
LabbookDB Documentation

Release 0.0.1

Horea Christian

Mar 23, 2020

Contents

1	Presentations	3
1.1	Video	3
1.2	Publications	3
2	Installation	5
2.1	Gentoo Linux	5
2.2	Python Package Manager (Users)	5
2.3	Python Package Manager (Developers)	5
3	Example Input	7
4	Dependencies	9
4.1	Optional Dependencies for Introspection	9
4.2	Optional Dependencies for PDF Protocol Generation	9
4.3	Optional Dependencies for Plotting	9
4.3.1	labbookdb	9
4.3.2	Objects Reference	39
	Python Module Index	41
	Index	43

This package contains a relational database structure for life science research, and a number of functions to conveniently add and retrieve information - and generate summaries. The core concept of LabbookDB is that most of the information classically tracked in a lab book can be more efficiently and more reliably stored in a relational database.

In comparison to a paper notebook, an **electronic** lab book is:

- More easily stored
- More easily shared
- More easily backed up

In comparison with other electronic formats based on a document concept, a **database** of experimental metadata is:

- More easily browsed
- More easily queried
- More easily integrated into data analysis functions

1.1 Video

- [LabbookDB - A Relational Framework for Laboratory Metadata](#), at SciPy 2017 in Austin (TX,USA).

1.2 Publications

- [LabbookDB - A Wet-Work-Tracking Database Application Framework](#), in Proceedings of the 15th Python in Science Conference (SciPy 2017).

2.1 Gentoo Linux

LabbookDB is available for Portage (the package manager of Gentoo Linux, derivative distributions, as well as BSD) via the [Gentoo Science Overlay](#). Upon enabling the overlay, the package can be emerged:

```
emerge labbookdb
```

2.2 Python Package Manager (Users)

Python's *setuptools* allows you to install Python packages independently of your distribution (or operating system, even). This approach cannot manage any of our numerous non-Python dependencies (by design) and at the moment will not even manage Python dependencies; as such, given any other alternative, **we do not recommend this approach**:

```
git clone git@github.com:TheChymera/LabbookDB.git
cd LabbookDB
python setup.py install --user
```

2.3 Python Package Manager (Developers)

Python's *setuptools* allows you to install Python packages independently of your distribution (or operating system, even); it also allows you to install a “live” version of the package - dynamically linking back to the source code. This permits you to test code (with real module functionality) as you develop it. This method is sub-par for dependency management (see above notice), but - as a developer - you should be able to manually ensure that your package manager provides the needed packages.

```
git clone git@github.com:TheChymera/LabbookDB.git
cd LabbookDB
echo "export PATH=\$HOME/.local/bin/:\$PATH" >> ~/.bashrc
source ~/.bashrc
python setup.py develop --user
```

CHAPTER 3

Example Input

LabbookDB is designed to organize complex wet work data. We publish example input to generate a relationship-rich database in a separate repository, [demolog](#).

- Argh
- Pandas
- simplejson
- SQLAlchemy

4.1 Optional Dependencies for Introspection

- Sdisplay

4.2 Optional Dependencies for PDF Protocol Generation

- TeX Live

4.3 Optional Dependencies for Plotting

- BehavioPy
- Matplotlib

4.3.1 labbookdb

labbookdb package

Subpackages

labbookdb.db package

Submodules

labbookdb.db.add module

`labbookdb.db.add.add_generic(db_path, parameters, session=None, engine=None)`

Adds new entries based on a LabbookDB-syntax parameter dictionary.

Parameters

- **db_path** (*str*) – Path to database to open if session and engine parameters are not already passed, can be relative or use tilde to specify the user \$HOME.
- **parameters** (*str or dict*) – A LabbookDB-style dictionary (or JSON interpretable as dictionary), where keys are “CATEGORY” and other strings specifying the attribute names for the object to be created, and values are the class name (for “CATEGORY”) and either the values to assign (verbatim: string, int, or float) or LabbookDB-syntax strings specifying a related entry, or a list of LabbookDB-syntax strings specifying related entries, or a list of LabbookDB-style dictionaries specifying new entries to be created and linked.
- **session** (*sqlalchemy.orm.session.Session, optional*) – Session instance, as created with `labbookdb.db.add.load_session()`.
- **engine** (*sqlalchemy.engine.Engine, optional*) – Engine instance corresponding to the Session instance under session, as created with `labbookdb.db.add.load_session()`.

Returns

- **myobject** (*object*) – LabbookDB object with SQLAlchemy-compatible attributes (e.g. as found under `labbookdb.db.common_classes`).
- **object_id** (*int*) – Value of `myobject.id` attribute.

`labbookdb.db.add.add_to_db(session, engine, myobject)`

Add an object to session and return the `.id` attribute value.

Parameters

- **session** (*sqlalchemy.orm.session.Session*) – Session instance, as created with `labbookdb.db.add.load_session()`.
- **engine** (*sqlalchemy.engine.Engine*) – Engine instance corresponding to the Session instance under session, as created with `labbookdb.db.add.load_session()`.
- **myobject** (*object*) – LabbookDB object with SQLAlchemy-compatible attributes (e.g. as found under `labbookdb.db.common_classes`).

Returns `object_id` – Value of `myobject.id` attribute

Return type

`int`

`labbookdb.db.add.append_parameter(db_path, entry_identification, parameters)`

Assigns a value to a given parameter of a given entry.

Parameters

- **db_path** (*str*) – A string specifying the database path

- **entry_identification** (*str*) – A LabbookDB syntax string specifying an instance of an object for which to update a parameter. Example strings: “Animal:external_ids.AnimalExternalIdentifier:database.ETH/AIC/cdb&identifier.275511” , “Cage:id.14”
- **parameters** (*str or dict*) – A LabbookDB-style dictionary (or JSON interpretable as dictionary), where keys are strings giving the names of attributes of the class selected by entry_identification, and values are either the values to assign (verbatim: string, int, or float) or LabbookDB-syntax strings specifying a related entry, or a list of LabbookDB-syntax strings specifying related entries, or a list of LabbookDB-style dictionaries specifying new entries to be created and linked.

labbookdb.db.add.**commit_and_close**(*session, engine*)

Commit and close session and dispose of engine. Nonfatal for sqlalchemy.exc.IntegrityError with print notification.

Parameters

- **session** (*sqlalchemy.orm.session.Session, optional*) – Session instance, as created with labbookdb.db.add.load_session().
- **engine** (*sqlalchemy.engine.Engine, optional*) – Engine instance corresponding to the Session instance under session, as created with labbookdb.db.add.load_session().

labbookdb.db.add.**get_related_ids**(*session, engine, parameters*)

Return the .id attribute value for existing entries matched by a string following the LabbookDB-syntax.

Parameters

- **session** (*sqlalchemy.orm.session.Session*) – Session instance, as created with labbookdb.db.add.load_session().
- **engine** (*sqlalchemy.engine.Engine*) – Engine instance corresponding to the Session instance under session, as created with labbookdb.db.add.load_session().
- **parameters** (*str*) – LabbookDB-syntax string specifying an existing entry.

Returns

- **ids** (*list of int*) – .id attribute values for the entries matched by the LabbookDB-syntax string.
- **sql_query** (*sqlalchemy.orm.query.Query*) – Query corresponding to the LabbookDB-syntax string

Examples

```
>>> from labbookdb.db import add
>>> session, engine = add.load_session("lala.db")
>>> add.get_related_ids(s,e,"Animal:external_ids.
↳AnimalExternalIdentifier:database.ETH/AIC/cdb&#x26;identifier.275511")
BaseException: No entry was found with a value of "275511" on the "identifier"
↳column of the "AnimalExternalIdentifier" CATEGORY, in the database.
>>> add.get_related_ids(s,e,"Animal:external_ids.
↳AnimalExternalIdentifier:database.ETH/AIC/cdb&#x26;identifier.275511")
BaseException: No entry was found with a value of "ETH/AIC/cdb" on the "database"
↳column of the "AnimalExternalIdentifier" CATEGORY, in the database.
```

Notes

Recursivity: This function calls itself recursively in order to get the .id attribute values of related entries (and related entries of related entries, etc.) specified in the LabbookDB-syntax string. Multiple constraints are separated by double ampersands which may be separated by none or up to two hashtags, to specify at which level the constraints should be applied to. One hashtag is removed on each recursion step, and the constraint is only evaluated when there are no hashtags left. “Animal:external_ids.AnimalExternalIdentifier:database.ETH/AIC/cdb&#&identifier.275511” will look for both the database and the identifier attributes in the AnimalExternalIdentifier class, while “Animal:external_ids.AnimalExternalIdentifier:database.ETH/AIC/cdb&#&identifier.275511” will look for the database attribute on the AnimalExternalIdentifier class, and for the identifier attribute on the Animal class.

