

---

# **pulsarpy Documentation**

***Release 1.2.0***

**Nathaniel Watson**

**Aug 06, 2019**



---

## Contents

---

<b>1</b>	<b>Scripts</b>	<b>3</b>
<b>2</b>	<b>Client API Modules</b>	<b>5</b>
<b>3</b>	<b>Indices and tables</b>	<b>25</b>
	<b>Python Module Index</b>	<b>27</b>
	<b>Index</b>	<b>29</b>



This project is read from [GitHub](#)



### 1.1 get\_id\_from\_name.py

Given a file with a column of names of records of a given type of model in Pulsar, fetches the IDs. The record IDs are written to a new file.

```
usage: get_id_from_name.py [-h] -m MODEL -i INFILE -o OUTFILE
```

#### 1.1.1 Named Arguments

<b>-m, --model</b>	The name of the Rails model class of the records for which we need to get to IDs.
<b>-i, --infile</b>	The input file containing record names, one per row.
<b>-o, --outfile</b>	The output file containing a column of record IDs. Each row corresponds to the same row in the input file.

### 1.2 get\_missing.py

Given an input file containing record names, one per row, indicates whether the record exists by outputting a 1 if it exists and a 0 if it doesn't. The input file of record names will be first sorted and deduplicated.

```
usage: get_missing.py [-h] -m MODEL -i INFILE -o OUTFILE
```

#### 1.2.1 Named Arguments

<b>-m, --model</b>	The name of the model to import the records to, i.e. Biosample or CrisprModification.
<b>-i, --infile</b>	One or more record names, one per line.

**-o, --outfile**                      The output file with two columns: 1) name, 2) status (1 for present, 0 for absent).

## 1.3 tab\_import.py

Given a tab-delimited sheet, creates new records of the specified Model into Pulsar LIMS or updates existing records if the patch option is provided. Array values should be comma-delimited as this program will split on the comma and add array literals. Array fields are only assumed when the field name has an 'ids' suffix.

```
usage: tab_import.py [-h] -m MODEL -i INFILE [-p] [--no-append] [--skip-dups]
                    [-u]
```

### 1.3.1 Named Arguments

<b>-m, --model</b>	The name of the model to import the records to, i.e. Biosample or CrisprModification.
<b>-i, --infile</b>	The tab-delimited input file containing records (1 per row). There must be a field-header line as the first row, and field names must match record attribute names. Any field names that start with a '#' will be skipped. Any rows that start with a '#' will also be skipped (apart from the header line).
<b>-p, --patch</b>	Presence of this option means to PATCH instead of POST. The input file must contain a column by the name of record_id to designate the existing record to PATCH. You can use a record's primary ID or name as the identifier.  Default: False
<b>--no-append</b>	This option only has meaning when the -patch option is also specified. It's presence indicates to not extend array values with the content to be patched when dealing with array data types. Thus, if you don't want to overwrite the existing value for arrays, then skip this option.  Default: False
<b>--skip-dups</b>	If an attempt to POST a duplicate record is made, the server will respond with a ActiveRecord::RecordNotUnique exception. Including this flag indicates to catch this exception and skip on to the next record to POST.  Default: False
<b>-u, --upstream-ids</b>	If patching records and you are providing the record identifiers using the value of a records upstream_identifier attribute, set this to True.  Default: False



### 2.1 pulsarpy.elasticsearch\_utils

**exception** pulsarpy.elasticsearch\_utils.**MultipleHitsException**

Bases: Exception

Raised when a search that is expected to return at most 1 hit has more than this.

### 2.2 pulsarpy.models

A client that contains classes named after each model in Pulsar to handle RESTful communication with the Pulsar API.

**exception** pulsarpy.models.**RecordNotFound**

Bases: Exception

” Raised when looking up a record by name, id, etc. and it isn’t found on the server.

**exception** pulsarpy.models.**RecordNotUnique**

Bases: Exception

Raised when posting a record and the Rails server returns with the exception ActiveRecord::RecordNotUnique.

pulsarpy.models.**remove\_model\_prefix**(uid)

Removes the optional model prefix from the given primary ID. For example, given the biosample record whose primary ID is 8, and given the fact that the model prefix for the Biosample model is “B-“, the record ID can be specified either as 8 or B-8. However, the model prefix needs to be stripped off prior to making API calls.

