OmicIDX

Sean Davis

Aug 30, 2019

CONTENTS

1 Documentation **3**
1.1 OmicIDX dataset on Bigquery 3
1.2 Commandline Interface 4
1.3 Genomic Metadata in OmicIDX 7
1.4 omicidx 7
1.5 Ipython notebooks 34
1.6 Rmarkdown workbooks 34

2 Indices and tables **35**

Python Module Index **37**

Index **39**

The OmicIDX project provides easy programmatic access to metadata associated with publicly available genomic data.

- Flexible search and query
- Analytics and data mining
- Data munging and mashups with other data resources (ontologies, for example)
- Performant bulk access via application programming interfaces
- Computable formats (json, avro) for data scientists
- Drive bulk processing of genomics datasets

CHAPTER ONE

DOCUMENTATION

1.1 OmicIDX dataset on Bigquery

Bigquery is a cloud-based, fully-managed data warehouse and relational database and analytical engine available from Google. Bigquery is capable of storing and querying very large datasets (think billions or even more rows, TB of data).

The OmicIDX data are accessible as a *public* dataset on Bigquery. For users who wish to use SQL for search, analytics, or other analytical workflows, Bigquery provides a ready-made solution that includes an online web-based query tool, a command-line tool, and clients in many languages including R, Python, Go, and Java.

For OmicIDX, though, Bigquery offers additional capabilities including:

- Export of large query results to Google cloud storage
- Joining OmicIDX with any other database tables available to the user, including private ones.
- Public data warehouse for publicly available genomics datasets
- Integration with other Google Cloud Platform services like machine learning, Dataflow, or natural language processing

Note: While OmicIDX data on Bigquery is public, accessing Bigquery requires a Google Cloud Platform account and an active billing project. A new account comes with free credits.

1.1.1 Bigquery OmicIDX tables

The OmicIDX data in Bigquery comprise a set of tables that mirror the data model available from NCBI.

Repository	Table Name	Accessions like:
sra	sra_study	SRP…
sra	sra_sample	SRS…
sra	sra_experiment	SRX…
sra	sra_run	SRR…
biosample	biosample	SAMN…

Unlike in a traditional relational database, these tables include “nested” columns.

1.2 Commandline Interface

1.2.1 omicidx-cli

Command-line interface for omicidx processing

```
omicidx-cli [OPTIONS] COMMAND [ARGS]...
```

biosample

Use these commands to process biosample records.

```
omicidx-cli biosample [OPTIONS] COMMAND [ARGS]...
```

download

Download biosample xml file from NCBI

```
omicidx-cli biosample download [OPTIONS]
```

etl-to-public

ETL process (copy) from etl schema to public

```
omicidx-cli biosample etl-to-public [OPTIONS]
```

gcs-dump

Write json.gz format of biosample to gcs

```
omicidx-cli biosample gcs-dump [OPTIONS]
```

gcs-to-elasticsearch

```
omicidx-cli biosample gcs-to-elasticsearch [OPTIONS]
```

load

Load the gcs biosample.json file to bigquery

```
omicidx-cli biosample load [OPTIONS]
```

parse

Parse xml to json, output to stdout

```
omicidx-cli biosample parse [OPTIONS] BIOSAMPLE_FILE
```

Arguments

BIOSAMPLE_FILE
Required argument

upload

Download biosample xml file from NCBI

```
omicidx-cli biosample upload [OPTIONS]
```

sra

Use these commands to process SRA metadata

```
omicidx-cli sra [OPTIONS] COMMAND [ARGS]...
```

download

Downloads the files necessary to build the SRA json conversions of the XML files.

Files will be placed in the <mirrordir> directory. Mirrordirs have the format *NCBI_SRA_Mirroring_20190801_Full*.

```
omicidx-cli sra download [OPTIONS] MIRRORDIR
```

Arguments

MIRRORDIR
Required argument

gcs-dump

Write json.gz format of sra entities to gcs

```
omicidx-cli sra gcs-dump [OPTIONS]
```

load-sra-data-to-bigquery

Load gcs files to Bigquery

```
omicidx-cli sra load-sra-data-to-bigquery [OPTIONS]
```

parse-entity

SRA XML to JSON

Transforms an SRA XML mirroring metadata file into corresponding JSON format files. JSON is line-delimited JSON (not an array).

```
omicidx-cli sra parse-entity [OPTIONS] ENTITY
```

Arguments

ENTITY
Required argument

sra-bigquery-for-elasticsearch

ETL queries to create elasticsearch tables in bigquery

```
omicidx-cli sra sra-bigquery-for-elasticsearch [OPTIONS]
```

sra-gcs-to-elasticsearch

ETL query to public schema for all SRA entities

```
omicidx-cli sra sra-gcs-to-elasticsearch [OPTIONS]
```

sra-to-bigquery

ETL query to public schema for all SRA entities

```
omicidx-cli sra sra-to-bigquery [OPTIONS]
```

upload

Upload SRA json to GCS

```
omicidx-cli sra upload [OPTIONS] MIRRORDIR
```

Arguments

MIRRORDIR
Required argument

1.3 Genomic Metadata in OmicIDX

1.4 omicidx

1.4.1 omicidx package

Subpackages

omicidx.data package

Subpackages

omicidx.data.bigquery_schemas package

Module contents

Module contents

omicidx.model package

Submodules

omicidx.model.etl module

class `omicidx.model.etl.`**`SraAccession`**(***kwargs*)

Bases: `sqlalchemy.ext.declarative.api.Base`

`accession`

`alias`

`bases`

`bio_project`

`bio_sample`

`center`

`experiment`

`loaded`

`md5sum`

`published`

`received`

`replaced_by`

`sample`

`spots`

`status`

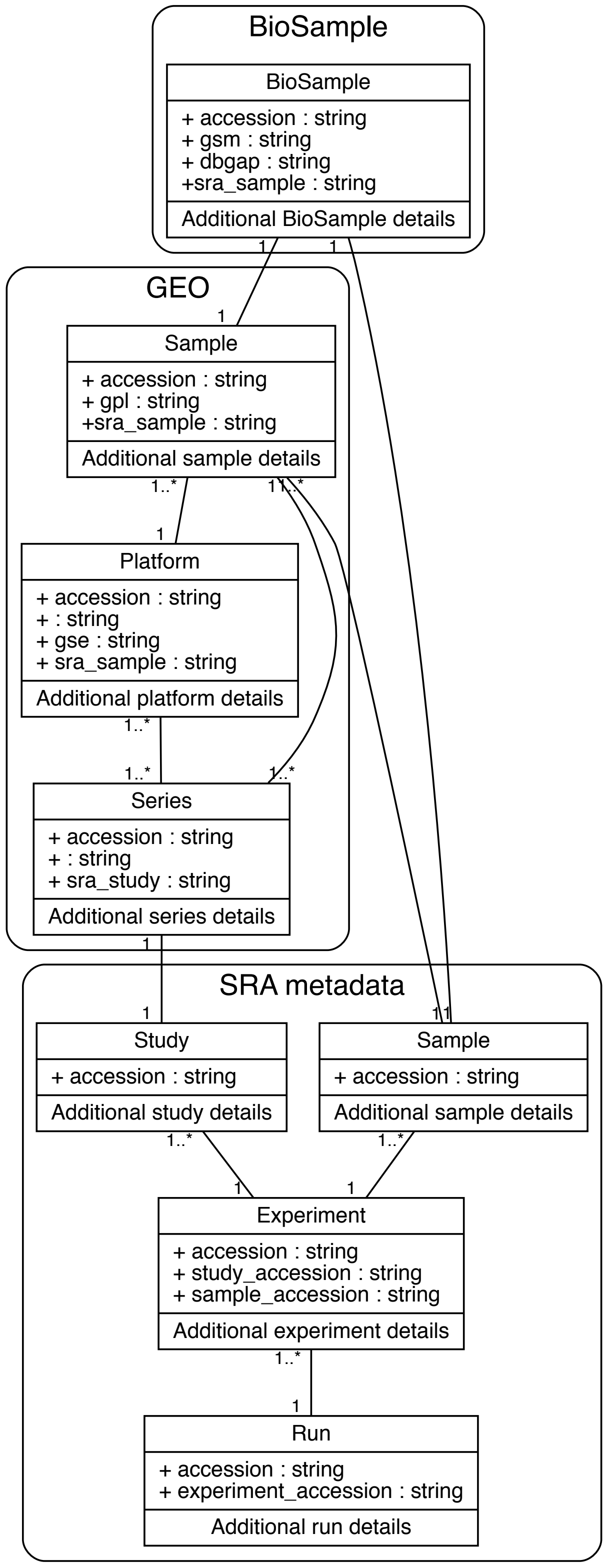

Fig. 1: "The OmicIDX data model approximates the data model from NCBI. Data from NCBI SRA, GEO, and BioSample are all represented."

study

submission

type

updated

visibility

omicidx.model.etl_schema module

class `omicidx.model.etl_schema.`**`ExperimentJson`**`(`***kwargs*`)`
Bases: `sqlalchemy.ext.declarative.api.Base`

doc

id

class `omicidx.model.etl_schema.`**`RunJson`**`(`***kwargs*`)`
Bases: `sqlalchemy.ext.declarative.api.Base`

doc

id

class `omicidx.model.etl_schema.`**`SRAExperiment`**`(`***kwargs*`)`
Bases: `sqlalchemy.ext.declarative.api.Base`

accession

study_accession

class `omicidx.model.etl_schema.`**`SRAStudy`**`(`***kwargs*`)`
Bases: `sqlalchemy.ext.declarative.api.Base`

abstract

accession

alias

attributes

bioproject

broker_name

center_name

description

gse_accession

identifiers

published

received

replaced_by

status

study_type

title

updated

visibility

xrefs

class omicidx.model.etl_schema.**SampleJson**(***kwargs*)
Bases: sqlalchemy.ext.declarative.api.Base

doc

id

class omicidx.model.etl_schema.**StudyJson**(***kwargs*)
Bases: sqlalchemy.ext.declarative.api.Base

doc

id

omicidx.model.etl_schema.**main**()

omicidx.model.public module

class omicidx.model.public.**AttributesMixin**
Bases: object

id = Column(None, Integer(), table=None, primary_key=True, nullable=False)

tag = Column(None, Text(), table=None)

value = Column(None, Text(), table=None)

class omicidx.model.public.**Biosample**(***kwargs*)
Bases: sqlalchemy.ext.declarative.api.Base

access

accession

attributes

dbgap

description

gsm

identifiers

is_reference

last_update

model

publication_date

sra_sample

submission_date

taxon_id

taxonomy_name

textsearchable_index_col

title

class omicidx.model.public.**IdentifiersMixin**

Bases: `object`

id = Column(None, Integer(), table=None, primary_key=True, nullable=False)

identifier = Column(None, Text(), table=None)

namespace = Column(None, Text(), table=None)

uuid = Column(None, UUID(as_uuid=True), table=None)

class omicidx.model.public.**SraExperiment**(***kwargs*)