`labbookdb.db.add.instructions(kind)`

Print helpful instructions for common error-causing input issues

Parameters `kind` (`{ "table_identifier", }`) – Shorthand for the instruction message to be printed.

`labbookdb.db.add.load_session(db_path)`

Load and return a new SQLAlchemy session and engine.

Parameters `db_path` (`str`) – Path to desired database location, can be relative or use tilde to specify the user \$HOME.

Returns

- `session` (`sqlalchemy.orm.session.Session`) – Session instance.
- `engine` (`sqlalchemy.engine.Engine`) – Engine instance.

labbookdb.db.base_classes module

```
class labbookdb.db.base_classes.Biopsy(**kwargs)
```

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

`animal_id`

`extraction_protocol`

`extraction_protocol_id`

`fluorescent_microscopy`

`id`

`sample_location`

`start_date`

`type`

```
class labbookdb.db.base_classes.FMRIScannerSetup(**kwargs)
```

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

`code`

`coil`

`id`

`resonator`


```
    scanner
    support
class labbookdb.db.base_classes.Genotype(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    code
    construct
    id
    zygotity
class labbookdb.db.base_classes.Irregularity(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    description
    id
class labbookdb.db.base_classes.Measurement(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    animal_id
    cage_id
    date
    id
    irregularities
    operator
    operator_id
    type
class labbookdb.db.base_classes.MeasurementUnit(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    code
    id
    long_name
    siunitx
class labbookdb.db.base_classes.Operator(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    affiliation
    code
    email
    full_name
    id
    validate_email(key, address)
class labbookdb.db.base_classes.OpticFiberImplant(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
```

```
angle
cannula_diameter
code
ferrule_diameter
id
length
long_code
manufacturer
manufacturer_code
numerical_aperture
transmittance

class labbookdb.db.base_classes.OrthogonalStereotacticTarget (**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    code
    depth
    id
    leftright
    pitch
    posteroanterior
    qualitative_depth_reference
    reference
    roll
    superoinferior
    yaw

class labbookdb.db.base_classes.Protocol (**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    authors
    code
    id
    name
    type

class labbookdb.db.base_classes.StimulationEvent (**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    duration
    frequency
    id
    onset
```

```
pulse_width
strength
target
trial_type
unit
unit_id
wavelength

class labbookdb.db.base_classes.Virus(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    addgene_identifier
    capsid
    code
    concentration
    credit
    id
    plasmid_summary
    source
```

labbookdb.db.common_classes module

```
class labbookdb.db.common_classes.AnesthesiaProtocol(**kwargs)
    Bases: labbookdb.db.base_classes.Protocol
    authors
    bolus
    bolus_to_maintenance_delay
    code
    id
    induction
    maintenance
    name
    recovery_bolus
    respiration
    type

class labbookdb.db.common_classes.Animal(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    biopsies
    birth_date
    cage_stays
```

```
death_date
death_reason
ear_punches
external_ids
genotypes
id
license
maximal_severity
measurements
observations
operations
sex
treatments

class labbookdb.db.common_classes.AnimalExternalIdentifier(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    animal_id
    database
    id
    identifier

class labbookdb.db.common_classes.Arena(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    code
    id
    measurements
    name
    shape
    wall_color
    x_dim
    y_dim
    z_dim

class labbookdb.db.common_classes.BrainBiopsy(**kwargs)
    Bases: labbookdb.db.base_classes.Biopsy
    animal_id
    data_path
    extraction_protocol
    extraction_protocol_id
    fluorescent_microscopy
```

```
id
sample_location
sectioning_protocol
sectioning_protocol_id
start_date
type

class labbookdb.db.common_classes.BrainExtractionProtocol(**kwargs)
    Bases: labbookdb.db.base_classes.Protocol

    authors
    code
    fixation_solution
    fixation_solution_id
    fixation_solution_volume
    flushing_solution
    flushing_solution_id
    flushing_solution_volume
    id
    name
    perfusion_flow
    perfusion_system
    peristaltic_frequency
    post_extraction_fixation_time
    storage_solution
    storage_solution_id
    storage_solution_volume
    storage_time
    type

class labbookdb.db.common_classes.Cage(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base

    environmental_enrichment
    handling_habituations
    id
    id_local
    location
    measurements
    stays
    treatments
```

```
class labbookdb.db.common_classes.CageStay (**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base

    cage
    cage_id
    id
    report_animals()
    single_caged
    start_date

class labbookdb.db.common_classes.DNAExtraction (**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base

    code
    id
    protocol
    protocol_id
    source
    source_id

class labbookdb.db.common_classes.DNAExtractionProtocol (**kwargs)
    Bases: labbookdb.db.base_classes.Protocol

    authors
    centrifugation
    centrifugation_id
    code
    cooling
    cooling_id
    digestion
    digestion_buffer
    digestion_buffer_id
    digestion_buffer_volume
    digestion_id
    id
    inactivation
    inactivation_id
    lysis
    lysis_buffer
    lysis_buffer_id
    lysis_buffer_volume
    lysis_id
```

```
mass_unit
mass_unit_id
name
proteinase
proteinase_id
proteinase_volume
sample_mass
type
volume_unit
volume_unit_id

class labbookdb.db.common_classes.DrinkingMeasurement (**kwargs)
    Bases: labbookdb.db.base_classes.Measurement
    animal_id
    cage_id
    consumption
    date
    end_amount
    id
    irregularities
    operator
    operator_id
    reference_date
    start_amount
    type

class labbookdb.db.common_classes.Evaluation (**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    author
    author_id
    id
    measurement_id
    path

class labbookdb.db.common_classes.FMRIMeasurement (**kwargs)
    Bases: labbookdb.db.base_classes.Measurement
    anesthesia
    anesthesia_id
    animal_id
    cage_id
```

```
data_path
date
id
irregularities
operator
operator_id
scanner_setup
scanner_setup_id
stimulations
temperature
type
class labbookdb.db.common_classes.FluorescentMicroscopyMeasurement (**kwargs)
    Bases: labbookdb.db.base_classes.Measurement
    animal_id
    biopsy_id
    cage_id
    data
    date
    exposure
    id
    imaged_wavelength
    irregularities
    light_source
    operator
    operator_id
    stimulation_wavelength
    type
class labbookdb.db.common_classes.ForcedSwimTestMeasurement (**kwargs)
    Bases: labbookdb.db.base_classes.Measurement
    animal_id
    cage_id
    data_path
    date
    evaluations
    id
    irregularities
    operator
```



```
operator_id
recording_bracket
temperature
type
class labbookdb.db.common_classes.HandlingHabituation(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    cage
    cage_id
    date
    id
    protocol
    protocol_id
class labbookdb.db.common_classes.HandlingHabituationProtocol(**kwargs)
    Bases: labbookdb.db.base_classes.Protocol
    authors
    code
    group_picking_up
    id
    individual_picking_up
    name
    session_duration
    transparent_tube
    type
class labbookdb.db.common_classes.Incubation(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    duration
    duration_unit
    duration_unit_id
    id
    movement
    revolutions_per_minute
    temperature
    temperature_unit
    temperature_unit_id
class labbookdb.db.common_classes.Ingredient(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    concentration
```

```
concentration_unit
concentration_unit_id
id
substance
substance_id

class labbookdb.db.common_classes.Observation(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    animal_id
    behaviour
    date
    id
    operator
    operator_id
    physiology
    severtity
    unit
    unit_id
    value

class labbookdb.db.common_classes.OpenFieldTestMeasurement(**kwargs)
    Bases: labbookdb.db.base_classes.Measurement
    animal_id
    arena_id
    cage_id
    center_luminostiy
    corner_luminostiy
    data_path
    date
    edge_luminostiy
    evaluations
    id
    irregularities
    operator
    operator_id
    type

class labbookdb.db.common_classes.Operation(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    anesthesia
```

```
    anesthesia_id
    animal_id
    date
    id
    irregularities
    operator
    operator_id
    protocols

class labbookdb.db.common_classes.OpticFiberImplantProtocol(**kwargs)
    Bases: labbookdb.db.base_classes.Protocol

    authors
    code
    id
    name
    optic_fiber_implant
    optic_fiber_implant_id
    stereotactic_target
    stereotactic_target_id
    type

class labbookdb.db.common_classes.SectioningProtocol(**kwargs)
    Bases: labbookdb.db.base_classes.Protocol

    authors
    blade_frequency
    blade_speed
    code
    id
    name
    slice_thickness
    start_bregma_distance
    start_depth
    start_interaural_distance
    start_lambda_distance
    start_midline_distance
    system
    type

class labbookdb.db.common_classes.Solution(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
```

```
code
contains
id
name
supplier
supplier_product_code

class labbookdb.db.common_classes.StimulationProtocol(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    code
    events
    id
    name

class labbookdb.db.common_classes.Substance(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    code
    concentration
    concentration_unit
    concentration_unit_id
    id
    long_name
    name
    pubchem_sid
    supplier
    supplier_product_code

class labbookdb.db.common_classes.SucrosePreferenceMeasurement(**kwargs)
    Bases: labbookdb.db.base_classes.Measurement
    animal_id
    cage_id
    concentration_unit
    concentration_unit_id
    date
    id
    irregularities
    operator
    operator_id
    reference_date
    sucrose_bottle_position
```

```
sucrose_concentration
sucrose_end_amount
sucrose_start_amount
type
water_end_amount
water_start_amount

class labbookdb.db.common_classes.Treatment(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Base
    end_date
    id
    protocol
    protocol_id
    start_date

class labbookdb.db.common_classes.TreatmentProtocol(**kwargs)
    Bases: labbookdb.db.base_classes.Protocol
    authors
    code
    dose
    dose_unit
    dose_unit_id
    frequency
    id
    name
    rate
    rate_unit
    rate_unit_id
    route
    solution
    solution_id
    type

class labbookdb.db.common_classes.VirusInjectionProtocol(**kwargs)
    Bases: labbookdb.db.base_classes.Protocol
    amount
    authors
    code
    id
    name
```

```
stereotactic_target
stereotactic_target_id
type
virus
virus_diffusion_time
virus_id
virus_injection_speed
```

```
class labbookdb.db.common_classes.WeightMeasurement (**kwargs)
    Bases: labbookdb.db.base_classes.Measurement
    animal_id
    cage_id
    date
    id
    irregularities
    operator
    operator_id
    type
    weight
    weight_unit
    weight_unit_id
```