**class** pulsarpy.models.**Model** (uid=None, upstream=None)

Bases: object

The superclass of all model classes. A model subclass is defined for each Rails model. An instance of a model class represents a record of the given Rails model.

Subclasses don't typically define their own init method, but if they do, they need to make a call to 'super' to run the init method defined here as well.

Subclasses must be instantiated with the `rec_id` argument set to a record's ID. A GET will immediately be done and the record's attributes will be stored in the `self.attrs dict`. The record's attributes can be accessed as normal instance attributes (i.e. `record.name`) rather than explicitly indexing the `attrs` dictionary, thanks to the employment of `__getattr__()`. Similarly, record attributes can be updated via normal assignment operations, i.e. (`record.name = "bob"`), thanks to employment of `__setattr__()`.

#### Required Environment Variables:

- 1) PULSAR\_API\_URL
- 2) PULSAR\_TOKEN

Find the record of the given model specified by `self.MODEL_NAME`. The record can be looked up in a few ways, depending on which argument is specified (`uid` or `upstream`). If both are specified, then the `upstream` argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its `upstream_identifier` attribute.

**UPSTREAM\_ATTR = 'upstream\_identifier'**

Most models have an attribute called `upstream_identifier` that is used to store the value of the record in an "upstream" database that is submitted to, i.e. the ENCODE Portal. Not all models have this attribute since not all are used for submission to an upstream portal.

**FKEY\_MAP = {}**

Abstract attribute of type *dict* that each subclass should fulfill if it has any foreign keys. Each key is a foreign key name and the value is the class name of the model it refers to.

**PULSAR\_LIMS\_PREFIX = 'p'**

A prefix that can be added in front of record IDs, names, model-record ID. This is useful when its necessary to add emphasis that these records exist or came from Pulsar ( i.e. when submitting them to an upstream database.

**debug\_logger = <Logger ppy\_debug (DEBUG)>**

This class adds a file handler, such that all messages sent to it are logged to this file in addition to STDOUT.

**error\_logger = <Logger ppy\_error (ERROR)>**

A logging instance with a file handler for logging terse error messages. The log file resides locally within the directory specified by the constant `p.LOG_DIR`. Accepts messages `>= logging.ERROR`.

**post\_logger = <Logger ppy\_post (INFO)>**

A logging instance with a file handler for logging successful POST operations. The log file resides locally within the directory specified by the constant `p.LOG_DIR`. Accepts messages `>= logging.INFO`.

**ES = <pulsarpy.elasticsearch\_utils.Connection object>**

Connection to Elasticsearch. Expects that the environment variables `ES_URL`, `ES_USER`, and `ES_PW` are set, which signify the Elasticsearch cluster URL, login username and login password, respectively.

**classmethod replace\_name\_with\_id (name)**

Used to replace a foreign key reference using a name with an ID. Works by searching the record in Pulsar

and expects to find exactly one hit. First, will check if the foreign key reference is an integer value and if so, returns that as it is presumed to be the foreign key.

#### Raises

- *pulsarpy.elasticsearch\_utils.MultipleHitsException* – Multiple hits were returned from the name search.
- *pulsarpy.models.RecordNotFound* – No results were produced from the name search.

#### **classmethod** `add_model_name_to_payload(payload)`

Checks whether the model name in question is in the payload. If not, the entire payload is set as a value of a key by the name of the model. This method is useful when some server-side Rails API calls expect the parameters to include the parameterized model name. For example, server-side endpoints that handle the updating of a biosample record or the creation of a new biosample record will expect the payload to be of the form:

```
{ "biosample": {
  "name": "new biosample",
  "donor": 3,
  ...
}
```

**Parameters** `payload` – *dict*. The data to send in an HTTP request.

**Returns** *dict*.

#### **abbrev\_id()**

This method is called when posting to the ENCODE Portal to grab an alias for the record being submitted. The alias here is composed of the record ID in Pulsar (i.e. B-1 for the Biosample with ID 1). However, the record ID is prefixed with a 'p' to designate that this record was submitted from Pulsar. This is used to generate a unique alias considering that we used to use a different LIMS (Syapse) to submit records. Many of the models in Syapse used the same model prefix as is used in Pulsar, i.e. (B)Biosample and (L)Library. Thus, w/o the 'p' prefix, the same alias could be generated in Pulsar as a previous one used in Syapse.