Bases: `sqlalchemy.ext.declarative.api.Base`

accession

alias

attributes

broker_name

center_name

description

design

identifiers

instrument_model

library_construction_protocol

library_layout

library_layout_length

library_layout_orientation

library_layout_sdev

library_name

library_selection

library_source

library_strategy

platform

published

received

replaced_by

sample_accession

status

study_accession

title

updated

visibility

xrefs

class omicidx.model.public.**SraExperimentAttribute**(***kwargs*)
Bases: sqlalchemy.ext.declarative.api.Base, *omicidx.model.public.AttributesMixin*

id

tag

value

class omicidx.model.public.**SraExperimentIdentifier**(***kwargs*)
Bases: sqlalchemy.ext.declarative.api.Base, *omicidx.model.public.IdentifiersMixin*

accession

id

identifier

namespace

uuid

class omicidx.model.public.**SraLibraryLayout**(***kwargs*)
Bases: sqlalchemy.ext.declarative.api.Base

value

class omicidx.model.public.**SraLibrarySelection**(***kwargs*)
Bases: sqlalchemy.ext.declarative.api.Base

value

class omicidx.model.public.**SraLibrarySource**(***kwargs*)
Bases: sqlalchemy.ext.declarative.api.Base

value

class omicidx.model.public.**SraLibraryStrategy**(***kwargs*)
Bases: sqlalchemy.ext.declarative.api.Base

value

class omicidx.model.public.**SraPlatform**(***kwargs*)
Bases: sqlalchemy.ext.declarative.api.Base

value

class omicidx.model.public.**SraRun**(***kwargs*)
Bases: sqlalchemy.ext.declarative.api.Base

accession

alias

attributes

bases

bio_project

broker_name

center_name

experiment_accession

identifiers

loaded

nreads

published

reads

received

replaced_by

run_center

run_date

spot_length

spots

status

updated

visibility

class omicidx.model.public.**SraRunAttribute**(***kwargs*)
Bases: sqlalchemy.ext.declarative.api.Base, *omicidx.model.public.AttributesMixin*

id

tag

value

class omicidx.model.public.**SraSample**(***kwargs*)
Bases: sqlalchemy.ext.declarative.api.Base

accession

alias

attributes

bio_sample

broker_name

center_name

description

gsm

identifiers

organism

published

received

replaced_by

status

study_accession

taxon_id

title

updated

visibility

xrefs

class omicidx.model.public.**SraSampleAttribute**(***kwargs*)

Bases: `sqlalchemy.ext.declarative.api.Base`, *omicidx.model.public.AttributesMixin*

id

tag

value

class omicidx.model.public.**SraSampleIdentifier**(***kwargs*)

Bases: `sqlalchemy.ext.declarative.api.Base`, *omicidx.model.public.IdentifiersMixin*

accession

id

identifier

namespace

uuid

class omicidx.model.public.**SraSampleXref**(***kwargs*)

Bases: `sqlalchemy.ext.declarative.api.Base`, *omicidx.model.public.XrefsMixin*

accession

db

id

identifier

uuid

class omicidx.model.public.**SraStudy**(***kwargs*)

Bases: `sqlalchemy.ext.declarative.api.Base`

abstract

accession

alias

attributes

bioproject

broker_name

center_name

description

gse

identifiers

published

received

replaced_by

status

study_type

title

updated

visibility

xrefs

class omicidx.model.public.**SraStudyAttribute**(***kwargs*)

Bases: sqlalchemy.ext.declarative.api.Base, *omicidx.model.public.AttributesMixin*

id

tag

value

class omicidx.model.public.**SraStudyIdentifier**(***kwargs*)

Bases: sqlalchemy.ext.declarative.api.Base, *omicidx.model.public.IdentifiersMixin*

accession

id

identifier

namespace

uuid

class omicidx.model.public.**SraStudyXref**(***kwargs*)

Bases: sqlalchemy.ext.declarative.api.Base, *omicidx.model.public.XrefsMixin*

accession

db

id

identifier

uuid

class omicidx.model.public.**XrefsMixin**

Bases: object

db = Column(None, Text(), table=None)

id = Column(None, Integer(), table=None, primary_key=True, nullable=False)

identifier = Column(None, Text(), table=None)

uuid = Column(None, UUID(as_uuid=True), table=None)

omicidx.model.public.**create_all_in_schema**(*schema='public2', drop_schema=True*) `omicidx.model.public.`**`to_tsvector_ix`**`(`**columns*`)`

create tsvector string from column names

Module contents

omicidx.schema package

Module contents

omicidx.scripts package

Submodules

omicidx.scripts.cli module

`omicidx.scripts.cli.`**`biosample_to_json`**`(`*biosample_file*`)`

`omicidx.scripts.cli.`**`dateconverter`**`(`*o*`)`

`omicidx.scripts.cli.`**`download_biosample`**`()`

`omicidx.scripts.cli.`**`load_biosample_from_gcs_to_bigquery`**`()`

`omicidx.scripts.cli.`**`upload_biosample`**`()`

omicidx.scripts.omicidx module

`async` `omicidx.scripts.omicidx.`**`consume`**`(`*queue*`)`

`omicidx.scripts.omicidx.`**`dateconverter`**`(`*o*`)`

`async` `omicidx.scripts.omicidx.`**`produce`**`(`*queue*, *queuename*`)`

`async` `omicidx.scripts.omicidx.`**`run`**`(`*queuename*`)`

omicidx.scripts.sra_entity_to_json module

`omicidx.scripts.sra_entity_to_json.`**`main`**`()`

Module contents

omicidx.sra package

Submodules

omicidx.sra.ebiutils module

`class` `omicidx.sra.ebiutils.`**`EBIEna`**

Bases: `object`

get_file_report_for_accession(*accession*)
Get a file report from EBI

Parameters **accession** (`str`) – Any SRA/ENA accession

Returns

Return type a list of rows representing available files for the accession

get_parsed_xml_for_accession(*accession*)

get_xml_for_accession(*accession*)
Get an XML record for a single accession as an ElementTree Node from EBI

Parameters **accession** (`str`) – Any SRA/ENA accession

Returns

Return type an etree ElementNode parsed from the retrieved XML.

omicidx.sra.etl module

ETL from gcs to bigquery

1. create_all_tables()
2. load_accession_table()
3. load_livelist_table()
4. load_all_json_tables()
5. do_all_join_tables()

   # Does copy into SRA schema from etl # AFTER deleting! # may need to delete_table('sra.experiment'), . . . first 1. study_join_accessions() 2. sample_join_accessions() 3. experiment_join_accessions() 4. run_join_accessions()

`omicidx.sra.etl.`**create_all_tables**()

`omicidx.sra.etl.`**create_table**(*table*)

`omicidx.sra.etl.`**delete_table**(*table*)
Just supply schema.tablename as table

`omicidx.sra.etl.`**do_all_join_tables**()

`omicidx.sra.etl.`**experiment_join_accessions**()

`omicidx.sra.etl.`**load_accession_table**()

`omicidx.sra.etl.`**load_all_json_tables**()

`omicidx.sra.etl.`**load_json_table**(*table*)

`omicidx.sra.etl.`**load_livelist_table**()

`omicidx.sra.etl.`**prep_bigquery_schemas**()

`omicidx.sra.etl.`**run_join_accessions**()

`omicidx.sra.etl.`**sample_join_accessions**()

`omicidx.sra.etl.`**study_join_accessions**()

omicidx.sra.pydantic_models module

class omicidx.sra.pydantic_models.**Attribute**(***data*)
Bases: pydantic.main.BaseModel

class omicidx.sra.pydantic_models.**BaseCounts**
Bases: list, typing.Generic

class omicidx.sra.pydantic_models.**BaseQualities**
Bases: list, typing.Generic

class omicidx.sra.pydantic_models.**FileAlternative**(***data*)
Bases: pydantic.main.BaseModel

class omicidx.sra.pydantic_models.**FileSet**(***data*)
Bases: pydantic.main.BaseModel

class omicidx.sra.pydantic_models.**FullSraRun**(***data*)
Bases: *omicidx.sra.pydantic_models.SraRun*

class omicidx.sra.pydantic_models.**Identifier**(***data*)
Bases: pydantic.main.BaseModel

class omicidx.sra.pydantic_models.**LiveList**(***data*)
Bases: pydantic.main.BaseModel

class omicidx.sra.pydantic_models.**RunRead**(***data*)
Bases: pydantic.main.BaseModel

class omicidx.sra.pydantic_models.**SraExperiment**(***data*)
Bases: *omicidx.sra.pydantic_models.LiveList*, pydantic.main.BaseModel

class omicidx.sra.pydantic_models.**SraRun**(***data*)
Bases: *omicidx.sra.pydantic_models.LiveList*, pydantic.main.BaseModel

class omicidx.sra.pydantic_models.**SraSample**(***data*)
Bases: *omicidx.sra.pydantic_models.LiveList*, pydantic.main.BaseModel

class omicidx.sra.pydantic_models.**SraStudy**(***data*)
Bases: *omicidx.sra.pydantic_models.LiveList*, pydantic.main.BaseModel

class omicidx.sra.pydantic_models.**TaxCountAnalysis**(***data*)
Bases: pydantic.main.BaseModel

class omicidx.sra.pydantic_models.**TaxCountEntry**(***data*)
Bases: pydantic.main.BaseModel

class omicidx.sra.pydantic_models.**Xref**(***data*)
Bases: pydantic.main.BaseModel

Module contents

Submodules

omicidx.bigquery_utils module

omicidx.bigquery_utils.**copy_table**(*src_dataset: str*, *dest_dataset: str*, *src_table: str*, *dest_table: str*, *drop=True*) `omicidx.bigquery_utils.`**`load_csv_to_bigquery`**`(`*dataset*, *table*, *uri*, *schema=None*, *drop=True*, ***kwargs*`)`

Load a file from google cloud storage into BigQuery

Parameters

- **`dataset`** (*`str`*) – The Bigquery dataset
- **`table`** (*`str`*) – The Bigquery table
- **`uri`** (*`str`*) – The google cloud storage uri (`gs://....`)
- **`schema`** (List[SchemaField] objects or `None`) – The schema as a list of *bigquery.SchemaField* objects
- **`drop`** (*`boolean`*) – Drop the table or not.