labbookdb.db.query module

```
labbookdb.db.query.add_all_columns(cols, class_name)
```

```
labbookdb.db.query.animal_info(identifier, database, db_path=None)
```

Return the `__str__` attribute of an Animal object query filtered by the id column OR by arguments of the external_id objects.

Parameters

- `db_path(string)` –
- to a LabbookDB formatted database. (*Path*) –
- `identifier(int or string)` –
- identifier of the animal (*The*) –
- `database(string or None, optional)` –
- specified gives a constraint on the `AnimalExternalIdentifier.database` column AND truns the identifier attribute into a constraint on the `AnimalExternalIdentifier.identifier` column. If unspecified, the identifier argument is used as a constraint on the `Animal.id` column. (*If*) –

```
labbookdb.db.query.cage_info(db_path, identifier)
```

Return the `__str__` attribute of an `Animal` object query filtered by the `id` column OR by arguments of the external_id objects.

Parameters

- `db_path(string)` –
- to a LabbookDB formatted database. (*Path*) –
- `identifier(int or string)` –
- identifier of the animal (*The*) –
- `database(string or None, optional)` –
- specified gives a constraint on the `AnimalExternalIdentifier.database` column AND truns the `identifier` attribute into a constraint on the `AnimalExternalIdentifier.identifier` column. If unspecified, the `identifier` argument is used as a constraint on the `Animal.id` column. (*If*) –

```
labbookdb.db.query.commit_and_close(session, engine)
```

```
labbookdb.db.query.get_df(db_path, col_entries=[], default_join='inner', filters=[],
                           join_entries=[], join_types=[])
```

Return a dataframe from a complex query of a LabbookDB-style database

Parameters

- `db_path(string)` – Path to database file.
- `col_entries(list)` – A list of tuples containing the columns to be queried: * 1-tuples indicate all attributes of a class are to be queried * 2-tuples indicate only the attribute specified by the second element, of the class specified by the first element is to be queried * 3-tuples indicate that an aliased class of the type given by the second element is to be created and named according to the first and second elements, separated by an underscore. The attribute given by the third element will be queried; if the thid element is empty, all attributes will be queried
- `join_entries(list)` – A list of tuples specifying the desired join operations: * 1-tuples give the join * 2-tuples give the class to be joined on the first element, and the explicit relationship (attribute of another class) on the second element If any of the elements contains a period, the expression will be evaluated as a class (preceeding the period) attribute (after the period)\$
- `filters(list)` – A list of lists giving filters for the query. In each sub-list the first and second elements give the class and attribute to be matched. Every following element specifies a possible value for the class attribute (implemented as inclusive disjunction). If the attribute name ends in “date” the function computes datetime objects from the subsequent strings containing numbers separated by commas.
- `documentation(!!!incomplete)` –

Examples

```
>>> col_entries=[
    ("Animal", "id"),
    ("Treatment", ),
    ("FMRIMeasurement", ),
```

(continues on next page)

(continued from previous page)

```
        ("TreatmentProtocol", "code"),
        ("Cage", "id"),
        ("Cage", "Treatment", ""),
        ("Cage", "TreatmentProtocol", "code")
    ]
>>> join_entries=[
        ("Animal.treatments",),
        ("FMRIMeasurement",),
        ("Treatment.protocol",),
        ("Animal.cage_stays",),
        ("CageStay.cage",),
        ("Cage_Treatment", "Cage.treatments"),
        ("Cage_TreatmentProtocol", "Cage_Treatment.protocol")
    ]
>>> filters = [{"Cage_Treatment", "start_date", "2016,4,25,19,30"}]
>>> reference_df = get_df("~/syncdata/meta.db", col_entries=col_entries, join_
↪entries=join_entries, filters=filters)
```

`labbookdb.db.query.get_for_protocolize(db_path, class_name, code)`

Return a dataframe containing a specific entry from a given class name, joined with its related tables up to three levels down.