#### **delete()**

Deletes the record.

#### **classmethod** `find_by(payload, require=False)`

Searches the model in question by AND joining the query parameters.

Implements a Railsy way of looking for a record using a method by the same name and passing in the query as a dict. as well. Only the first hit is returned, and there is no particular ordering specified in the server-side API method.

#### Parameters

- **payload** – *dict*. The attributes of a record to restrict the search to.
- **require** – *bool*. True means to raise a *pulsarpy.models.RecordNotFound* exception if no record is found.

**Returns** The JSON serialization of the record, if any, found by the API call. *None*: If the API call didn't return any results.

**Return type** *dict*

**Raises** *pulsarpy.models.RecordNotFound* – No records were found, and the *require* parameter is True.

**classmethod** `find_by_or(payload)`

Searches the model in question by OR joining the query parameters.

Implements a Railsy way of looking for a record using a method by the same name and passing in the query as a string (for the OR operator joining to be specified).

Only the first hit is returned, and there is not particular ordering specified in the server-side API method.

**Parameters** `payload` – *dict*. The attributes of a record to search for by using OR operator joining for each query parameter.

**Returns** The JSON serialization of the record, if any, found by the API call. *None*: If the API call didn't return any results.

**Return type** *dict*

**classmethod** `index()`

Fetches all records.

**Returns** *dict*. The JSON formatted response.

**Raises** *requests.exceptions.HTTPError* – The status code is not ok.

**patch** (*payload*, *append\_to\_arrays=True*)

Patches current record and updates the current instance's 'attrs' attribute to reflect the new changes.

**Parameters** – *hash*. **This will be JSON-formatted prior to sending the request.** (*payload*) –

**Returns** *dict*. The JSON formatted response.

**Raises** *requests.exceptions.HTTPError* – The status code is not ok.

**classmethod** `set_id_in_fkeys(payload)`

Looks for any keys in the payload that end with either `_id` or `_ids`, signaling a foreign key field. For each foreign key field, checks whether the value is using the name of the record or the actual primary ID of the record (which may include the model abbreviation, i.e. B-1). If the former case, the name is replaced with the record's primary ID.

**Parameters** `payload` – *dict*. The payload to POST or PATCH.

**Returns** *dict*. The payload.

**classmethod** `pre_post(payload)`

This class method should be implemented in subclasses only when there is sub-class specific logic that needs to occur prior to using the generalized post class method defined below in this class.

**classmethod** `post(payload)`

Posts the data to the specified record.

**Parameters** `payload` – *dict*. This will be JSON-formatted prior to sending the request.

**Returns** *dict*. The JSON formatted response.

**Raises**

- *Requests.exceptions.HTTPError* – The status code is not ok.
- *RecordNotUnique* – The Rails server returned the exception *ActiveRecord::RecordNotUnique*.

**classmethod** `log_error(msg)`

Logs the provided error message to both the error logger and the debug logger logging instances.

**Parameters** `msg` – *str*. The error message to log.

**static write\_response\_html\_to\_file** (*response, filename*)

An aid in troubleshooting internal application errors, i.e. <Response [500]>, to be mainly beneficial when developing the server-side API. This method will write the response HTML for viewing the error details in the browser.

#### Parameters

- **response** – *requests.models.Response* instance.
- **filename** – *str*. The output file name.

**class** pulsarpy.models.**Address** (*uid=None, upstream=None*)

Bases: [\*pulsarpy.models.Model\*](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.**Antibody** (*uid=None, upstream=None*)

Bases: [\*pulsarpy.models.Model\*](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.**AntibodyPurification** (*uid=None, upstream=None*)

Bases: [\*pulsarpy.models.Model\*](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.**Atacseq** (*uid=None, upstream=None*)

Bases: [\*pulsarpy.models.Model\*](#)

Find the record of the given model specified by `self.MODEL_NAME`. The record can be looked up in a few ways, depending on which argument is specified (`uid` or `upstream`). If both are specified, then the `upstream` argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its `upstream_identifier` attribute.

**class** `pulsarpy.models.Barcode` (*uid=None, upstream=None*)

Bases: `pulsarpy.models.Model`

Find the record of the given model specified by `self.MODEL_NAME`. The record can be looked up in a few ways, depending on which argument is specified (`uid` or `upstream`). If both are specified, then the `upstream` argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its `upstream_identifier` attribute.