`omicidx.bigquery_utils.`**`load_json_to_bigquery`**`(`*dataset*, *table*, *uri*, *schema=None*, *drop=True*`)`

Load a file from google cloud storage into BigQuery

Parameters

- **`dataset`** (*`str`*) – The Bigquery dataset
- **`table`** (*`str`*) – The Bigquery table
- **`uri`** (*`str`*) – The google cloud storage uri (`gs://....`)
- **`schema`** (List[SchemaField] objects or `None`) – The schema as a list of *bigquery.SchemaField* objects
- **`drop`** (*`boolean`*) – Drop the table or not.

`omicidx.bigquery_utils.`**`parse_bq_json_schema`**`(`*schema_filename*`)`

Convert bigquery JSON file to python Bigquery Schema

Parameters **`schema_filename`** (*`str`*) – A json file with bigquery schema dump

`omicidx.bigquery_utils.`**`query`**`(`*sql: str*`)`

`omicidx.bigquery_utils.`**`query_to_destination`**`(`*dest_dataset: str*, *dest_table: str*, *sql: str*, *drop=True*`)`

`omicidx.bigquery_utils.`**`table_to_gcs`**`(`*dataset*, *table*, *uri*, *gzip=True*`)`

Load a file from google cloud storage into BigQuery

Parameters

- **`dataset`** (*`str`*) – The Bigquery dataset
- **`table`** (*`str`*) – The Bigquery table
- **`uri`** (*`str`*) – The google cloud storage uri (`gs://....`)
- **`gzip`** (*`bool`*) – Compress output with gzip or not

omicidx.biosample module

Biosample parser

Implemented as an iterator

```
>>> import omicidx.biosample as b
>>> for bios in b.BioSampleParser('biosample_set.xml.gz'):
>>>     print(bios.as_json())
```

class omicidx.biosample.**BioSample**

Bases: dict

BioSample class

as_json(*indent=None*)

class omicidx.biosample.**BioSampleParser**(*fname*)

Bases: object

Parse a BioSample xml file.

Implemented as an iterator

omicidx.db module

class omicidx.db.**OmicIDXDb**(*config*)

Bases: object

create_table()

get_conn(**args, **kwargs*)

save_object(*obj*)

omicidx.elasticsearch_utils module

omicidx.elasticsearch_utils.**bulk_index**(*fname, index, id_field=None, **kwargs*)

omicidx.elasticsearch_utils.**bulk_index_from_gcs**(*bucket, prefix, index, id_field=None, **kwargs*)

Perform bulk indexing from a set of gcs blobs

Parameters

- **bucket** (*str*) – GCS bucket name
- **prefix** (*str*) – The prefix string (without wildcard) to get the right blobs
- **index** (*str*) – The elasticsearch index name
- **id_field** (*str*) – The id field name (default None) that will be used as the *_id* field in elasticsearch

omicidx.gcs_utils module

Utilities for working with google cloud storage

omicidx.gcs_utils.**list_blobs**(*bucket_name, prefix*)

list blobs in a bucket given a prefix

Parameters

- **name** (*bucket*) –
- **prefix** (*str*) – a *matching* string

omicidx.gcs_utils.**upload_blob_to_gcs**(*bucket_name*, *source_file_name*, *destination_blob_name*)

Uploads a file to the bucket.

Parameters

- **name** (*bucket*) –
- **source_file_name** (*str*) – A local filename
- **destination_blob_name** (*str*) – The `path` of the object in storage

omicidx.geometa module

Usage

python -m omicidx.geometa –help

class omicidx.geometa.**GEOBase**

Bases: `object`

GEO Base class

as_dict()

Return object as a dict

class omicidx.geometa.**GEOChannel**(*d*, *ch*)

Bases: *omicidx.geometa.GEOBase*

Captures a single channel from a GSM

class omicidx.geometa.**GEOContact**(*d*)

Bases: *omicidx.geometa.GEOBase*

class omicidx.geometa.**GEOEnitity**(*d*)

Bases: *omicidx.geometa.GEOBase*

as_dict()

Return object as a dict

class omicidx.geometa.**GEOPlatform**(*d*)

Bases: *omicidx.geometa.GEOEnitity*

class omicidx.geometa.**GEOSample**(*d*)

Bases: *omicidx.geometa.GEOEnitity*

as_dict()

Return object as a dict

class omicidx.geometa.**GEOSeries**(*d*)

Bases: *omicidx.geometa.GEOEnitity*

omicidx.geometa.**geo_soft_entity_iterator**(*fh*)

Returns an iterator of GEO entities

Given a GEO accession (typically a GSE, will return an iterator of the GEO entities associated with the record, including all GSMs, GPLs, and the GSE record itself

Parameters **fh** (*anything that can iterate over lines of text*) – Could be a list of text lines, a file handle, or an open url.

Yields *Iterator of GEO entities*

```
>>> for i in geo_soft_entity_iterator(get_geo_accession_soft('GSE2553')):
...     print(i)
```

omicidx.geometa.**geo_soft_iterator**(*self, txt*)

omicidx.geometa.**get_SRA_from_relations**(*relation_list*)

omicidx.geometa.**get_bioprojects_from_relations**(*relation_list*)

omicidx.geometa.**get_biosample_from_relations**(*relation_list*)

omicidx.geometa.**get_entrez_instance**(*email='user@example.com'*)

Return a Bio::Entrez object

# Arguments email (str): the email to be used with the Entrez instance

Returns

Return type A Bio::Entrez instance

omicidx.geometa.**get_geo_accession_soft**(*accession, targ='all'*)

Open a connection to get the GEO SOFT for an accession

Parameters

- **accession** (*str the GEO accesssion*) –
- **targ** (*str what to fetch. One of "all", "series", "platform",*) – "samples", "self"

Returns

- *A file-like object for reading or readlines*
- *>>> handle = get_geo_accession_soft('GSE2553')*
- *>>> handle.readlines()*

omicidx.geometa.**get_geo_accession_xml**(*accession*)

omicidx.geometa.**get_geo_accessions**(*etyp='GSE', batch_size=1000, add_term=None, email='user@example.com'*)

get GEO accessions

Useful for getting all the ETYP accessions for later bulk processing

Parameters

- **etyp** (*str*) – One of GSE, GPL, GSM, GDS
- **batch_size** (*int*) – the number of accessions to return in one batch. Transparent to the user, as this returns an iterator.
- **add_term** (*str*) – Add a search term for the query. Useful to limit by date or search for specific text.
- **email** (*str*) – user email (not important)

Returns

Return type an iterator of accessions, each as a string

omicidx.geometa.**get_geo_entities**(*txt*)

Get the text associated with each entity from a block of text

Parameters **txt** (*list(str)*) –

Returns

Return type A dict of entities keyed by accession and values a list of txt lines

omicidx.geometa.**get_subseries_from_relations**(*relation_list*)

omicidx.lambda_handlers module

omicidx.lambda_handlers.**create_dynamodb_table**(*tname*)

omicidx.lambda_handlers.**full_experiment_package**(*accession*)

omicidx.lambda_handlers.**insert_if_empty**(*accession*)

omicidx.lambda_handlers.**insert_to_dynamo**(*expt_pkg*)

omicidx.lambda_handlers.**lambda_insert_from_sqs_to_dynamo**(*event*, *context*)

omicidx.lambda_handlers.**lambda_insert_to_dynamo**(*event*, *context*)

omicidx.lambda_handlers.**lambda_return_full_experiment_json**(*event*, *context*)

omicidx.lambda_handlers.**main**()

omicidx.lambda_handlers.**replace_decimals**(*obj*)

omicidx.lambda_handlers.**replace_floats**(*obj*)

omicidx.models module

class omicidx.models.**AttributesMixin**
Bases: object

tag = Column(None, Text(), table=None)

value = Column(None, Text(), table=None)

class omicidx.models.**Biosample**(***kwargs*)
Bases: sqlalchemy.ext.declarative.api.Base

access

attributes

dbgap

description

id

last_update

model

publication_date

submission_date

taxon_id

taxonomy_name

title

class omicidx.models.**BiosampleAttribute**(***kwargs*)
Bases: sqlalchemy.ext.declarative.api.Base

biosamples

display_name

harmonized_name

id

name

value

class `omicidx.models.`**`BiosampleIdentifier`**(***kwargs*)

Bases: `sqlalchemy.ext.declarative.api.Base`

biosample

db

id

identifier

label

class `omicidx.models.`**`BiosampleModel`**(***kwargs*)

Bases: `sqlalchemy.ext.declarative.api.Base`

id

name

class `omicidx.models.`**`GEOContact`**(***kwargs*)

Bases: `sqlalchemy.ext.declarative.api.Base`

address

city

country

email

fax

institute

name

phone

postal_code

web_link

class `omicidx.models.`**`GEOPlatform`**(***kwargs*)

Bases: `sqlalchemy.ext.declarative.api.Base`

accession

contributors

data_row_count

description

distribution

gses

gsms

last_update_date

status

submission_date

summary

technology

title

class omicidx.models.**GEOSeries**(***kwargs*)
Bases: `sqlalchemy.ext.declarative.api.Base`

accession

contributors

data_processing

description

gpls

gsms

last_update_date

overall_design

status

submission_date

summary

title

class omicidx.models.**IdentifiersMixin**
Bases: `object`

identifier = Column(None, Text(), table=None)

namespace = Column(None, Text(), table=None)

class omicidx.models.**SraAccession**(***kwargs*)
Bases: `sqlalchemy.ext.declarative.api.Base`

accession

alias

bases

bio_project

bio_sample

center

experiment

loaded

md5sum

published

received

replaced_by

sample

spots

status

study

submission

type

updated

visibility

class omicidx.models.**SraExperiment**(***kwargs*)

Bases: `sqlalchemy.ext.declarative.api.Base`

accession

alias

attributes

broker_name

center_name

description

design

identifiers

instrument_model

library_construction_protocol

library_layout

library_layout_length

library_layout_orientation

library_layout_sdev

library_name

library_selection

library_source

library_strategy

platform

published

received

replaced_by

sample_accession

sra_sample

sra_study

status

study_accession

title

updated

visibility

xrefs

class omicidx.models.**SraLibraryLayout**(***kwargs*)
Bases: `sqlalchemy.ext.declarative.api.Base`

value

class omicidx.models.**SraLibrarySelection**(***kwargs*)
Bases: `sqlalchemy.ext.declarative.api.Base`

value

class omicidx.models.**SraLibrarySource**(***kwargs*)
Bases: `sqlalchemy.ext.declarative.api.Base`

value

class omicidx.models.**SraLibraryStrategy**(***kwargs*)
Bases: `sqlalchemy.ext.declarative.api.Base`

value

class omicidx.models.**SraPlatform**(***kwargs*)
Bases: `sqlalchemy.ext.declarative.api.Base`

value

class omicidx.models.**SraRun**(***kwargs*)
Bases: `sqlalchemy.ext.declarative.api.Base`

accession

alias

attributes

bases

bio_project

broker_name

center_name

experiment_accession

identifiers

loaded

nreads

published

reads

received

replaced_by

run_center

run_date

sample_accession

spot_length

spots

sra_experiment

sra_sample

sra_study

status

study_accession

updated

visibility

class omicidx.models.**SraSample**(***kwargs*)