`labbookdb.db.query.get_related_id(session, engine, parameters)`

`labbookdb.db.query.load_session(db_path)`

labbookdb.db.utils module

`labbookdb.db.utils.arange_by_date(attribute)`

`labbookdb.db.utils.dt_format(dt)`

Module contents

labbookdb.evaluate package

Submodules

labbookdb.evaluate.manual module

Module contents

labbookdb.introspection package

Submodules

labbookdb.introspection.schema module

Module contents

labbookdb.report package

Submodules

labbookdb.report.behaviour module

labbookdb.report.development module

```
labbookdb.report.development.animal_multiselect (db_path, cage_treatments=[],
                                                    implant_targets=['dr_impl'],
                                                    virus_targets=['dr_skull',
                                                                    'dr_dura',
                                                                    'dr_dura_shallow',
                                                                    'dr_skull_perpendicular'],
                                                    geno-
                                                    types=['eptg'], external_id="")
```

```
labbookdb.report.development.animal_weights_()
```

```
labbookdb.report.development.drinking_water_by_cage_treatment (treatment_relative_date=True,
                                                                rounding='D')
```

labbookdb.report.examples module

labbookdb.report.formatting module

```
labbookdb.report.formatting.plottable_sucrosepreference_df (reference_df)
```

```
labbookdb.report.formatting.plottable_sums (reference_df, behaviour, identifier_column='Animal_id',
                                              periods={}, period_label='period',
                                              meta-
                                              data_columns={'TreatmentProtocol_code':
                                                           'Treatment'})
```

labbookdb.report.processing module

```
labbookdb.report.processing.rounded_days (datetime_obj)
```

```
labbookdb.report.processing.timedelta_sums (evaluation_path, index_name="",
                                              period_start=False, period_end=False)
```

Return the per-behaviour sums of timedelta intervals.

Parameters

- **timedelta_df** (*pandas_dataframe*) – A pandas dataframe containing a “behaviour” and a “timedelta” column
- **index_name** (*string, optional*) – The name to add as an index of the returned series (useful for concatenating multiple outputs)
- **period_start** (*float, optional*) – The timepoint at which the evaluation period for the timedelta sums starts.
- **period_end** (*float, optional*) – The timepoint at which the evaluation period for the timedelta sums ends.

`labbookdb.report.processing.timedeltas` (*evaluation_path*, *period_start=False*, *period_end=False*)

Return per-behaviour timedelta intervals.

Parameters

- **timedelta_df** (*pandas_dataframe*) – A pandas dataframe containing a “behaviour” and a “start” column
- **period_start** (*float*, *optional*) – The timepoint at which the evaluation period for the timedelta starts.
- **period_end** (*float*, *optional*) – The timepoint at which the evaluation period for the timedelta ends.

labbookdb.report.protocolize module

`labbookdb.report.protocolize.compose_tex` (*db_path*, *class_name*, *code*)

Create a TeX document containing the protocol of the *class_name* entry identified by a given code.

`labbookdb.report.protocolize.print_document` (*tex*, *pdfname='protocol'*)

labbookdb.report.selection module

`labbookdb.report.selection.animal_id` (*db_path*, *database*, *identifier*, *reverse=False*)

Return the main LabbookDB animal identifier given an external database identifier.

Parameters

- **db_path** (*string*) – Path to the database file to query.
- **database** (*string*) – Valid *AnimalExternalIdentifier.database* value.
- **identifier** (*string*) – Valid *AnimalExternalIdentifier.identifier* value.
- **reverse** (*bool*, *optional*) – Whether to reverse the query. A reverse query means that a LabbookDB *Animal.id* filter is applied and an *AnimalExternalIdentifier.identifier* value is returned.

Returns LabbookDB animal identifier.

Return type `int`

`labbookdb.report.selection.animal_operations` (*db_path*, *animal_ids=[]*, *implant_targets=[]*, *virus_targets=[]*)

Select a dataframe of animals having been subjected to operative interventions with the given anatomical targets.

Parameters

- **db_path** (*str*) –
- **to a LabbookDB formatted database.** (*Path*) –
- **animal_ids** (*list*, *optional*) – A List of LabbookDB *Animal.id* values by which to filter the query. It is faster to filter using this mechanism than to return a dataframe for all animals and then filter that.
- **implant_targets** (*list*, *optional*) – A List of LabbookDB *OrthogonalStereotacticTarget.code* values which should be used to filter the query, while being joined to *Operation* objects via the *OpticFiberImplantProtocol* class. It is faster to filter using this mechanism than to return a dataframe for all animals and then filter that.

- **virus_targets** (*list, optional*) – A List of LabbookDB *OrthogonalStereotactic-Target.code* values which should be used to filter the query, while being joined to *Operation* objects via the *VirusInjectionProtocol* class. It is faster to filter using this mechanism than to return a dataframe for all animals and then filter that.

Notes

CAREFUL: Providing both *implant_targets* and *virus_targets* will return entries including only animals which have undergone an operation which has included protocols targeting both areas. If the areas were targeted by protocols included in different operations, the correspondence will not be detected. To obtain such a list please call the function twice and create a new dataframe from the intersection of the inputs on the *Animal_id* column.

```
labbookdb.report.selection.animal_treatments (db_path,          animal_ids=[],          ani-
                                              mal_treatments=[],  cage_treatments=[],
                                              conjunctive=True)
```

Select a dataframe of animals and all treatments including animal-level or cage-level treatments.

Parameters

- **db_path** (*str*) – Path to a LabbookDB formatted database.
- **animal_ids** (*list, optional*) – A List of LabbookDB *Animal.id* values by which to filter the query. It is faster to filter using this mechanism than to return a dataframe for all animals and then filter that.
- **animal_treatments** (*list, optional*) – A List of LabbookDB *Treatment.code* values which should be used to filter the query, while being joined to *Animal* objects.
- **cage_treatments** (*list, optional*) – A List of LabbookDB *Treatment.code* values which should be used to filter the query, while being joined to *Cage* objects - and further to *Animal* objects via *CageStay* objects. An onset check is also applied by the function, to ascertain that there is an overlap between the animal's presence in the cage and the cage treatment application.
- **conjunctive** (*bool, optional*) – Whether both *cage_treatments* and *animal_treatments* need to be satisfied (statements within each list are always disjunctive).

Notes

Currently *conjunctive=False* does not work; cage treatment and animal treatment filters are always conjunctive.

```
labbookdb.report.selection.animals_by_cage_treatment (db_path,          codes=[],
                                                       end_dates=[], start_dates=[])
```

Select a dataframe of animals and all related tables through to cage treatments based on cage_treatment filters.

Parameters

- **db_path** (*string*) –
- **to a LabbookDB formatted database.** (*Path*) –
- **codes** (*list, optional*) –
- **TreatmentProtocol.code values based on which to filter dataframe** (*Related*) –
- **end_dates** (*list, optional*) –
- **Treatment.end_date values based on which to filter dataframe** (*Related*) –

- **start_dates** (*list*, *optional*) –
- **Treatment.start_date** values based on which to filter **dataframe** (*Related*) –

`labbookdb.report.selection.animals_by_genotype (db_path, genotypes, attribute='code')`

Return *pandas.DataFrame* object containing ID and genotype table columns of animals as matched by selected values on a selected Genotype attribute field.

Parameters

- **db_path** (*string*) – Path to database file to query.
- **genotypes** (*list*) – List of strings containing values to be matched for the selected Genotype attribute.
- **attribute** (*str*) – Genotype attribute to match on.

`labbookdb.report.selection.animals_by_treatment (db_path, codes=[], end_dates=[], start_dates=[])`

Select a dataframe of animals and all related tables through to treatments based on treatment filters.

Parameters

- **db_path** (*string*) –
- **to a LabbookDB formatted database.** (*Path*) –
- **codes** (*list*, *optional*) –
- **TreatmentProtocol.code** values based on which to filter **dataframe** (*Related*) –
- **end_dates** (*list*, *optional*) –
- **Treatment.end_date** values based on which to filter **dataframe** (*Related*) –
- **start_dates** (*list*, *optional*) –
- **Treatment.start_date** values based on which to filter **dataframe** (*Related*) –

`labbookdb.report.selection.by_animals (db_path, select, animals)`

Select a dataframe of animals and all related tables through to the “select” table based on animal filters.

Parameters

- **db_path** (*str*) –
- **to a LabbookDB formatted database.** (*Path*) –
- **select** (*str*) –
- **which kind of evaluation to select dataframe.** (*For*) –
- **animals** (*list of str*) –
- **values based on which to filter dataframe** (*Animal.id*) –

`labbookdb.report.selection.cage_drinking_measurements (db_path, treatments=[])`

`labbookdb.report.selection.cage_periods (db_path, animal_filter=[], cage_filter=[])`

Return a *pandas.DataFrame* object containing the periods which animals spent in which cages.