**class** `pulsarpy.models.Biosample` (*uid=None, upstream=None*)

Bases: `pulsarpy.models.Model`

Find the record of the given model specified by `self.MODEL_NAME`. The record can be looked up in a few ways, depending on which argument is specified (`uid` or `upstream`). If both are specified, then the `upstream` argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its `upstream_identifier` attribute.

**parent\_ids** ()

Returns an array of parent Biosample IDs. If the current Biosample has a `part_of` relationship, the Biosampled referenced there will be returned. Otherwise, if the current Biosample was generated from a pool of Biosamples (`pooled_from_biosample_ids`), then those will be returned. Otherwise, the result will be an empty array.

**find\_first\_wt\_parent** (*with\_ip=False*)

Recursively looks at the `part_of` parent ancestry line (ignoring `pooled_from` parents) and returns a parent Biosample ID if its `wild_type` attribute is `True`.

**Parameters** `with_ip` – *bool*. `True` means to restrict the search to the first parental Wild Type that also has an Immunoblot linked to it, which may serve as a control between another immunoblot. For example, it could be useful to compare the target protein bands in Immunoblots between a Wild Type sample and a CRISPR eGFP-tagged gene in a descendent sample.

#### Returns

There isn't a WT parent, or there is but not one with an Immunoblot linked to it (if the *with\_ip* parameter is set to True).

*int*: The ID of the WT parent.

**Return type** *False*

**get\_latest\_library()**

Returns the associated library having the largest ID (the most recent one created). It's possible for a Biosample in Pulsar to have more than one Library, but this is rare.

**class** pulsarpy.models.BiosampleOntology (*uid=None, upstream=None*)

Bases: *pulsarpy.models.Model*

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (*uid* or *upstream*). If both are specified, then the *upstream* argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its *upstream\_identifier* attribute.

**class** pulsarpy.models.BiosampleTermName (*uid=None, upstream=None*)

Bases: *pulsarpy.models.Model*

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (*uid* or *upstream*). If both are specified, then the *upstream* argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its *upstream\_identifier* attribute.

**class** pulsarpy.models.BiosampleType (*uid=None, upstream=None*)

Bases: *pulsarpy.models.Model*

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (*uid* or *upstream*). If both are specified, then the *upstream* argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its *upstream\_identifier* attribute.

**class** pulsarpy.models.Batch (*uid=None, upstream=None*)

Bases: *pulsarpy.models.Model*

Find the record of the given model specified by `self.MODEL_NAME`. The record can be looked up in a few ways, depending on which argument is specified (`uid` or `upstream`). If both are specified, then the `upstream` argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its `upstream_identifier` attribute.

**class** `pulsarpy.models.BatchItem` (*uid=None, upstream=None*)

Bases: `pulsarpy.models.Model`

Find the record of the given model specified by `self.MODEL_NAME`. The record can be looked up in a few ways, depending on which argument is specified (`uid` or `upstream`). If both are specified, then the `upstream` argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its `upstream_identifier` attribute.

**class** `pulsarpy.models.ChipseqExperiment` (*uid=None, upstream=None*)

Bases: `pulsarpy.models.Model`

Find the record of the given model specified by `self.MODEL_NAME`. The record can be looked up in a few ways, depending on which argument is specified (`uid` or `upstream`). If both are specified, then the `upstream` argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its `upstream_identifier` attribute.

**paired\_input\_control\_map** ()

Creates a dict. where each key is the ID of a non-control Biosample record on the `ChipseqExperiment`, and each value is the

**Returns** *dict*.

**class** `pulsarpy.models.DataStorage` (*uid=None, upstream=None*)

Bases: `pulsarpy.models.Model`

Find the record of the given model specified by `self.MODEL_NAME`. The record can be looked up in a few ways, depending on which argument is specified (`uid` or `upstream`). If both are specified, then the `upstream` argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the



record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.

- **upstream** – If set, then the record will be searched on its `upstream_identifier` attribute.