Bases: `sqlalchemy.ext.declarative.api.Base`

accession

alias

attributes

bio_sample

broker_name

center_name

description

gsm

identifiers

organism

published

received

replaced_by

sra_study

status

study_accession

taxon_id

title

updated

visibility

xrefs

class omicidx.models.**SraStudy**(***kwargs*)
Bases: `sqlalchemy.ext.declarative.api.Base`

abstract

accession

alias

attributes

bio_project

bioproject

broker_name

center_name

description

gse

identifiers

published

received

replaced_by

status

study_type

title

updated

visibility

xrefs

class omicidx.models.**XrefsMixin**
Bases: `object`

db = Column(None, Text(), table=None)

identifier = Column(None, Text(), table=None)

uuid = Column(None, Text(), table=None)

omicidx.schema_tools module

class omicidx.schema_tools.**BaseType**(*name*, *nullable*)
Bases: `object`

class omicidx.schema_tools.**BooleanType**(*name*, *nullable*)
Bases: *omicidx.schema_tools.BaseType*

mapping()

class omicidx.schema_tools.**DateType**(*name*, *nullable*)
Bases: *omicidx.schema_tools.BaseType*

mapping()

class omicidx.schema_tools.**IntType**(*name*, *nullable*)
Bases: *omicidx.schema_tools.BaseType*

mapping()

class omicidx.schema_tools.**LongType**(*name*, *nullable*)
Bases: *omicidx.schema_tools.BaseType*

mapping()

class omicidx.schema_tools.**NestedType**(*name*, *nullable*)
Bases: *omicidx.schema_tools.BaseType*

mapping()

class omicidx.schema_tools.**ObjectType**(*name*, *nullable*)
Bases: *omicidx.schema_tools.BaseType*

mapping()

class omicidx.schema_tools.**StringType**(*name*, *nullable*)
Bases: *omicidx.schema_tools.BaseType*

mapping()

omicidx.schema_tools.**walk_schema**(*schema*, *parent=None*)
Walk a pyspark "schema.jsonValue" schema

This function converts a pyspark schema into a set of classes that can then convert the schema to an elasticsearch mapping.

Parameters

- **schema** (*a dict*) – Typically, this would come from *df.schema.jsonValue()*. It could also be loaded from a string or file using the *json* library.
- **parent** (a container class, so an *ObjectType* or a *NestedType*) – If empty, will be initialized with an *ObjectType* with name = "root".
- **s = walk_schema(df.schema.jsonValue(), ObjectType(name='root'))** (>>>) –
- **s.mapping()** (>>>) –

omicidx.scratch module

omicidx.scratch.**dumps**(*d*)

omicidx.scratch.**main**()

omicidx.scratch.**month_range**(*year*, *month*)

omicidx.scratch.**myconverter**(*o*)

omicidx.sra_parsers module

SRA parsers including:

- study
- sample
- experiment

- run

These parsers each parse XML format files of the format available from the fullxml api.

The main entry point into this module is the *parse_xml_file* function.

class omicidx.sra_parsers.**LiveList**(*from_date='2004-01-01'*, *to_date=None*, *count=2500*, *offset=0*, *entity='EXPERIMENT'*, *status='live'*, *max_retries=5*)

Bases: `collections.abc.Iterable`, `typing.Generic`

Return an iterator of pydantic_models

class omicidx.sra_parsers.**SRAExperimentPackage**(*node*)

Bases: `object`

expanded_experiment()

experiment()

nested_runs()

runs()

sample()

study()

class omicidx.sra_parsers.**SRAExperimentRecord**(*xml*)

Bases: *omicidx.sra_parsers.SRAXMLRecord*

parse_xml()

Parse an SRA xml EXPERIMENT element

class omicidx.sra_parsers.**SRARunRecord**(*xml*)

Bases: *omicidx.sra_parsers.SRAXMLRecord*

parse_xml()

class omicidx.sra_parsers.**SRASampleRecord**(*xml*)

Bases: *omicidx.sra_parsers.SRAXMLRecord*

parse_xml()

class omicidx.sra_parsers.**SRAStudyRecord**(*xml*)

Bases: *omicidx.sra_parsers.SRAXMLRecord*

parse_xml()

class omicidx.sra_parsers.**SRASubmissionRecord**(*xml*)

Bases: *omicidx.sra_parsers.SRAXMLRecord*

parse_xml()

class omicidx.sra_parsers.**SRAXMLRecord**(*xml*)

Bases: `object`

parse_xml()

omicidx.sra_parsers.**dict_from_single_xml**(*txt*)

omicidx.sra_parsers.**get_accession_list**(*from_date='2001-01-01'*, *to_date='2050-01-01'*, *count=100*, *offset=0*, *type='EXPERIMENT'*)

Get SRA accessions from SRA API

The API is at https://www.ncbi.nlm.nih.gov/Traces/sra/status/srastatrep.fcgi/acc-mirroring and can be used to get “RUN”, “EXPERIMENT”, “SAMPLE”, and “STUDY”.

Parameters

- **`from_date`** (*`str`*) – Collect accessions starting from the given date
- **`to_date`** (*`str`*) – Collect accessions starting from the given date
- **`count`** (*`int`*) – How many accessions to collect from the SRA API in one pass
- **`offset`** (*`int`*) – Start at offset…
- **`type`** (*`str`*) – One of "RUN", "EXPERIMENT", "SAMPLE", and "STUDY".

Returns

Return type An iterator of accessions as strings.

`omicidx.sra_parsers.`**`lambda_handler`**(*event*, *context*)

`omicidx.sra_parsers.`**`load_experiment_xml_by_accession`**(*accession*)

`omicidx.sra_parsers.`**`load_runbrowser_xml_by_accession`**(*accession*)

`omicidx.sra_parsers.`**`model_from_single_xml`**(*txt*)

`omicidx.sra_parsers.`**`models_from_runbrowser`**(*accession*)

`omicidx.sra_parsers.`**`open_file`**(*fname*, *encoding='UTF-8'*)
Open a file, generalized to deal with gzip files

Parameters **`fname`** (*`string`*) – The filename. If ends in '.gz', is opened with gzip.

Returns

Return type an open file handle

`omicidx.sra_parsers.`**`parse_addons_info`**(*fname*)

`omicidx.sra_parsers.`**`parse_experiment`**(*xml*)
Parse an SRA xml EXPERIMENT element

Parameters **`xml`** (*`xml.etree.ElementTree.Element`*) –

Returns

Return type a dict of experiment

`omicidx.sra_parsers.`**`parse_livelist`**(*fname*)

`omicidx.sra_parsers.`**`parse_run`**(*xml*)
Parse an SRA xml RUN element

Parameters **`xml`** (*`an xml.etree Element`*) –

Returns

Return type A dict object parsed from the XML

`omicidx.sra_parsers.`**`parse_run_info`**(*fname*)

`omicidx.sra_parsers.`**`parse_sample`**(*xml*)
Parse an SRA xml SAMPLE element

Parameters **`xml`** (*`an xml.etree Element`*) –

Returns

Return type A dict object parsed from the XML

`omicidx.sra_parsers.`**`parse_study`**(*xml*)
Parse an SRA xml STUDY element

Parameters **xml** (*an xml.etree Element*) –

Returns

Return type A dict object parsed from the XML

`omicidx.sra_parsers.`**`parse_submission`**(*xml*)

Parse an SRA xml SUBMISSION element

Parameters **xml** (*xml.etree.ElementTree.Element*) –

Returns

Return type a dict of experiment

`omicidx.sra_parsers.`**`parse_xml_file`**(*xmlfilename*)

Parse an NCBI SRA mirroring XML file

This function returns an iterator over the records in the xml file, returning a dict of parsed records.

For example:

wget –mirror -nH –cut-dirs=3 ftp://ftp.ncbi.nlm.nih.gov/sra/reports/Mirroring/NCBI_SRA_Mirroring_20181027/

```
>>> import omicidx.sra_parsers as sp
>>> studies = sp.parse_xml_file("NCBI_SRA_Mirroring_20181027/meta_study_set.xml.gz
↪")
>>> next(studies)
...
```

Parameters **xmlfilename** (*string*) – the filename to be parsed. Can be gzipped. Must include the “entity” name in the filename (eg., “run”, “experiment”)

Returns An iterator of dict records from parsing each xml record.

Return type iterator

`omicidx.sra_parsers.`**`results_from_runbrowser`**(*accession*)

Return complete record from runbrowser

Parameters **accession** (*string*) – Any SRA accession, but the only one-to-one results are for SRR records.

Returns

Return type a dict with experiment, run, study, and sample records

`omicidx.sra_parsers.`**`run_from_runbrowser`**(*accession*)

Just the run part of runbrowser output

`omicidx.sra_parsers.`**`run_iterator`**(*from_date='2001-01-01', to_date='2050-01-01', count=100, offset=0*)

`omicidx.sra_parsers.`**`sra_object_generator`**(*fname*)

Iterate over objects in an SRA meta_XXX_set xml file

Param fname str() name of xml file, may be gzipped or not

Returns An iterator over SRA objects. Access actual data as a dict using Object.data

`omicidx.sra_parsers.`**`srastatrep`**(*accessions*)

Access the SRA statrep api

Parameters **accessions** (*List[str]*) – A list (or can be single string) of accessions

Returns

Return type A named tuple with statrep column names as field_names

`omicidx.sra_parsers.`**`try_update`**(*d*, *value*)

omicidx.utils module

`omicidx.utils.`**`open_file`**(*fname*, *mode='rt'*)

open a file, including dealing with gzipped files

Parameters

- **`fname`** (`str`) – a filename. If ending in .gz, will use gzip.open(). Otherwise, a regular open() will be used.
- **`mode`** (`str`) – The file opening mode.