Parameters

- **db_path** (*string*) – Path to database file to query.

- **animal_filter** (*list*, *optional*) – A list of *Animal.id* attribute values for which to specifically filter the query.

Notes

Operations on *pandas.DataFrame* objects should be performed in *labbookdb.report.tracking*, however, the cagestay end date is not explicitly recorded, so to select it or select animals by it, we calculate it here.

```
labbookdb.report.selection.parameterized(db_path, data_type, animal_filter=[],
                                         cage_filter=[], treatment_start_dates=[])
```

Select dataframe from a LabbookDB style database.

Parameters

- **db_path** (*string*) – Path to a LabbookDB formatted database.
- **data_type** (*list*) – What type of data should be selected values can be: `({"animals id", "animals info", "animals measurements", "animals measurements irregularities", "cage list", "forced swim"})`
- **animal_filter** (*list*, *optional*) – A list of animal identifiers (*Animal.id* attributes) for which to limit the query.
- **treatment_start_dates** (*list*, *optional*) – A list containing the treatment start date or dates by which to filter the cages for the sucrose preference measurements. Items should be strings in datetime format, e.g. "2016,4,25,19,30".

```
labbookdb.report.selection.stimulation_protocol(db_path, code)
```

Select a 'pandas.DataFrame' object containing all events and associated measurement units for a specific stimulation protocol.

Parameters

- **db_path** (*string*) – Path to the database file to query.
- **code** (*string*) – Code (valid *StimulationProtocol.code* value) which identifies the stimulation protocol to format.

```
labbookdb.report.selection.timetable(db_path, filters, default_join='outer', join_types=[])
```

Select a dataframe with animals as rown and all timetable-relevant events as columns.

Parameters

- **db_path** (*str*) –
- **to a LabbookDB formatted database.** (*Path*) –
- **filters** (*list of list*) –
- **list of lists giving filters for the query. It is passed to ..query.get_df()** (*A*) –
- **outerjoin_all** (*bool*) –
- **as outerjoin_all to ..query.get_df()** (*Pased*) –

labbookdb.report.tracking module

```
labbookdb.report.tracking.animal_weights(db_path, reference={}, rounding='D')
```

Return a dataframe containing animal weights and dates.

Parameters

- **db_path** (*string*) – Path to database file to query.
- **reference** (*dict*, *optional*) – Dictionary based on which to apply a reference date for the dates of each animal. Keys of this dictionary must be “animal” or “cage”, and values must be lists of treatment codes.
- **rounding** (*string*, *optional*) – Whether to round dates and timedeltas - use strings as supported by pandas. [\[1\]](#)

References

`labbookdb.report.tracking.animals_id(db_path, save_as=None)`

Extract list of animal (database and external) IDs, and either print it to screen or save it as an HTML file.

Parameters

- **db_path** (*string*) –
- **to the database file to query.** (*Path*) –
- **save_as** (*string or None*, *optional*) –
- **under which to save the HTML report (".html" is automatically appended) If None, the report is printed to the terminal.** (*Path*) –

`labbookdb.report.tracking.animals_info(db_path, save_as=None, functional_scan_responders=True, treatments=True)`

Extract list of animal (database and external) IDs and their death dates and genotypes, and either print it to screen or save it as an HTML file.

Parameters

- **db_path** (*string*) – Path to the database file to query.
- **save_as** (*string or None*, *optional*) – Path under which to save the HTML report (“.html” is automatically appended to the name, if not already present). If None, the report is printed to the terminal.
- **functional_scan_responders** (*bool*, *optional*) – Whether to create and list a column tracking how many of the scans in which an animal was exposed to stimulation show ICA results in a qualitative analysis.
- **treatments** (*bool*, *optional*) – Whether to create a and list columns tracking what animal-based and cage-based treatments the animal was subjected to.

`labbookdb.report.tracking.append_external_identifiers(db_path, df, concatenate=[])`

Append external animal IDs to a dataframe containing an *Animal_id* (*Animal.id*) column.

Parameters

- **db_path** (*str*) – Path to database file to query.
- **df** (*pandas.DataFrame*) – A *pandas.DataFrame* object containing an *Animal_id* (*Animal.id*) column.
- **concatenate** (*list*, *optional*) – A list containing any combination of ‘Animal_death_date’, ‘Genotype_id’, ‘Genotype_code’, ‘Genotype_construct’.

`labbookdb.report.tracking.bids_eventsfile(db_path, code, strict=False)`

Return a BIDS-formatted eventfile for a given code

Parameters

- **db_path** (*string*) – Path to the database file to query.
- **code** (*string*) – Code (valid *StimulationProtocol.code* value) which identifies the stimulation protocol to format.
- **strict** (*bool, optional*) – Whether to strict about respecting BIDS specifics. (currently removes coumns with only empty cells)

```
labbookdb.report.tracking.cage_consumption(db_path, df, treatment_relative_date=True,
                                           rounding='D')
```

Return a *pandas.DataFrame* object containing information about the per-animal drinking solution consumption of single cages.

Parameters

- **db_path** (*string*) – Path to the database file to query.
- **df** (*pandas.DataFrame*) – A *pandas.DataFrame* object with *DrinkingMeasurement_id*, *DrinkingMeasurement_reference_date*, *DrinkingMeasurement_date*, *DrinkingMeasurement_start_amount*, *DrinkingMeasurement_start_amount* columns. This can be obtained e.g. from *labbookdb.report.selection.cage_drinking_measurements()*.
- **treatment_relative_date** (*bool, optional*) – Whether to express the dates relative to a treatment onset. It is assumed that only one cage treatment is recorded per cage, if this is not so, this function may not work as expected.
- **rounding** (*string, optional*) – Whether to round dates and timedeltas - use strings as supported by pandas. [\[1\]](#)

Notes

This function caluclates the per-day consumption based on phase-agnostic and potentially rounded and day values.

This is prone to some inaccuracy, as drinking is generally restricted to specific times of the day. Ideally, a *waking_hour_consumption* should be estimated based on exact times of day and day cycle.

References

```
labbookdb.report.tracking.further_cages(db_path)
```

Returns cage numbers that should be selected for incoming cages.

Parameters **db_path** (*path to database file to query (needs to be protocolizer-style)*) –

```
labbookdb.report.tracking.overview(db_path, default_join=False, filters=[], join_types=[],
                                   relative_dates=True, save_as="", rounding='D', round-
                                   ing_type='round', protect_duplicates=['Animal_id',
                                   'Cage_id', 'Cage_Treatment_start_date',
                                   'Cage_TreatmentProtocol_code'])
```

Returns an overview of events per animal.

Parameters

- **db_path** (*string*) –
- **to the database file to query.** (*Path*) –
- **outerjoin_all** (*bool*) –
- **as outerjoin_all to ..query.get_df()** (*Pased*) –
- **filters** (*list of list, optional*) –

- **list of lists giving filters for the query.** It is passed to `..query.get_df()` (A)–
- **saturate** (*{list of str, list of dict}, optional*)–
- **list of dictionaries or strings specifying by which criteria to saturate cells.** It is passed to `behaviopy.timetable.multi_plot()` (A)–
- **save_df** (*string, optional*)–
- **under which to save the plotted dataframe.** ".csv" will be appended to the string, and the data will be saved in CSV format. (*Path*)–
- **window_end** (*string*)–
- **datetime-formatted string** (e.g. "2016,12,18") to apply as the timetable end date (overrides autodetected end) (A)–
- **rounding_type** (*{'round', 'floor', 'ceil'}, optional*) – Whether to round the dates (splits e.g. days apart at noon, hours at 30 minutes, etc.) or to take the floor or the ceiling.

```
labbookdb.report.tracking.qualitative_dates(df, iterator_column='Animal_id',
                                             date_column='relative_date', label=
                                             'qualitative_date', fuzzy_matching={})
```

Assign qualitative date labels.