**class** `pulsarpy.models.DataStorageProvider` (*uid=None, upstream=None*)

Bases: `pulsarpy.models.Model`

Find the record of the given model specified by `self.MODEL_NAME`. The record can be looked up in a few ways, depending on which argument is specified (`uid` or `upstream`). If both are specified, then the `upstream` argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its `upstream_identifier` attribute.

**class** `pulsarpy.models.Document` (*uid=None, upstream=None*)

Bases: `pulsarpy.models.Model`

Find the record of the given model specified by `self.MODEL_NAME`. The record can be looked up in a few ways, depending on which argument is specified (`uid` or `upstream`). If both are specified, then the `upstream` argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its `upstream_identifier` attribute.

**classmethod** `upload` (*path, document\_type, is\_protocol, description=""*)

#### Parameters

- **path** – *str*. The path to the document to upload.
- **document\_type** – *str*. `DocumentType` identified by the value of its name attribute.
- **is\_protocol** – *bool*.
- **description** – *str*.

**class** `pulsarpy.models.DocumentType` (*uid=None, upstream=None*)

Bases: `pulsarpy.models.Model`

Find the record of the given model specified by `self.MODEL_NAME`. The record can be looked up in a few ways, depending on which argument is specified (`uid` or `upstream`). If both are specified, then the `upstream` argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its `upstream_identifier` attribute.

```
class pulsarpy.models.Unit (uid=None, upstream=None)
```

Bases: [`pulsarpy.models.Model`](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

```
class pulsarpy.models.ConstructTag (uid=None, upstream=None)
```

Bases: [`pulsarpy.models.Model`](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

```
class pulsarpy.models.CrisprConstruct (uid=None, upstream=None)
```

Bases: [`pulsarpy.models.Model`](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

```
class pulsarpy.models.CrisprModification (uid=None, upstream=None)
```

Bases: [`pulsarpy.models.Model`](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

```
class pulsarpy.models.Donor (uid=None, upstream=None)
```

Bases: [\*pulsarpy.models.Model\*](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

```
class pulsarpy.models.DonorConstruct (uid=None, upstream=None)
```

Bases: [\*pulsarpy.models.Model\*](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

```
class pulsarpy.models.FileReference (uid=None, upstream=None)
```

Bases: [\*pulsarpy.models.Model\*](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

```
class pulsarpy.models.Gel (uid=None, upstream=None)
```

Bases: [\*pulsarpy.models.Model\*](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.GelImage (uid=None, upstream=None)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.GelLane (uid=None, upstream=None)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.Immunoblot (uid=None, upstream=None)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.Library (uid=None, upstream=None)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**classmethod** `pre_post` (*payload*)

A wrapper over `Model.post()` that handles the case where a Library has a PairedBarcode and the user may have supplied the PairedBarcode in the form of index1-index2, i.e. GATTTCCTCA-GGCGTCTCA. This isn't the PairedBarcode's record name or a record ID, thus `Model.post()` won't be able to figure out the PairedBarcode's ID to substitute in the payload (via a call to `cls.replace_name_with_id()`). Thus, this wrapper will attempt to replace a PairedBarcode sequence in the payload with a PairedBarcode ID, then pass the payload off to `Model.post()`.

**class** `pulsarpy.models.LibraryFragmentationMethod` (*uid=None, upstream=None*)

Bases: `pulsarpy.models.Model`

Find the record of the given model specified by `self.MODEL_NAME`. The record can be looked up in a few ways, depending on which argument is specified (`uid` or `upstream`). If both are specified, then the `upstream` argument will be ignored.

**Parameters**

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its `upstream_identifier` attribute.

**class** `pulsarpy.models.NucleicAcidTerm` (*uid=None, upstream=None*)

Bases: `pulsarpy.models.Model`

Find the record of the given model specified by `self.MODEL_NAME`. The record can be looked up in a few ways, depending on which argument is specified (`uid` or `upstream`). If both are specified, then the `upstream` argument will be ignored.

**Parameters**

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its `upstream_identifier` attribute.

**class** `pulsarpy.models.PairedBarcode` (*uid=None, upstream=None*)

Bases: `pulsarpy.models.Model`

Find the record of the given model specified by `self.MODEL_NAME`. The record can be looked up in a few ways, depending on which argument is specified (`uid` or `upstream`). If both are specified, then the `upstream` argument will be ignored.

**Parameters**

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its `upstream_identifier` attribute.