Module contents

Hello there

1.5 Ipython notebooks

- intro

1.6 Rmarkdown workbooks

- intro

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

O

omicidx, 34
omicidx.bigquery_utils, 18
omicidx.biosample, 19
omicidx.data, 7
omicidx.data.bigquery_schemas, 7
omicidx.db, 20
omicidx.elasticsearch_utils, 20
omicidx.gcs_utils, 20
omicidx.geometa, 21
omicidx.lambda_handlers, 23
omicidx.model, 16
omicidx.model.etl, 7
omicidx.model.etl_schema, 9
omicidx.model.public, 10
omicidx.models, 23
omicidx.schema, 16
omicidx.schema_tools, 29
omicidx.scratch, 30
omicidx.scripts, 16
omicidx.scripts.cli, 16
omicidx.scripts.omicidx, 16
omicidx.scripts.sra_entity_to_json, 16
omicidx.sra, 18
omicidx.sra.ebiutils, 16
omicidx.sra.etl, 17
omicidx.sra.pydantic_models, 18
omicidx.sra_parsers, 30
omicidx.utils, 34

INDEX

A

abstract (*omicidx.model.etl_schema.SRAStudy attribute*), 9
abstract (*omicidx.model.public.SraStudy attribute*), 14
abstract (*omicidx.models.SraStudy attribute*), 29
access (*omicidx.model.public.Biosample attribute*), 10
access (*omicidx.models.Biosample attribute*), 23
accession (*omicidx.model.etl.SraAccession attribute*), 7
accession (*omicidx.model.etl_schema.SRAExperiment attribute*), 9
accession (*omicidx.model.etl_schema.SRAStudy attribute*), 9
accession (*omicidx.model.public.Biosample attribute*), 10
accession (*omicidx.model.public.SraExperiment attribute*), 11
accession (*omicidx.model.public.SraExperimentIdentifier attribute*), 12
accession (*omicidx.model.public.SraRun attribute*), 12
accession (*omicidx.model.public.SraSample attribute*), 13
accession (*omicidx.model.public.SraSampleIdentifier attribute*), 14
accession (*omicidx.model.public.SraSampleXref attribute*), 14
accession (*omicidx.model.public.SraStudy attribute*), 14
accession (*omicidx.model.public.SraStudyIdentifier attribute*), 15
accession (*omicidx.model.public.SraStudyXref attribute*), 15
accession (*omicidx.models.GEOPlatform attribute*), 24
accession (*omicidx.models.GEOSeries attribute*), 25
accession (*omicidx.models.SraAccession attribute*), 25
accession (*omicidx.models.SraExperiment attribute*), 26
accession (*omicidx.models.SraRun attribute*), 27
accession (*omicidx.models.SraSample attribute*), 28
accession (*omicidx.models.SraStudy attribute*), 29
address (*omicidx.models.GEOContact attribute*), 24
alias (*omicidx.model.etl.SraAccession attribute*), 7
alias (*omicidx.model.etl_schema.SRAStudy attribute*), 9
alias (*omicidx.model.public.SraExperiment attribute*), 11
alias (*omicidx.model.public.SraRun attribute*), 12
alias (*omicidx.model.public.SraSample attribute*), 13
alias (*omicidx.model.public.SraStudy attribute*), 14
alias (*omicidx.models.SraAccession attribute*), 25
alias (*omicidx.models.SraExperiment attribute*), 26
alias (*omicidx.models.SraRun attribute*), 27
alias (*omicidx.models.SraSample attribute*), 28
alias (*omicidx.models.SraStudy attribute*), 29
as_dict() (*omicidx.geometa.GEOBase method*), 21
as_dict() (*omicidx.geometa.GEOEnitity method*), 21
as_dict() (*omicidx.geometa.GEOSample method*), 21
as_json() (*omicidx.biosample.BioSample method*), 20
Attribute (*class in omicidx.sra.pydantic_models*), 18
attributes (*omicidx.model.etl_schema.SRAStudy attribute*), 9
attributes (*omicidx.model.public.Biosample attribute*), 10
attributes (*omicidx.model.public.SraExperiment attribute*), 11
attributes (*omicidx.model.public.SraRun attribute*), 12
attributes (*omicidx.model.public.SraSample attribute*), 13
attributes (*omicidx.model.public.SraStudy attribute*), 14
attributes (*omicidx.models.Biosample attribute*), 23
attributes (*omicidx.models.SraExperiment attribute*), 26
attributes (*omicidx.models.SraRun attribute*), 27
attributes (*omicidx.models.SraSample attribute*), 28
attributes (*omicidx.models.SraStudy attribute*), 29
AttributesMixin (*class in omicidx.model.public*), 10
AttributesMixin (*class in omicidx.models*), 23

B

BaseCounts (*class in omicidx.sra.pydantic_models*), 18
BaseQualities (*class in omicidx.sra.pydantic_models*), 18
bases (*omicidx.model.etl.SraAccession attribute*), 7
bases (*omicidx.model.public.SraRun attribute*), 12
bases (*omicidx.models.SraAccession attribute*), 25
bases (*omicidx.models.SraRun attribute*), 27
BaseType (*class in omicidx.schema_tools*), 29
bio_project (*omicidx.model.etl.SraAccession attribute*), 7
bio_project (*omicidx.model.public.SraRun attribute*), 12
bio_project (*omicidx.models.SraAccession attribute*), 25
bio_project (*omicidx.models.SraRun attribute*), 27
bio_project (*omicidx.models.SraStudy attribute*), 29
bio_sample (*omicidx.model.etl.SraAccession attribute*), 7
bio_sample (*omicidx.model.public.SraSample attribute*), 13
bio_sample (*omicidx.models.SraAccession attribute*), 25
bio_sample (*omicidx.models.SraSample attribute*), 28
bioproject (*omicidx.model.etl_schema.SRAStudy attribute*), 9
bioproject (*omicidx.model.public.SraStudy attribute*), 14
bioproject (*omicidx.models.SraStudy attribute*), 29
BioSample (*class in omicidx.biosample*), 20
Biosample (*class in omicidx.model.public*), 10
Biosample (*class in omicidx.models*), 23
biosample (*omicidx.models.BiosampleIdentifier attribute*), 24
BIOSAMPLE_FILE
    omicidx-cli-biosample-parse command line option, 5
biosample_to_json() (*in module omicidx.scripts.cli*), 16
BiosampleAttribute (*class in omicidx.models*), 23
BiosampleIdentifier (*class in omicidx.models*), 24
BiosampleModel (*class in omicidx.models*), 24
BioSampleParser (*class in omicidx.biosample*), 20
biosamples (*omicidx.models.BiosampleAttribute attribute*), 23
BooleanType (*class in omicidx.schema_tools*), 29
broker_name (*omicidx.model.etl_schema.SRAStudy attribute*), 9
broker_name (*omicidx.model.public.SraExperiment attribute*), 11
broker_name (*omicidx.model.public.SraRun attribute*), 12
broker_name (*omicidx.model.public.SraSample attribute*), 13
broker_name (*omicidx.model.public.SraStudy attribute*), 14
broker_name (*omicidx.models.SraExperiment attribute*), 26
broker_name (*omicidx.models.SraRun attribute*), 27
broker_name (*omicidx.models.SraSample attribute*), 28
broker_name (*omicidx.models.SraStudy attribute*), 29
bulk_index() (*in module omicidx.elasticsearch_utils*), 20
bulk_index_from_gcs() (*in module omicidx.elasticsearch_utils*), 20

C

center (*omicidx.model.etl.SraAccession attribute*), 7
center (*omicidx.models.SraAccession attribute*), 25
center_name (*omicidx.model.etl_schema.SRAStudy attribute*), 9
center_name (*omicidx.model.public.SraExperiment attribute*), 11
center_name (*omicidx.model.public.SraRun attribute*), 12
center_name (*omicidx.model.public.SraSample attribute*), 13
center_name (*omicidx.model.public.SraStudy attribute*), 14
center_name (*omicidx.models.SraExperiment attribute*), 26
center_name (*omicidx.models.SraRun attribute*), 27
center_name (*omicidx.models.SraSample attribute*), 28
center_name (*omicidx.models.SraStudy attribute*), 29
city (*omicidx.models.GEOContact attribute*), 24
consume() (*in module omicidx.scripts.omicidx*), 16
contributors (*omicidx.models.GEOPlatform attribute*), 24
contributors (*omicidx.models.GEOSeries attribute*), 25
copy_table() (*in module omicidx.bigquery_utils*), 18
country (*omicidx.models.GEOContact attribute*), 24
create_all_in_schema() (*in module omicidx.model.public*), 15
create_all_tables() (*in module omicidx.sra.etl*), 17
create_dynamodb_table() (*in module omicidx.lambda_handlers*), 23
create_table() (*in module omicidx.sra.etl*), 17
create_table() (*omicidx.db.OmicIDXDb method*), 20

D

data_processing (*omicidx.models.GEOSeries at-*

tribute), 25
data_row_count (*omicidx.models.GEOPlatform attribute*), 24
dateconverter() (*in module omicidx.scripts.cli*), 16
dateconverter() (*in module omicidx.scripts.omicidx*), 16
DateType (*class in omicidx.schema_tools*), 29
db (*omicidx.model.public.SraSampleXref attribute*), 14
db (*omicidx.model.public.SraStudyXref attribute*), 15
db (*omicidx.model.public.XrefsMixin attribute*), 15
db (*omicidx.models.BiosampleIdentifier attribute*), 24
db (*omicidx.models.XrefsMixin attribute*), 29
dbgap (*omicidx.model.public.Biosample attribute*), 10
dbgap (*omicidx.models.Biosample attribute*), 23
delete_table() (*in module omicidx.sra.etl*), 17
description (*omicidx.model.etl_schema.SRAStudy attribute*), 9
description (*omicidx.model.public.Biosample attribute*), 10
description (*omicidx.model.public.SraExperiment attribute*), 11
description (*omicidx.model.public.SraSample attribute*), 13
description (*omicidx.model.public.SraStudy attribute*), 14
description (*omicidx.models.Biosample attribute*), 23
description (*omicidx.models.GEOPlatform attribute*), 24
description (*omicidx.models.GEOSeries attribute*), 25
description (*omicidx.models.SraExperiment attribute*), 26
description (*omicidx.models.SraSample attribute*), 28
description (*omicidx.models.SraStudy attribute*), 29
design (*omicidx.model.public.SraExperiment attribute*), 11
design (*omicidx.models.SraExperiment attribute*), 26
dict_from_single_xml() (*in module omicidx.sra_parsers*), 31
display_name (*omicidx.models.BiosampleAttribute attribute*), 24
distribution (*omicidx.models.GEOPlatform attribute*), 24
do_all_join_tables() (*in module omicidx.sra.etl*), 17
doc (*omicidx.model.etl_schema.ExperimentJson attribute*), 9
doc (*omicidx.model.etl_schema.RunJson attribute*), 9
doc (*omicidx.model.etl_schema.SampleJson attribute*), 10
doc (*omicidx.model.etl_schema.StudyJson attribute*), 10
download_biosample() (*in module omicidx.scripts.cli*), 16
dumps() (*in module omicidx.scratch*), 30