Parameters

- **df** (*pandas.DataFrame*) – A *pandas.DataFrame* object containing a date column.
- **iteraor_column** (*string, optional*) – The label of the column which identifies the base entities of which each should be assigned a set of qualitatie dates (most commonly this is *Animal_id*, or *Cage_id*).
- **date_column** (*string, optional*) – The label of the column which serves as the quantitative record which is to be discretized into qualitative dates.
- **label** (*string, optional*) – The label to assign to the new qualitative date column.
- **fuzzy_assignment** (*dict, optional*) – A dictionary the keys of which are qualitative date labels to be assigned, and the values of which are lists giving the quantitative date labels in the order of preference based on which to assign the labels.

```
labbookdb.report.tracking.treatment_group(db_path, treatments, level="")
```

Return a *pandas.DataFrame* object containing the per animal start dates of a particular treatment code (applied either at the animal or the cage levels).

Parameters

- **db_path** (*string*) – Path to database file to query.
- **code** (*string*) – Desired treatment code (*Treatment.code* attribute) to filter for.
- **level** (*{ "animal", "cage" }*) – Whether to query animal treatments or cage treatments.

Notes

This function checks whether cage-level treatment onsets indeed happened during the period in which the animal was housed in teh cage. We do not check for the treatment end dates, as an animal which has received a partial

treatment has received a treatment. Checks for treatment discontinuation due to e.g. death should be performed elsewhere.

labbookdb.report.utilities module

`labbookdb.report.utilities.collapse_rename(df, groupby, collapse, rename=False)`

Collapse long form columns according to a lambda function, so that groupby column values are rendered unique

Parameters

- **df** (*pandas.DataFrame*) – A *pandas.DataFrame* object which you want to collapse.
- **groupby** (*string*) – The name of a column from *df*, the values of which you want to render unique.
- **collapse** (*dict*) – A dictionary the keys of which are columns you want to collapse, and the values of which are lambda functions instructing how to collapse (e.g. concatenate) the values.
- **rename** (*dict, optional*) – A dictionary the keys of which are names of columns from *df*, and the values of which are new names for these columns.

`labbookdb.report.utilities.concurrent_cagetreatment(df, cagestays, protect_duplicates=['Animal_id', 'Cage_id', 'Cage_Treatment_start_date', 'Cage_Treatment_end_date', 'Cage_TreatmentProtocol_code', 'Treatment_end_date', 'Treatment_end_date', 'TreatmentProtocol_code'])`

Return a *pandas.DataFrame* object containing only *Cage_Treatment** entries which are concurrent with the animal stay in the cage to which they were administered.

Parameters

- **df** (*pandas.DataFrame*) –
Pandas Dataframe, with columns containing: *Animal_id, Animal_death_date, CageStay_start_date, Cage_Treatment_start_date, Cage_TreatmentProtocol_code.*
- **cagestays** (*pandas.DataFrame*) –
Pandas Dataframe, with columns containing: *Animal_id, CageStay_end_date, CageStay_start_date,*

Notes

This function checks whether cage-level treatment onsets indeed happened during the period in which the animal was housed in the cage. We do not check for the treatment end dates, as an animal which has received a partial treatment has received a treatment. Checks for treatment discontinuation due to e.g. death should be performed elsewhere.

`labbookdb.report.utilities.make_identifier_short_form(df, in-dex_name='Animal_id')`

Convert the long form *AnimalExternalIdentifier_identifier* column of a *pandas.DataFrame* to short-form identifier columns named after the corresponding values on the *AnimalExternalIdentifier_database* column.

Parameters

- **df** (*pandas.DataFrame*) – A *pandas.DataFrame* object containing a long-form *AnimalExternalIdentifier_identifier* column and a dedicated *AnimalExternalIdentifier_database* column.
- **index_name** (*str*, *optional*) – The name of a column from *df*, the values of which can be rendered unique. This column will serve as the index of the resulting dataframe.

```
labbookdb.report.utilities.relativize_dates(df,          date_suffix='_date',          round-  
                                             ing='D',    rounding_type='round',    refer-  
                                             ence_date=True)
```

Express dates on each row of a Pandas Dataframe as datetime objects relative to the row value on the 'reference_date' column.

Parameters

- **df** (*pandas.DataFrame*) – Pandas Dataframe, with columns containing 'reference_date' and strings ending in *date_suffix*.
- **date_suffix** (*str*, *optional*) – String suffix via which to identify date columns needing manipulation.
- **rounding** (*str*, *optional*) – Datetime increment for date rounding.
- **rounding_type** (*{'round', 'floor', 'ceil'}*, *optional*) – Whether to round the dates (splits e.g. days apart at noon, hours at 30 minutes, etc.) or to take the floor or the ceiling.

Module contents

labbookdb.tests package

Submodules

labbookdb.tests.test_basic module

```
labbookdb.tests.test_basic.test_load()
```

labbookdb.tests.test_query module

labbookdb.tests.test_report module

labbookdb.tests.test_selection module

Module contents

Submodules

labbookdb.cli module

```
labbookdb.cli.main()
```

labbookdb.decorators module

`labbookdb.decorators.environment_db_path()`

Add a default value to the *db_path* positional argument, based on the *LDB_PATH* environment variable, and fail elegantly if not.

Module contents

4.3.2 Objects Reference

I

- labbookdb, 39
- labbookdb.cli, 38
- labbookdb.db, 28
- labbookdb.db.add, 10
- labbookdb.db.base_classes, 12
- labbookdb.db.common_classes, 15
- labbookdb.db.query, 26
- labbookdb.db.utils, 28
- labbookdb.decorators, 39
- labbookdb.evaluate, 28
- labbookdb.introspection, 28
- labbookdb.report, 38
- labbookdb.report.development, 29
- labbookdb.report.formatting, 29
- labbookdb.report.processing, 29
- labbookdb.report.protocolize, 30
- labbookdb.report.selection, 30
- labbookdb.report.tracking, 33
- labbookdb.report.utilities, 37
- labbookdb.tests, 38
- labbookdb.tests.test_basic, 38

A

`add_all_columns()` (in module `labbookdb.db.query`), 26
`add_generic()` (in module `labbookdb.db.add`), 10
`add_to_db()` (in module `labbookdb.db.add`), 10
`addgene_identifier` (`labbookdb.db.base_classes.Virus` attribute), 15
`affiliation` (`labbookdb.db.base_classes.Operator` attribute), 13
`amount` (`labbookdb.db.common_classes.VirusInjectionProtocol` attribute), 25
`anesthesia` (`labbookdb.db.common_classes.FMRIMeasurement` attribute), 19
`anesthesia` (`labbookdb.db.common_classes.Operation` attribute), 22
`anesthesia_id` (`labbookdb.db.common_classes.FMRIMeasurement` attribute), 19
`anesthesia_id` (`labbookdb.db.common_classes.Operation` attribute), 22
`AnesthesiaProtocol` (class in `labbookdb.db.common_classes`), 15
`angle` (`labbookdb.db.base_classes.OpticFiberImplant` attribute), 13
`Animal` (class in `labbookdb.db.common_classes`), 15
`animal_id` (`labbookdb.db.base_classes.Biopsy` attribute), 12
`animal_id` (`labbookdb.db.base_classes.Measurement` attribute), 13
`animal_id` (`labbookdb.db.common_classes.AnimalExternalIdentifier` attribute), 16
`animal_id` (`labbookdb.db.common_classes.BrainBiopsy` attribute), 16
`animal_id` (`labbookdb.db.common_classes.DrinkingMeasurement` attribute), 19
`animal_id` (`labbookdb.db.common_classes.FluorescentMicroscopyMeasurement` attribute), 20
`animal_id` (`labbookdb.db.common_classes.FMRIMeasurement` attribute), 19
`animal_id` (`labbookdb.db.common_classes.ForcedSwimTestMeasurement` attribute), 20
`animal_id` (`labbookdb.db.common_classes.Observation` attribute), 22
`animal_id` (`labbookdb.db.common_classes.OpenFieldTestMeasurement` attribute), 22
`animal_id` (`labbookdb.db.common_classes.Operation` attribute), 23
`animal_id` (`labbookdb.db.common_classes.SucrosePreferenceMeasurement` attribute), 24
`animal_id` (`labbookdb.db.common_classes.WeightMeasurement` attribute), 26
`animal_id()` (in module `labbookdb.report.selection`), 30
`animal_info()` (in module `labbookdb.db.query`), 26
`animal_multiselect()` (in module `labbookdb.report.development`), 29
`animal_operations()` (in module `labbookdb.report.selection`), 30
`animal_treatments()` (in module `labbookdb.report.selection`), 31
`animal_weights()` (in module `labbookdb.report.tracking`), 33
`animal_weights_()` (in module `labbookdb.report.development`), 29
`AnimalExternalIdentifier` (class in `labbookdb.db.common_classes`), 16
`animals_by_cage_treatment()` (in module `labbookdb.report.selection`), 31
`animals_by_genotype()` (in module `labbookdb.report.selection`), 32
`animals_by_treatment()` (in module `labbookdb.report.selection`), 32
`animals_id()` (in module `labbookdb.report.tracking`), 34
`animals_info()` (in module `labbookdb.report.tracking`), 34
`append_external_identifiers()` (in module