**class** `pulsarpy.models.Plate` (*uid=None, upstream=None*)

Bases: `pulsarpy.models.Model`

Find the record of the given model specified by `self.MODEL_NAME`. The record can be looked up in a few ways, depending on which argument is specified (`uid` or `upstream`). If both are specified, then the `upstream` argument will be ignored.

### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.SequencingCenter (uid=None, upstream=None)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.SequencingLibraryPrepKit (uid=None, upstream=None)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.SequencingRequest (uid=None, upstream=None)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**get\_library\_barcode\_sequence\_hash** (inverse=False)

Calls the SequencingRequest's get\_library\_barcode\_sequence\_hash server-side endpoint to create a hash of the form {LibraryID -> barcode\_sequence} for all Libraries on the SequencingRequest.

**Parameters** **inverse** – *bool*. True means to inverse the key and value pairs such that the barcode sequence serves as the key.

Returns: *dict*.

**class** pulsarpy.models.**SequencingPlatform** (*uid=None, upstream=None*)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.**SequencingRun** (*uid=None, upstream=None*)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**library\_sequencing\_result** (*library\_id*)

Fetches a SequencingResult record for a given Library ID.

**library\_sequencing\_results** ()

Generates a dict. where each key is a Library ID on the SequencingRequest and each value is the associated SequencingResult. Libraries that aren't yet with a SequencingResult are not included in the dict.

**class** pulsarpy.models.**SequencingResult** (*uid=None, upstream=None*)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.**Shipping** (*uid=None, upstream=None*)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.



### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.SingleCellSorting (uid=None, upstream=None)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.Target (uid=None, upstream=None)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.Treatment (uid=None, upstream=None)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.TreatmentTermName (uid=None, upstream=None)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.



**Parameters**

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**class** pulsarpy.models.**User** (uid=None, upstream=None)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

**Parameters**

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its upstream\_identifier attribute.

**archive\_user** (user\_id)

Archives the user with the specified user ID.

**Parameters** **user\_id** – *int*. The ID of the user to archive.

**Returns** None.

**Return type** *NoneType*

**unarchive\_user** (user\_id)

Unarchives the user with the specified user ID.

**Parameters** **user\_id** – *int*. The ID of the user to unarchive.

**Returns** None.

**Return type** *NoneType*

**generate\_api\_key** ()

Generates an API key for the user, replacing any existing one.

**Returns** The new API key.

**Return type** *str*

**remove\_api\_key** ()

Removes the user's existing API key, if present, and sets the current instance's 'api\_key' attribute to the empty string.

**Returns** None.

**Return type** *NoneType*

**class** pulsarpy.models.**Vendor** (uid=None, upstream=None)

Bases: [pulsarpy.models.Model](#)

Find the record of the given model specified by self.MODEL\_NAME. The record can be looked up in a few ways, depending on which argument is specified (uid or upstream). If both are specified, then the upstream argument will be ignored.

**Parameters**

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its `upstream_identifier` attribute.

**class** `pulsarpy.models.Well` (*uid=None, upstream=None*)

Bases: `pulsarpy.models.Model`

Find the record of the given model specified by `self.MODEL_NAME`. The record can be looked up in a few ways, depending on which argument is specified (`uid` or `upstream`). If both are specified, then the `upstream` argument will be ignored.

#### Parameters

- **uid** – The database identifier of the record to fetch, which can be specified either as the primary id (i.e. 8) or the model prefix plus the primary id (i.e. B-8). Could also be the record's name if it has a name attribute (not all models do) and if so will be converted to the record ID.
- **upstream** – If set, then the record will be searched on its `upstream_identifier` attribute.

## 2.3 pulsarpy

The official Python client for Pulsar LIMS.

#### Required Environment Variables:

- 1) `PULSAR_API_URL`
- 2) `PULSAR_TOKEN`

`pulsarpy.LOG_DIR = 'Pulsarpy_Logs'`

The directory that contains the log files created by the *Model* class.

`pulsarpy.DEBUG_LOGGER_NAME = 'ppy_debug'`

The name of the debug logging instance.

`pulsarpy.ERROR_LOGGER_NAME = 'ppy_error'`

The name of the error logging instance created in the `pulsarpy.models.Model` class. and referenced elsewhere.