E

EBIEna (*class in omicidx.sra.ebiutils*), 16
email (*omicidx.models.GEOContact attribute*), 24
ENTITY
  omicidx-cli-sra-parse-entity command line option, 6
expanded_experiment() (*omicidx.sra_parsers.SRAExperimentPackage method*), 31
experiment (*omicidx.model.etl.SraAccession attribute*), 7
experiment (*omicidx.models.SraAccession attribute*), 25
experiment() (*omicidx.sra_parsers.SRAExperimentPackage method*), 31
experiment_accession (*omicidx.model.public.SraRun attribute*), 12
experiment_accession (*omicidx.models.SraRun attribute*), 27
experiment_join_accessions() (*in module omicidx.sra.etl*), 17
ExperimentJson (*class in omicidx.model.etl_schema*), 9

F

fax (*omicidx.models.GEOContact attribute*), 24
FileAlternative (*class in omicidx.sra.pydantic_models*), 18
FileSet (*class in omicidx.sra.pydantic_models*), 18
full_experiment_package() (*in module omicidx.lambda_handlers*), 23
FullSraRun (*class in omicidx.sra.pydantic_models*), 18

G

geo_soft_entity_iterator() (*in module omicidx.geometa*), 21
geo_soft_iterator() (*in module omicidx.geometa*), 22
GEOBase (*class in omicidx.geometa*), 21
GEOChannel (*class in omicidx.geometa*), 21
GEOContact (*class in omicidx.geometa*), 21
GEOContact (*class in omicidx.models*), 24
GEOEnitity (*class in omicidx.geometa*), 21
GEOPlatform (*class in omicidx.geometa*), 21
GEOPlatform (*class in omicidx.models*), 24
GEOSample (*class in omicidx.geometa*), 21
GEOSeries (*class in omicidx.geometa*), 21
GEOSeries (*class in omicidx.models*), 25

get_accession_list() (*in module omicidx.sra_parsers*), 31
get_bioprojects_from_relations() (*in module omicidx.geometa*), 22
get_biosample_from_relations() (*in module omicidx.geometa*), 22
get_conn() (*omicidx.db.OmicIDXDb method*), 20
get_entrez_instance() (*in module omicidx.geometa*), 22
get_file_report_for_accession() (*omicidx.sra.ebiutils.EBIEna method*), 16
get_geo_accession_soft() (*in module omicidx.geometa*), 22
get_geo_accession_xml() (*in module omicidx.geometa*), 22
get_geo_accessions() (*in module omicidx.geometa*), 22
get_geo_entities() (*in module omicidx.geometa*), 22
get_parsed_xml_for_accession() (*omicidx.sra.ebiutils.EBIEna method*), 17
get_SRA_from_relations() (*in module omicidx.geometa*), 22
get_subseries_from_relations() (*in module omicidx.geometa*), 23
get_xml_for_accession() (*omicidx.sra.ebiutils.EBIEna method*), 17
gpls (*omicidx.models.GEOSeries attribute*), 25
gse (*omicidx.model.public.SraStudy attribute*), 14
gse (*omicidx.models.SraStudy attribute*), 29
gse_accession (*omicidx.model.etl_schema.SRAStudy attribute*), 9
gses (*omicidx.models.GEOPlatform attribute*), 24
gsm (*omicidx.model.public.Biosample attribute*), 10
gsm (*omicidx.model.public.SraSample attribute*), 13
gsm (*omicidx.models.SraSample attribute*), 28
gsms (*omicidx.models.GEOPlatform attribute*), 24
gsms (*omicidx.models.GEOSeries attribute*), 25

H

harmonized_name (*omicidx.models.BiosampleAttribute attribute*), 24

I

id (*omicidx.model.etl_schema.ExperimentJson attribute*), 9
id (*omicidx.model.etl_schema.RunJson attribute*), 9
id (*omicidx.model.etl_schema.SampleJson attribute*), 10
id (*omicidx.model.etl_schema.StudyJson attribute*), 10
id (*omicidx.model.public.AttributesMixin attribute*), 10
id (*omicidx.model.public.IdentifiersMixin attribute*), 11
id (*omicidx.model.public.SraExperimentAttribute attribute*), 12
id (*omicidx.model.public.SraExperimentIdentifier attribute*), 12
id (*omicidx.model.public.SraRunAttribute attribute*), 13
id (*omicidx.model.public.SraSampleAttribute attribute*), 14
id (*omicidx.model.public.SraSampleIdentifier attribute*), 14
id (*omicidx.model.public.SraSampleXref attribute*), 14
id (*omicidx.model.public.SraStudyAttribute attribute*), 15
id (*omicidx.model.public.SraStudyIdentifier attribute*), 15
id (*omicidx.model.public.SraStudyXref attribute*), 15
id (*omicidx.model.public.XrefsMixin attribute*), 15
id (*omicidx.models.Biosample attribute*), 23
id (*omicidx.models.BiosampleAttribute attribute*), 24
id (*omicidx.models.BiosampleIdentifier attribute*), 24
id (*omicidx.models.BiosampleModel attribute*), 24
Identifier (*class in omicidx.sra.pydantic_models*), 18
identifier (*omicidx.model.public.IdentifiersMixin attribute*), 11
identifier (*omicidx.model.public.SraExperimentIdentifier attribute*), 12
identifier (*omicidx.model.public.SraSampleIdentifier attribute*), 14
identifier (*omicidx.model.public.SraSampleXref attribute*), 14
identifier (*omicidx.model.public.SraStudyIdentifier attribute*), 15
identifier (*omicidx.model.public.SraStudyXref attribute*), 15
identifier (*omicidx.model.public.XrefsMixin attribute*), 15
identifier (*omicidx.models.BiosampleIdentifier attribute*), 24
identifier (*omicidx.models.IdentifiersMixin attribute*), 25
identifier (*omicidx.models.XrefsMixin attribute*), 29
identifiers (*omicidx.model.etl_schema.SRAStudy attribute*), 9
identifiers (*omicidx.model.public.Biosample attribute*), 10
identifiers (*omicidx.model.public.SraExperiment attribute*), 11
identifiers (*omicidx.model.public.SraRun attribute*), 13
identifiers (*omicidx.model.public.SraSample attribute*), 13
identifiers (*omicidx.model.public.SraStudy attribute*), 14
identifiers (*omicidx.models.SraExperiment at-*

tribute), 26
identifiers (*omicidx.models.SraRun attribute*), 27
identifiers (*omicidx.models.SraSample attribute*), 28
identifiers (*omicidx.models.SraStudy attribute*), 29
IdentifiersMixin (*class in omicidx.model.public*), 11
IdentifiersMixin (*class in omicidx.models*), 25
insert_if_empty() (*in module omicidx.lambda_handlers*), 23
insert_to_dynamo() (*in module omicidx.lambda_handlers*), 23
institute (*omicidx.models.GEOContact attribute*), 24
instrument_model (*omicidx.model.public.SraExperiment attribute*), 11
instrument_model (*omicidx.models.SraExperiment attribute*), 26
IntType (*class in omicidx.schema_tools*), 29
is_reference (*omicidx.model.public.Biosample attribute*), 10

L

label (*omicidx.models.BiosampleIdentifier attribute*), 24
lambda_handler() (*in module omicidx.sra_parsers*), 32
lambda_insert_from_sqs_to_dynamo() (*in module omicidx.lambda_handlers*), 23
lambda_insert_to_dynamo() (*in module omicidx.lambda_handlers*), 23
lambda_return_full_experiment_json() (*in module omicidx.lambda_handlers*), 23
last_update (*omicidx.model.public.Biosample attribute*), 10
last_update (*omicidx.models.Biosample attribute*), 23
last_update_date (*omicidx.models.GEOPlatform attribute*), 25
last_update_date (*omicidx.models.GEOSeries attribute*), 25
library_construction_protocol (*omicidx.model.public.SraExperiment attribute*), 11
library_construction_protocol (*omicidx.models.SraExperiment attribute*), 26
library_layout (*omicidx.model.public.SraExperiment attribute*), 11
library_layout (*omicidx.models.SraExperiment attribute*), 26
library_layout_length (*omicidx.model.public.SraExperiment attribute*), 11
library_layout_length (*omicidx.models.SraExperiment attribute*), 26
library_layout_orientation (*omicidx.model.public.SraExperiment attribute*), 11
library_layout_orientation (*omicidx.models.SraExperiment attribute*), 26
library_layout_sdev (*omicidx.model.public.SraExperiment attribute*), 11
library_layout_sdev (*omicidx.models.SraExperiment attribute*), 26
library_name (*omicidx.model.public.SraExperiment attribute*), 11
library_name (*omicidx.models.SraExperiment attribute*), 26
library_selection (*omicidx.model.public.SraExperiment attribute*), 11
library_selection (*omicidx.models.SraExperiment attribute*), 26
library_source (*omicidx.model.public.SraExperiment attribute*), 11
library_source (*omicidx.models.SraExperiment attribute*), 26
library_strategy (*omicidx.model.public.SraExperiment attribute*), 11
library_strategy (*omicidx.models.SraExperiment attribute*), 26
list_blobs() (*in module omicidx.gcs_utils*), 20
LiveList (*class in omicidx.sra.pydantic_models*), 18
LiveList (*class in omicidx.sra_parsers*), 31
load_accession_table() (*in module omicidx.sra.etl*), 17
load_all_json_tables() (*in module omicidx.sra.etl*), 17
load_biosample_from_gcs_to_bigquery() (*in module omicidx.scripts.cli*), 16
load_csv_to_bigquery() (*in module omicidx.bigquery_utils*), 18
load_experiment_xml_by_accession() (*in module omicidx.sra_parsers*), 32
load_json_table() (*in module omicidx.sra.etl*), 17
load_json_to_bigquery() (*in module omicidx.bigquery_utils*), 19
load_livelist_table() (*in module omicidx.sra.etl*), 17
load_runbrowser_xml_by_accession() (*in module omicidx.sra_parsers*), 32
loaded (*omicidx.model.etl.SraAccession attribute*), 7
loaded (*omicidx.model.public.SraRun attribute*), 13

loaded (*omicidx.models.SraAccession attribute*), 25
loaded (*omicidx.models.SraRun attribute*), 27
LongType (*class in omicidx.schema_tools*), 30