labbookdb.report.tracking), 34
 append_parameter() (in module lab-
 bookdb.db.add), 10
 arange_by_date() (in module labbookdb.db.utils),
 28
 Arena (class in labbookdb.db.common_classes), 16
 arena_id (labbookdb.db.common_classes.OpenFieldTestMeasurement
 attribute), 22
 author (labbookdb.db.common_classes.Evaluation at-
 tribute), 19
 author_id (labbookdb.db.common_classes.Evaluation
 attribute), 19
 authors (labbookdb.db.base_classes.Protocol at-
 tribute), 14
 authors (labbookdb.db.common_classes.AnesthesiaProtocol
 attribute), 15
 authors (labbookdb.db.common_classes.BrainExtractionProtocol
 attribute), 17
 authors (labbookdb.db.common_classes.DNAExtractionProtocol
 attribute), 18
 authors (labbookdb.db.common_classes.HandlingHabituationProtocol
 attribute), 21
 authors (labbookdb.db.common_classes.OpticFiberImplantProtocol
 attribute), 23
 authors (labbookdb.db.common_classes.SectioningProtocol
 attribute), 23
 authors (labbookdb.db.common_classes.TreatmentProtocol
 attribute), 25
 authors (labbookdb.db.common_classes.VirusInjectionProtocol
 attribute), 25

B

behaviour (labbookdb.db.common_classes.Observation
 attribute), 22
 bids_eventsfile() (in module lab-
 bookdb.report.tracking), 34
 biopsies (labbookdb.db.common_classes.Animal at-
 tribute), 15
 Biopsy (class in labbookdb.db.base_classes), 12
 biopsy_id (labbookdb.db.common_classes.FluorescentMicroscopyMeasurement
 attribute), 20
 birth_date (labbookdb.db.common_classes.Animal
 attribute), 15
 blade_frequency (lab-
 bookdb.db.common_classes.SectioningProtocol
 attribute), 23
 blade_speed (labbookdb.db.common_classes.SectioningProtocol
 attribute), 23
 bolus (labbookdb.db.common_classes.AnesthesiaProtocol
 attribute), 15
 bolus_to_maintenance_delay (lab-
 bookdb.db.common_classes.AnesthesiaProtocol
 attribute), 15

BrainBiopsy (class in lab-
 bookdb.db.common_classes), 16
 BrainExtractionProtocol (class in lab-
 bookdb.db.common_classes), 17
 by_animals() (in module lab-
 bookdb.report.selection), 32

C

Cage (class in labbookdb.db.common_classes), 17
 cage (labbookdb.db.common_classes.CageStay at-
 tribute), 18
 cage (labbookdb.db.common_classes.HandlingHabituation
 attribute), 21
 cage_consumption() (in module lab-
 bookdb.report.tracking), 35
 cage_drinking_measurements() (in module
 labbookdb.report.selection), 32
 cage_id (labbookdb.db.base_classes.Measurement at-
 tribute), 13
 cage_id (labbookdb.db.common_classes.CageStay at-
 tribute), 18
 cage_id (labbookdb.db.common_classes.DrinkingMeasurement
 attribute), 19
 cage_id (labbookdb.db.common_classes.FluorescentMicroscopyMeasurement
 attribute), 20
 cage_id (labbookdb.db.common_classes.FMRIMeasurement
 attribute), 19
 cage_id (labbookdb.db.common_classes.ForcedSwimTestMeasurement
 attribute), 20
 cage_id (labbookdb.db.common_classes.HandlingHabituation
 attribute), 21
 cage_id (labbookdb.db.common_classes.OpenFieldTestMeasurement
 attribute), 22
 cage_id (labbookdb.db.common_classes.SucrosePreferenceMeasurement
 attribute), 24
 cage_id (labbookdb.db.common_classes.WeightMeasurement
 attribute), 26
 cage_info() (in module labbookdb.db.query), 26
 cage_periods() (in module lab-
 bookdb.report.selection), 32
 cage_stays (labbookdb.db.common_classes.Animal
 attribute), 15
 CageStay (class in labbookdb.db.common_classes), 17
 cannula_diameter (lab-
 bookdb.db.base_classes.OpticFiberImplant
 attribute), 14
 cannula_id (labbookdb.db.base_classes.Virus attribute), 15
 center_luminostiy (lab-
 bookdb.db.common_classes.OpenFieldTestMeasurement
 attribute), 22
 centrifugation (lab-
 bookdb.db.common_classes.DNAExtractionProtocol
 attribute), 18

