class in fair-  
graph.electrophysiology), 34

PatchClampExperiment (class in fair-  
graph.electrophysiology), 34

PatchedCell (class in fairgraph.electrophysiology),  
32

PatchedCellCollection (class in fair-  
graph.electrophysiology), 33

PatchedSlice (class in fairgraph.electrophysiology),  
33

Person (class in fairgraph.core), 45

Person (class in fairgraph.minds), 13

Person (class in fairgraph.uniminds), 19

PLAComponent (class in fairgraph.minds), 17

Preparation (class in fairgraph.minds), 17

ProgrammingLanguage (class in fair-  
graph.software), 44

Project (class in fairgraph.uniminds), 27

Project (in module fairgraph.minds), 19

Protocol (class in fairgraph.core), 46

Protocol (class in fairgraph.minds), 17

Publication (class in fairgraph.minds), 17

Publication (class in fairgraph.uniminds), 27

PublicationId (class in fairgraph.uniminds), 27

PublicationIdType (class in fairgraph.uniminds),  
28

## Q

QualifiedMultiTraceGeneration (class in fair-  
graph.electrophysiology), 38

QualifiedTraceGeneration (class in fair-  
graph.electrophysiology), 34

QuantitativeValue (class in fairgraph.common),  
47

QuantitativeValueRange (class in fair-  
graph.common), 47

## R

ReferenceSpace (class in fairgraph.minds), 17

resolve() (fairgraph.core.Person method), 46

[resolve\(\)](#) (*fairgraph.electrophysiology.BrainSlicingActivity* method), 33  
[resolve\(\)](#) (*fairgraph.electrophysiology.ElectrodeImplantationActivity* method), 35  
[resolve\(\)](#) (*fairgraph.electrophysiology.ImplantedBrainTissueValidationTestDefinition* method), 35  
[resolve\(\)](#) (*fairgraph.electrophysiology.Slice* method), 33  
[Resolve](#) (class in *fairgraph.minds*), 17

## S

[Sample](#) (class in *fairgraph.minds*), 18  
[save\(\)](#) (*fairgraph.brainsimulation.AnalysisResult* method), 42  
[Sex](#) (class in *fairgraph.common*s), 47  
[Sex](#) (class in *fairgraph.minds*), 18  
[Sex](#) (class in *fairgraph.uniminds*), 28  
[Slice](#) (class in *fairgraph.electrophysiology*), 32  
[Software](#) (class in *fairgraph.software*), 44  
[SoftwareAgent](#) (class in *fairgraph.minds*), 18  
[SoftwareCategory](#) (class in *fairgraph.software*), 44  
[Species](#) (class in *fairgraph.common*s), 46  
[Species](#) (class in *fairgraph.minds*), 18  
[Species](#) (class in *fairgraph.uniminds*), 28  
[SpecimenGroup](#) (class in *fairgraph.minds*), 18  
[StimulusType](#) (class in *fairgraph.common*s), 47  
[Strain](#) (class in *fairgraph.common*s), 47  
[Strain](#) (class in *fairgraph.uniminds*), 28  
[StudyTarget](#) (class in *fairgraph.uniminds*), 28  
[StudyTargetSource](#) (class in *fairgraph.uniminds*), 29  
[StudyTargetType](#) (class in *fairgraph.uniminds*), 29  
[Subject](#) (class in *fairgraph.core*), 45  
[Subject](#) (class in *fairgraph.minds*), 19  
[Subject](#) (class in *fairgraph.uniminds*), 29  
[SubjectGroup](#) (class in *fairgraph.uniminds*), 29

## T

[TissueSample](#) (class in *fairgraph.uniminds*), 30  
[Trace](#) (class in *fairgraph.electrophysiology*), 31

## U

[UnimindsObject](#) (class in *fairgraph.uniminds*), 19  
[UnimindsOption](#) (class in *fairgraph.uniminds*), 19  
[use\\_namespace\(\)](#) (in module *fairgraph.brainsimulation*), 44  
[use\\_namespace\(\)](#) (in module *fairgraph.core*), 46  
[use\\_namespace\(\)](#) (in module *fairgraph.electrophysiology*), 38  
[use\\_namespace\(\)](#) (in module *fairgraph.software*), 45

## V

[ValidationActivity](#) (class in *fairgraph.brainsimulation*), 44