M

main() (*in module omicidx.lambda_handlers*), 23
main() (*in module omicidx.model.etl_schema*), 10
main() (*in module omicidx.scratch*), 30
main() (*in module omicidx.scripts.sra_entity_to_json*), 16
mapping() (*omicidx.schema_tools.BooleanType method*), 29
mapping() (*omicidx.schema_tools.DateType method*), 29
mapping() (*omicidx.schema_tools.IntType method*), 30
mapping() (*omicidx.schema_tools.LongType method*), 30
mapping() (*omicidx.schema_tools.NestedType method*), 30
mapping() (*omicidx.schema_tools.ObjectType method*), 30
mapping() (*omicidx.schema_tools.StringType method*), 30
md5sum (*omicidx.model.etl.SraAccession attribute*), 7
md5sum (*omicidx.models.SraAccession attribute*), 25
MIRRORDIR
  omicidx-cli-sra-download command line option, 5
  omicidx-cli-sra-upload command line option, 6
model (*omicidx.model.public.Biosample attribute*), 10
model (*omicidx.models.Biosample attribute*), 23
model_from_single_xml() (*in module omicidx.sra_parsers*), 32
models_from_runbrowser() (*in module omicidx.sra_parsers*), 32
month_range() (*in module omicidx.scratch*), 30
myconverter() (*in module omicidx.scratch*), 30

N

name (*omicidx.models.BiosampleAttribute attribute*), 24
name (*omicidx.models.BiosampleModel attribute*), 24
name (*omicidx.models.GEOContact attribute*), 24
namespace (*omicidx.model.public.IdentifiersMixin attribute*), 11
namespace (*omicidx.model.public.SraExperimentIdentifier attribute*), 12
namespace (*omicidx.model.public.SraSampleIdentifier attribute*), 14
namespace (*omicidx.model.public.SraStudyIdentifier attribute*), 15
namespace (*omicidx.models.IdentifiersMixin attribute*), 25
nested_runs() (*omicidx.sra_parsers.SRAExperimentPackage method*), 31
NestedType (*class in omicidx.schema_tools*), 30
nreads (*omicidx.model.public.SraRun attribute*), 13
nreads (*omicidx.models.SraRun attribute*), 27

O

ObjectType (*class in omicidx.schema_tools*), 30
omicidx (*module*), 34
omicidx-cli-biosample-parse command line option
  BIOSAMPLE_FILE, 5
omicidx-cli-sra-download command line option
  MIRRORDIR, 5
omicidx-cli-sra-parse-entity command line option
  ENTITY, 6
omicidx-cli-sra-upload command line option
  MIRRORDIR, 6
omicidx.bigquery_utils (*module*), 18
omicidx.biosample (*module*), 19
omicidx.data (*module*), 7
omicidx.data.bigquery_schemas (*module*), 7
omicidx.db (*module*), 20
omicidx.elasticsearch_utils (*module*), 20
omicidx.gcs_utils (*module*), 20
omicidx.geometa (*module*), 21
omicidx.lambda_handlers (*module*), 23
omicidx.model (*module*), 16
omicidx.model.etl (*module*), 7
omicidx.model.etl_schema (*module*), 9
omicidx.model.public (*module*), 10
omicidx.models (*module*), 23
omicidx.schema (*module*), 16
omicidx.schema_tools (*module*), 29
omicidx.scratch (*module*), 30
omicidx.scripts (*module*), 16
omicidx.scripts.cli (*module*), 16
omicidx.scripts.omicidx (*module*), 16
omicidx.scripts.sra_entity_to_json (*module*), 16
omicidx.sra (*module*), 18
omicidx.sra.ebiutils (*module*), 16
omicidx.sra.etl (*module*), 17
omicidx.sra.pydantic_models (*module*), 18
omicidx.sra_parsers (*module*), 30
omicidx.utils (*module*), 34
OmicIDXDb (*class in omicidx.db*), 20
open_file() (*in module omicidx.sra_parsers*), 32
open_file() (*in module omicidx.utils*), 34

organism (*omicidx.model.public.SraSample attribute*), 13
organism (*omicidx.models.SraSample attribute*), 28
overall_design (*omicidx.models.GEOSeries attribute*), 25

P

parse_addons_info() (*in module omicidx.sra_parsers*), 32
parse_bq_json_schema() (*in module omicidx.bigquery_utils*), 19
parse_experiment() (*in module omicidx.sra_parsers*), 32
parse_livelist() (*in module omicidx.sra_parsers*), 32
parse_run() (*in module omicidx.sra_parsers*), 32
parse_run_info() (*in module omicidx.sra_parsers*), 32
parse_sample() (*in module omicidx.sra_parsers*), 32
parse_study() (*in module omicidx.sra_parsers*), 32
parse_submission() (*in module omicidx.sra_parsers*), 33
parse_xml() (*omicidx.sra_parsers.SRAExperimentRecord method*), 31
parse_xml() (*omicidx.sra_parsers.SRARunRecord method*), 31
parse_xml() (*omicidx.sra_parsers.SRASampleRecord method*), 31
parse_xml() (*omicidx.sra_parsers.SRAStudyRecord method*), 31
parse_xml() (*omicidx.sra_parsers.SRASubmissionRecord method*), 31
parse_xml() (*omicidx.sra_parsers.SRAXMLRecord method*), 31
parse_xml_file() (*in module omicidx.sra_parsers*), 33
phone (*omicidx.models.GEOContact attribute*), 24
platform (*omicidx.model.public.SraExperiment attribute*), 11
platform (*omicidx.models.SraExperiment attribute*), 26
postal_code (*omicidx.models.GEOContact attribute*), 24
prep_bigquery_schemas() (*in module omicidx.sra.etl*), 17
produce() (*in module omicidx.scripts.omicidx*), 16
publication_date (*omicidx.model.public.Biosample attribute*), 10
publication_date (*omicidx.models.Biosample attribute*), 23
published (*omicidx.model.etl.SraAccession attribute*), 7
published (*omicidx.model.etl_schema.SRAStudy attribute*), 9
published (*omicidx.model.public.SraExperiment attribute*), 11
published (*omicidx.model.public.SraRun attribute*), 13
published (*omicidx.model.public.SraSample attribute*), 13
published (*omicidx.model.public.SraStudy attribute*), 15
published (*omicidx.models.SraAccession attribute*), 25
published (*omicidx.models.SraExperiment attribute*), 26
published (*omicidx.models.SraRun attribute*), 27
published (*omicidx.models.SraSample attribute*), 28
published (*omicidx.models.SraStudy attribute*), 29

Q

query() (*in module omicidx.bigquery_utils*), 19
query_to_destination() (*in module omicidx.bigquery_utils*), 19

R

reads (*omicidx.model.public.SraRun attribute*), 13
reads (*omicidx.models.SraRun attribute*), 27
received (*omicidx.model.etl.SraAccession attribute*), 7
received (*omicidx.model.etl_schema.SRAStudy attribute*), 9
received (*omicidx.model.public.SraExperiment attribute*), 11
received (*omicidx.model.public.SraRun attribute*), 13
received (*omicidx.model.public.SraSample attribute*), 13
received (*omicidx.model.public.SraStudy attribute*), 15
received (*omicidx.models.SraAccession attribute*), 25
received (*omicidx.models.SraExperiment attribute*), 26
received (*omicidx.models.SraRun attribute*), 27
received (*omicidx.models.SraSample attribute*), 28
received (*omicidx.models.SraStudy attribute*), 29
replace_decimals() (*in module omicidx.lambda_handlers*), 23
replace_floats() (*in module omicidx.lambda_handlers*), 23
replaced_by (*omicidx.model.etl.SraAccession attribute*), 7
replaced_by (*omicidx.model.etl_schema.SRAStudy attribute*), 9
replaced_by (*omicidx.model.public.SraExperiment attribute*), 11
replaced_by (*omicidx.model.public.SraRun attribute*), 13

replaced_by (*omicidx.model.public.SraSample attribute*), 13
replaced_by (*omicidx.model.public.SraStudy attribute*), 15
replaced_by (*omicidx.models.SraAccession attribute*), 26
replaced_by (*omicidx.models.SraExperiment attribute*), 26
replaced_by (*omicidx.models.SraRun attribute*), 27
replaced_by (*omicidx.models.SraSample attribute*), 28
replaced_by (*omicidx.models.SraStudy attribute*), 29
results_from_runbrowser() (*in module omicidx.sra_parsers*), 33
run() (*in module omicidx.scripts.omicidx*), 16
run_center (*omicidx.model.public.SraRun attribute*), 13
run_center (*omicidx.models.SraRun attribute*), 28
run_date (*omicidx.model.public.SraRun attribute*), 13
run_date (*omicidx.models.SraRun attribute*), 28
run_from_runbrowser() (*in module omicidx.sra_parsers*), 33
run_iterator() (*in module omicidx.sra_parsers*), 33
run_join_accessions() (*in module omicidx.sra.etl*), 17
RunJson (*class in omicidx.model.etl_schema*), 9
RunRead (*class in omicidx.sra.pydantic_models*), 18
runs() (*omicidx.sra_parsers.SRAExperimentPackage method*), 31

S

sample (*omicidx.model.etl.SraAccession attribute*), 7
sample (*omicidx.models.SraAccession attribute*), 26
sample() (*omicidx.sra_parsers.SRAExperimentPackage method*), 31
sample_accession (*omicidx.model.public.SraExperiment attribute*), 11
sample_accession (*omicidx.models.SraExperiment attribute*), 26
sample_accession (*omicidx.models.SraRun attribute*), 28
sample_join_accessions() (*in module omicidx.sra.etl*), 17
SampleJson (*class in omicidx.model.etl_schema*), 10
save_object() (*omicidx.db.OmicIDXDb method*), 20
spot_length (*omicidx.model.public.SraRun attribute*), 13
spot_length (*omicidx.models.SraRun attribute*), 28
spots (*omicidx.model.etl.SraAccession attribute*), 7
spots (*omicidx.model.public.SraRun attribute*), 13
spots (*omicidx.models.SraAccession attribute*), 26
spots (*omicidx.models.SraRun attribute*), 28
sra_experiment (*omicidx.models.SraRun attribute*), 28
sra_object_generator() (*in module omicidx.sra_parsers*), 33
sra_sample (*omicidx.model.public.Biosample attribute*), 10
sra_sample (*omicidx.models.SraExperiment attribute*), 26
sra_sample (*omicidx.models.SraRun attribute*), 28
sra_study (*omicidx.models.SraExperiment attribute*), 26
sra_study (*omicidx.models.SraRun attribute*), 28
sra_study (*omicidx.models.SraSample attribute*), 28
SraAccession (*class in omicidx.model.etl*), 7
SraAccession (*class in omicidx.models*), 25
SRAExperiment (*class in omicidx.model.etl_schema*), 9
SraExperiment (*class in omicidx.model.public*), 11
SraExperiment (*class in omicidx.models*), 26
SraExperiment (*class in omicidx.sra.pydantic_models*), 18
SraExperimentAttribute (*class in omicidx.model.public*), 12
SraExperimentIdentifier (*class in omicidx.model.public*), 12
SRAExperimentPackage (*class in omicidx.sra_parsers*), 31
SRAExperimentRecord (*class in omicidx.sra_parsers*), 31
SraLibraryLayout (*class in omicidx.model.public*), 12
SraLibraryLayout (*class in omicidx.models*), 27
SraLibrarySelection (*class in omicidx.model.public*), 12
SraLibrarySelection (*class in omicidx.models*), 27
SraLibrarySource (*class in omicidx.model.public*), 12
SraLibrarySource (*class in omicidx.models*), 27
SraLibraryStrategy (*class in omicidx.model.public*), 12
SraLibraryStrategy (*class in omicidx.models*), 27
SraPlatform (*class in omicidx.model.public*), 12
SraPlatform (*class in omicidx.models*), 27
SraRun (*class in omicidx.model.public*), 12
SraRun (*class in omicidx.models*), 27
SraRun (*class in omicidx.sra.pydantic_models*), 18
SraRunAttribute (*class in omicidx.model.public*), 13
SRARunRecord (*class in omicidx.sra_parsers*), 31
SraSample (*class in omicidx.model.public*), 13
SraSample (*class in omicidx.models*), 28
SraSample (*class in omicidx.sra.pydantic_models*), 18