`pulsarpy.POST_LOGGER_NAME = 'ppy_post'`

The name of the POST logging instance created in the `pulsarpy.models.Model` class. and referenced elsewhere.

`pulsarpy.debug_logger = <Logger ppy_debug (DEBUG)>`

A logging instance that logs all messages sent to it to STDOUT.

## 2.4 pulsarpy.utils

`pulsarpy.utils.send_mail` (*form, from\_name*)

Sends a mail using the configured mail server for Pulsar. See mailgun documentation at [https://documentation.mailgun.com/en/latest/user\\_manual.html#sending-via-api](https://documentation.mailgun.com/en/latest/user_manual.html#sending-via-api) for specifics.

**Parameters** `form` – *dict*. The mail form fields, i.e. 'to', 'from', ...

**Returns** *requests.models.Response* instance.

**Raises**

- *requests.exceptions.HTTPError* – The status code is not ok.
- *Exception* – The environment variable MAILGUN\_DOMAIN or MAILGUN\_API\_KEY isn't set.

Example:

```
payload = {
    "from"="{ } <mailgun@{}>".format(from_name, pulsarpy.MAIL_DOMAIN),
    "subject": "mailgun test",
    "text": "howdy there",
    "to": "nathankw@stanford.edu",
}
send_mail(payload)
```

`pulsarpy.utils.get_exp_of_biosample(biosample_rec)`

Determines whether the biosample is part of a ChipseqExperiment or SingleCellSorting Experiment, and if so, returns the associated experiment as a `models.Model` instance that is one of those two classes. The biosample is determined to be part of a ChipseqExperiment if the `Biosample.chipseq_experiment_id` attribute is set, meaning that the biosample can be associated to the ChipseqExperiment as a replicate via any of the following ChipseqExperiment attributes:

`ChipseqExperiment.replicates` `ChipseqExperiment.control_replicates`

The biosample will be determined to be part of a SingleCellSorting experiment if the `Biosample.sorting_biosample_single_cell_sorting` attribute is set, meaning that it is the `SingleCellSorting.sorting_biosample`.

**Parameters** `biosample_rec` – *dict*. A Biosample record as returned by instantiating *models.Biosample*.

**Raises** *Exception* – An experiment is not associated to this biosample.



## CHAPTER 3

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### p

- `pulsarpy`, [22](#)
- `pulsarpy.elasticsearch_utils`, [5](#)
- `pulsarpy.models`, [5](#)
- `pulsarpy.utils`, [22](#)





## A

abbrev\_id() (*pulsarpy.models.Model* method), 7  
 add\_model\_name\_to\_payload() (*pulsarpy.models.Model* class method), 7  
 Address (*class in pulsarpy.models*), 9  
 Antibody (*class in pulsarpy.models*), 9  
 AntibodyPurification (*class in pulsarpy.models*), 9  
 archive\_user() (*pulsarpy.models.User* method), 21  
 Atacseq (*class in pulsarpy.models*), 9

## B

Barcode (*class in pulsarpy.models*), 10  
 Batch (*class in pulsarpy.models*), 11  
 BatchItem (*class in pulsarpy.models*), 12  
 Biosample (*class in pulsarpy.models*), 10  
 BiosampleOntology (*class in pulsarpy.models*), 11  
 BiosampleTermName (*class in pulsarpy.models*), 11  
 BiosampleType (*class in pulsarpy.models*), 11

## C

ChipseqExperiment (*class in pulsarpy.models*), 12  
 ConstructTag (*class in pulsarpy.models*), 14  
 CrisprConstruct (*class in pulsarpy.models*), 14  
 CrisprModification (*class in pulsarpy.models*), 14

## D

DataStorage (*class in pulsarpy.models*), 12  
 DataStorageProvider (*class in pulsarpy.models*), 13  
 debug\_logger (*in module pulsarpy*), 22  
 debug\_logger (*pulsarpy.models.Model* attribute), 6  
 DEBUG\_LOGGER\_NAME (*in module pulsarpy*), 22  
 delete() (*pulsarpy.models.Model* method), 7  
 Document (*class in pulsarpy.models*), 13  
 DocumentType (*class in pulsarpy.models*), 13  
 Donor (*class in pulsarpy.models*), 14  
 DonorConstruct (*class in pulsarpy.models*), 15