SraSampleAttribute (*class in omicidx.model.public*), 14
SraSampleIdentifier (*class in omicidx.model.public*), 14
SRASampleRecord (*class in omicidx.sra_parsers*), 31
SraSampleXref (*class in omicidx.model.public*), 14
srastatrep() (*in module omicidx.sra_parsers*), 33
SRAStudy (*class in omicidx.model.etl_schema*), 9
SraStudy (*class in omicidx.model.public*), 14
SraStudy (*class in omicidx.models*), 28
SraStudy (*class in omicidx.sra.pydantic_models*), 18
SraStudyAttribute (*class in omicidx.model.public*), 15
SraStudyIdentifier (*class in omicidx.model.public*), 15
SRAStudyRecord (*class in omicidx.sra_parsers*), 31
SraStudyXref (*class in omicidx.model.public*), 15
SRASubmissionRecord (*class in omicidx.sra_parsers*), 31
SRAXMLRecord (*class in omicidx.sra_parsers*), 31
status (*omicidx.model.etl.SraAccession attribute*), 7
status (*omicidx.model.etl_schema.SRAStudy attribute*), 9
status (*omicidx.model.public.SraExperiment attribute*), 11
status (*omicidx.model.public.SraRun attribute*), 13
status (*omicidx.model.public.SraSample attribute*), 13
status (*omicidx.model.public.SraStudy attribute*), 15
status (*omicidx.models.GEOPlatform attribute*), 25
status (*omicidx.models.GEOSeries attribute*), 25
status (*omicidx.models.SraAccession attribute*), 26
status (*omicidx.models.SraExperiment attribute*), 27
status (*omicidx.models.SraRun attribute*), 28
status (*omicidx.models.SraSample attribute*), 28
status (*omicidx.models.SraStudy attribute*), 29
StringType (*class in omicidx.schema_tools*), 30
study (*omicidx.model.etl.SraAccession attribute*), 7
study (*omicidx.models.SraAccession attribute*), 26
study() (*omicidx.sra_parsers.SRAExperimentPackage method*), 31
study_accession (*omicidx.model.etl_schema.SRAExperiment attribute*), 9
study_accession (*omicidx.model.public.SraExperiment attribute*), 11
study_accession (*omicidx.model.public.SraSample attribute*), 13
study_accession (*omicidx.models.SraExperiment attribute*), 27
study_accession (*omicidx.models.SraRun attribute*), 28
study_accession (*omicidx.models.SraSample attribute*), 28
study_join_accessions() (*in module omicidx.sra.etl*), 17
study_type (*omicidx.model.etl_schema.SRAStudy attribute*), 9
study_type (*omicidx.model.public.SraStudy attribute*), 15
study_type (*omicidx.models.SraStudy attribute*), 29
StudyJson (*class in omicidx.model.etl_schema*), 10
submission (*omicidx.model.etl.SraAccession attribute*), 9
submission (*omicidx.models.SraAccession attribute*), 26
submission_date (*omicidx.model.public.Biosample attribute*), 10
submission_date (*omicidx.models.Biosample attribute*), 23
submission_date (*omicidx.models.GEOPlatform attribute*), 25
submission_date (*omicidx.models.GEOSeries attribute*), 25
summary (*omicidx.models.GEOPlatform attribute*), 25
summary (*omicidx.models.GEOSeries attribute*), 25

T

table_to_gcs() (*in module omicidx.bigquery_utils*), 19
tag (*omicidx.model.public.AttributesMixin attribute*), 10
tag (*omicidx.model.public.SraExperimentAttribute attribute*), 12
tag (*omicidx.model.public.SraRunAttribute attribute*), 13
tag (*omicidx.model.public.SraSampleAttribute attribute*), 14
tag (*omicidx.model.public.SraStudyAttribute attribute*), 15
tag (*omicidx.models.AttributesMixin attribute*), 23
TaxCountAnalysis (*class in omicidx.sra.pydantic_models*), 18
TaxCountEntry (*class in omicidx.sra.pydantic_models*), 18
taxon_id (*omicidx.model.public.Biosample attribute*), 10
taxon_id (*omicidx.model.public.SraSample attribute*), 14
taxon_id (*omicidx.models.Biosample attribute*), 23
taxon_id (*omicidx.models.SraSample attribute*), 28
taxonomy_name (*omicidx.model.public.Biosample attribute*), 10
taxonomy_name (*omicidx.models.Biosample attribute*), 23
technology (*omicidx.models.GEOPlatform attribute*), 25
textsearchable_index_col (*omicidx.model.public.Biosample attribute*),

10
title (*omicidx.model.etl_schema.SRAStudy attribute*), 9
title (*omicidx.model.public.Biosample attribute*), 10
title (*omicidx.model.public.SraExperiment attribute*), 11
title (*omicidx.model.public.SraSample attribute*), 14
title (*omicidx.model.public.SraStudy attribute*), 15
title (*omicidx.models.Biosample attribute*), 23
title (*omicidx.models.GEOPlatform attribute*), 25
title (*omicidx.models.GEOSeries attribute*), 25
title (*omicidx.models.SraExperiment attribute*), 27
title (*omicidx.models.SraSample attribute*), 28
title (*omicidx.models.SraStudy attribute*), 29
to_tsvector_ix() (*in module omicidx.model.public*), 15
try_update() (*in module omicidx.sra_parsers*), 34
type (*omicidx.model.etl.SraAccession attribute*), 9
type (*omicidx.models.SraAccession attribute*), 26

U

updated (*omicidx.model.etl.SraAccession attribute*), 9
updated (*omicidx.model.etl_schema.SRAStudy attribute*), 9
updated (*omicidx.model.public.SraExperiment attribute*), 11
updated (*omicidx.model.public.SraRun attribute*), 13
updated (*omicidx.model.public.SraSample attribute*), 14
updated (*omicidx.model.public.SraStudy attribute*), 15
updated (*omicidx.models.SraAccession attribute*), 26
updated (*omicidx.models.SraExperiment attribute*), 27
updated (*omicidx.models.SraRun attribute*), 28
updated (*omicidx.models.SraSample attribute*), 28
updated (*omicidx.models.SraStudy attribute*), 29
upload_biosample() (*in module omicidx.scripts.cli*), 16
upload_blob_to_gcs() (*in module omicidx.gcs_utils*), 20
uuid (*omicidx.model.public.IdentifiersMixin attribute*), 11
uuid (*omicidx.model.public.SraExperimentIdentifier attribute*), 12
uuid (*omicidx.model.public.SraSampleIdentifier attribute*), 14
uuid (*omicidx.model.public.SraSampleXref attribute*), 14
uuid (*omicidx.model.public.SraStudyIdentifier attribute*), 15
uuid (*omicidx.model.public.SraStudyXref attribute*), 15
uuid (*omicidx.model.public.XrefsMixin attribute*), 15
uuid (*omicidx.models.XrefsMixin attribute*), 29

V

value (*omicidx.model.public.AttributesMixin attribute*), 10
value (*omicidx.model.public.SraExperimentAttribute attribute*), 12
value (*omicidx.model.public.SraLibraryLayout attribute*), 12
value (*omicidx.model.public.SraLibrarySelection attribute*), 12
value (*omicidx.model.public.SraLibrarySource attribute*), 12
value (*omicidx.model.public.SraLibraryStrategy attribute*), 12
value (*omicidx.model.public.SraPlatform attribute*), 12
value (*omicidx.model.public.SraRunAttribute attribute*), 13
value (*omicidx.model.public.SraSampleAttribute attribute*), 14
value (*omicidx.model.public.SraStudyAttribute attribute*), 15
value (*omicidx.models.AttributesMixin attribute*), 23
value (*omicidx.models.BiosampleAttribute attribute*), 24
value (*omicidx.models.SraLibraryLayout attribute*), 27
value (*omicidx.models.SraLibrarySelection attribute*), 27
value (*omicidx.models.SraLibrarySource attribute*), 27
value (*omicidx.models.SraLibraryStrategy attribute*), 27
value (*omicidx.models.SraPlatform attribute*), 27
visibility (*omicidx.model.etl.SraAccession attribute*), 9
visibility (*omicidx.model.etl_schema.SRAStudy attribute*), 10
visibility (*omicidx.model.public.SraExperiment attribute*), 11
visibility (*omicidx.model.public.SraRun attribute*), 13
visibility (*omicidx.model.public.SraSample attribute*), 14
visibility (*omicidx.model.public.SraStudy attribute*), 15
visibility (*omicidx.models.SraAccession attribute*), 26
visibility (*omicidx.models.SraExperiment attribute*), 27
visibility (*omicidx.models.SraRun attribute*), 28
visibility (*omicidx.models.SraSample attribute*), 28
visibility (*omicidx.models.SraStudy attribute*), 29

W

walk_schema() (*in module omicidx.schema_tools*), 30
web_link (*omicidx.models.GEOContact attribute*), 24

X

Xref (*class in omicidx.sra.pydantic_models*), 18
xrefs (*omicidx.model.etl_schema.SRAStudy attribute*), 10
xrefs (*omicidx.model.public.SraExperiment attribute*), 11
xrefs (*omicidx.model.public.SraSample attribute*), 14
xrefs (*omicidx.model.public.SraStudy attribute*), 15
xrefs (*omicidx.models.SraExperiment attribute*), 27
xrefs (*omicidx.models.SraSample attribute*), 28
xrefs (*omicidx.models.SraStudy attribute*), 29
XrefsMixin (*class in omicidx.model.public*), 15
XrefsMixin (*class in omicidx.models*), 29