## E

error\_logger (*pulsarpy.models.Model* attribute), 6  
 ERROR\_LOGGER\_NAME (*in module pulsarpy*), 22  
 ES (*pulsarpy.models.Model* attribute), 6

## F

FileReference (*class in pulsarpy.models*), 15  
 find\_by() (*pulsarpy.models.Model* class method), 7  
 find\_by\_or() (*pulsarpy.models.Model* class method), 7  
 find\_first\_wt\_parent() (*pulsarpy.models.Biosample* method), 10  
 FKEY\_MAP (*pulsarpy.models.Model* attribute), 6

## G

Gel (*class in pulsarpy.models*), 15  
 GelImage (*class in pulsarpy.models*), 15  
 GelLane (*class in pulsarpy.models*), 16  
 generate\_api\_key() (*pulsarpy.models.User* method), 21  
 get\_exp\_of\_biosample() (*in module pulsarpy.utils*), 23  
 get\_latest\_library() (*pulsarpy.models.Biosample* method), 11  
 get\_library\_barcode\_sequence\_hash() (*pulsarpy.models.SequencingRequest* method), 18

## I

Immunoblot (*class in pulsarpy.models*), 16  
 index() (*pulsarpy.models.Model* class method), 8

## L

Library (*class in pulsarpy.models*), 16  
 library\_sequencing\_result() (*pulsarpy.models.SequencingRun* method), 19  
 library\_sequencing\_results() (*pulsarpy.models.SequencingRun* method), 19

LibraryFragmentationMethod (class in pulsarpy.models), 17  
 LOG\_DIR (in module pulsarpy), 22  
 log\_error() (pulsarpy.models.Model class method), 8

## M

Model (class in pulsarpy.models), 5  
 MultipleHitsException, 5

## N

NucleicAcidTerm (class in pulsarpy.models), 17

## P

paired\_input\_control\_map() (pulsarpy.models.ChipseqExperiment method), 12  
 PairedBarcode (class in pulsarpy.models), 17  
 parent\_ids() (pulsarpy.models.Biosample method), 10  
 patch() (pulsarpy.models.Model method), 8  
 Plate (class in pulsarpy.models), 17  
 post() (pulsarpy.models.Model class method), 8  
 post\_logger (pulsarpy.models.Model attribute), 6  
 POST\_LOGGER\_NAME (in module pulsarpy), 22  
 pre\_post() (pulsarpy.models.Library class method), 16  
 pre\_post() (pulsarpy.models.Model class method), 8  
 PULSAR\_LIMS\_PREFIX (pulsarpy.models.Model attribute), 6  
 pulsarpy (module), 22  
 pulsarpy.elasticsearch\_utils (module), 5  
 pulsarpy.models (module), 5  
 pulsarpy.utils (module), 22

## R

RecordNotFound, 5  
 RecordNotUnique, 5  
 remove\_api\_key() (pulsarpy.models.User method), 21  
 remove\_model\_prefix() (in module pulsarpy.models), 5  
 replace\_name\_with\_id() (pulsarpy.models.Model class method), 6

## S

send\_mail() (in module pulsarpy.utils), 22  
 SequencingCenter (class in pulsarpy.models), 18  
 SequencingLibraryPrepKit (class in pulsarpy.models), 18  
 SequencingPlatform (class in pulsarpy.models), 19  
 SequencingRequest (class in pulsarpy.models), 18  
 SequencingResult (class in pulsarpy.models), 19

SequencingRun (class in pulsarpy.models), 19  
 set\_id\_in\_fkeys() (pulsarpy.models.Model class method), 8  
 Shipping (class in pulsarpy.models), 19  
 SingleCellSorting (class in pulsarpy.models), 20

## T

Target (class in pulsarpy.models), 20  
 Treatment (class in pulsarpy.models), 20  
 TreatmentTermName (class in pulsarpy.models), 20

## U

unarchive\_user() (pulsarpy.models.User method), 21  
 Unit (class in pulsarpy.models), 13  
 upload() (pulsarpy.models.Document class method), 13  
 UPSTREAM\_ATTR (pulsarpy.models.Model attribute), 6  
 User (class in pulsarpy.models), 21

## V

Vendor (class in pulsarpy.models), 21

## W

Well (class in pulsarpy.models), 22  
 write\_response\_html\_to\_file() (pulsarpy.models.Model static method), 8