centrifugation_id (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 15

code (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 18

code (labbookdb.db.common_classes.Ingredient attribute), 18

code (labbookdb.db.base_classes.FMRIScannerSetup attribute), 12

code (labbookdb.db.common_classes.Substance attribute), 24

code (labbookdb.db.base_classes.Genotype attribute), 13

code (labbookdb.db.common_classes.Substance attribute), 24

code (labbookdb.db.base_classes.MeasurementUnit attribute), 13

code (labbookdb.db.common_classes.Ingredient attribute), 21

code (labbookdb.db.base_classes.Operator attribute), 13

code (labbookdb.db.common_classes.Substance attribute), 24

code (labbookdb.db.base_classes.OpticFiberImplant attribute), 14

code (labbookdb.db.common_classes.Substance attribute), 24

code (labbookdb.db.base_classes.OrthogonalStereotacticTarget attribute), 14

code (labbookdb.db.common_classes.SucrosePreferenceMeasurement attribute), 24

code (labbookdb.db.base_classes.Protocol attribute), 14

code (labbookdb.db.common_classes.SucrosePreferenceMeasurement attribute), 24

code (labbookdb.db.base_classes.Virus attribute), 15

code (labbookdb.db.common_classes.Ingredient attribute), 22

code (labbookdb.db.common_classes.AnesthesiaProtocol attribute), 15

code (labbookdb.db.common_classes.Ingredient attribute), 22

code (labbookdb.db.common_classes.Arena attribute), 16

code (labbookdb.db.common_classes.Substance attribute), 24

code (labbookdb.db.common_classes.BrainExtractionProtocol attribute), 17

code (labbookdb.db.common_classes.Substance attribute), 24

code (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 18

code (labbookdb.db.common_classes.SucrosePreferenceMeasurement attribute), 24

code (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 18

code (labbookdb.db.common_classes.SucrosePreferenceMeasurement attribute), 24

code (labbookdb.db.common_classes.HandlingHabituationProtocol attribute), 21

code (labbookdb.db.common_classes.SucrosePreferenceMeasurement attribute), 24

code (labbookdb.db.common_classes.OpticFiberImplantProtocol attribute), 23

code (labbookdb.db.common_classes.SucrosePreferenceMeasurement attribute), 19

code (labbookdb.db.common_classes.SectioningProtocol attribute), 23

code (labbookdb.db.common_classes.Solution attribute), 24

code (labbookdb.db.common_classes.Solution attribute), 23

code (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 18

code (labbookdb.db.common_classes.StimulationProtocol attribute), 24

code (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 18

code (labbookdb.db.common_classes.Substance attribute), 24

code (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 18

code (labbookdb.db.common_classes.Substance attribute), 24

code (labbookdb.db.common_classes.OpenFieldTestMeasurement attribute), 22

code (labbookdb.db.common_classes.TreatmentProtocol attribute), 25

code (labbookdb.db.common_classes.Virus attribute), 15

code (labbookdb.db.common_classes.VirusInjectionProtocol attribute), 25

coil (labbookdb.db.base_classes.FMRIScannerSetup attribute), 12

collapse_rename() (in module labbookdb.report.utilities), 37

commit_and_close() (in module labbookdb.db.add), 11

commit_and_close() (in module labbookdb.db.query), 27

compose_tex() (in module labbookdb.report.protocolize), 30

concentration (labbookdb.db.base_classes.Virus attribute), 15

concentration (labbookdb.db.common_classes.Ingredient attribute), 21

concentration (labbookdb.db.common_classes.Substance attribute), 24

concentration_unit (labbookdb.db.common_classes.Ingredient attribute), 21

concentration_unit (labbookdb.db.common_classes.Substance attribute), 24

concentration_unit (labbookdb.db.common_classes.SucrosePreferenceMeasurement attribute), 24

concentration_unit_id (labbookdb.db.common_classes.Ingredient attribute), 22

concentration_unit_id (labbookdb.db.common_classes.Substance attribute), 24

concentration_unit_id (labbookdb.db.common_classes.SucrosePreferenceMeasurement attribute), 24

concurrent_cagetreatment() (in module labbookdb.report.utilities), 37

product (labbookdb.db.base_classes.Genotype attribute), 13

assumption (labbookdb.db.common_classes.DrinkingMeasurement attribute), 19

contains (labbookdb.db.common_classes.Solution attribute), 24

cooling (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 18

cooling_id (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 18

corner_luminostiy (labbookdb.db.common_classes.OpenFieldTestMeasurement attribute), 22

credit (labbookdb.db.base_classes.Virus attribute), 15

D

data (labbookdb.db.common_classes.FluorescentMicroscopyMeasurement attribute), 20

data_path (labbookdb.db.common_classes.BrainBiopsy attribute), 16

data_path (labbookdb.db.common_classes.FMRIMeasurement attribute), 19

data_path (labbookdb.db.common_classes.ForcedSwimTestMeasurement attribute), 20

data_path (labbookdb.db.common_classes.OpenFieldTestMeasurement attribute), 22

database (*labbookdb.db.common_classes.AnimalExternalIdentifier* attribute), 25
 attribute), 16
 dose_unit_id (*labbookdb.db.common_classes.TreatmentProtocol* attribute), 25
 date (*labbookdb.db.base_classes.Measurement* attribute), 13
 date (*labbookdb.db.common_classes.DrinkingMeasurement* attribute), 19
 date (*labbookdb.db.common_classes.FluorescentMicroscopyMeasurement* attribute), 20
 date (*labbookdb.db.common_classes.FMRIMeasurement* attribute), 20
 date (*labbookdb.db.common_classes.ForcedSwimTestMeasurement* attribute), 14
 date (*labbookdb.db.common_classes.HandlingHabituation* attribute), 21
 date (*labbookdb.db.common_classes.Observation* attribute), 22
 date (*labbookdb.db.common_classes.OpenFieldTestMeasurement* attribute), 22
 date (*labbookdb.db.common_classes.Operation* attribute), 23
 date (*labbookdb.db.common_classes.SucrosePreferenceMeasurement* attribute), 24
 date (*labbookdb.db.common_classes.WeightMeasurement* attribute), 26
 death_date (*labbookdb.db.common_classes.Animal* attribute), 15
 death_reason (*labbookdb.db.common_classes.Animal* attribute), 16
 depth (*labbookdb.db.base_classes.OrthogonalStereotacticTarget* attribute), 14
 description (*labbookdb.db.base_classes.Irregularity* attribute), 13
 digestion (*labbookdb.db.common_classes.DNAExtractionProtocol* attribute), 18
 digestion_buffer (*labbookdb.db.common_classes.DNAExtractionProtocol* attribute), 18
 digestion_buffer_id (*labbookdb.db.common_classes.DNAExtractionProtocol* attribute), 18
 digestion_buffer_volume (*labbookdb.db.common_classes.DNAExtractionProtocol* attribute), 18
 digestion_id (*labbookdb.db.common_classes.DNAExtractionProtocol* attribute), 18
 DNAExtraction (class in *labbookdb.db.common_classes*), 18
 DNAExtractionProtocol (class in *labbookdb.db.common_classes*), 18
 dose (*labbookdb.db.common_classes.TreatmentProtocol* attribute), 25
 dose_unit (*labbookdb.db.common_classes.TreatmentProtocol* attribute), 16
 dose_unit_id (*labbookdb.db.common_classes.TreatmentProtocol* attribute), 25
 drinking_water_by_cage_treatment() (in module *labbookdb.report.development*), 29
 PyMeasurement (class in *labbookdb.db.common_classes*), 19
 dt_format() (in module *labbookdb.db.utils*), 28
 duration (*labbookdb.db.base_classes.StimulationEvent* attribute), 14
 duration (*labbookdb.db.common_classes.Incubation* attribute), 21
 duration_unit (*labbookdb.db.common_classes.Incubation* attribute), 21
 incubation_unit_id (*labbookdb.db.common_classes.Incubation* attribute), 21
 ear_punches (*labbookdb.db.common_classes.Animal* attribute), 16
 edge_luminostiy (*labbookdb.db.common_classes.OpenFieldTestMeasurement* attribute), 22
 email (*labbookdb.db.base_classes.Operator* attribute), 13
 end_amount (*labbookdb.db.common_classes.DrinkingMeasurement* attribute), 19
 end_date (*labbookdb.db.common_classes.Treatment* attribute), 25
 environment_db_path() (in module *labbookdb.decorators*), 39
 environmental_enrichment (*labbookdb.db.common_classes.Cage* attribute), 17
 Evaluation (class in *labbookdb.db.common_classes*), 19
 evaluations (*labbookdb.db.common_classes.ForcedSwimTestMeasurement* attribute), 20
 evaluations (*labbookdb.db.common_classes.OpenFieldTestMeasurement* attribute), 22
 events (*labbookdb.db.common_classes.StimulationProtocol* attribute), 24
 exposure (*labbookdb.db.common_classes.FluorescentMicroscopyMeasurement* attribute), 20
 external_ids (*labbookdb.db.common_classes.Animal* attribute), 16
 extraction_protocol (*labbookdb.db.base_classes.Biopsy* attribute), 12

extraction_protocol	(lab- bookdb.db.common_classes.BrainBiopsy attribute), 16	G	Genotype (class in labbookdb.db.base_classes), 13
extraction_protocol_id	(lab- bookdb.db.base_classes.Biopsy attribute), 12		genotypes (labbookdb.db.common_classes.Animal attribute), 16
extraction_protocol_id	(lab- bookdb.db.common_classes.BrainBiopsy attribute), 16		get_df() (in module labbookdb.db.query), 27
			get_for_protocolize() (in module lab- bookdb.db.query), 28
			get_related_id() (in module labbookdb.db.query), 28
			get_related_ids() (in module labbookdb.db.add), 11
F			group_picking_up (lab- bookdb.db.common_classes.HandlingHabituationProtocol attribute), 21
ferrule_diameter	(lab- bookdb.db.base_classes.OpticFiberImplant attribute), 14		
fixation_solution	(lab- bookdb.db.common_classes.BrainExtractionProtocol attribute), 17	H	handling_habituations (lab- bookdb.db.common_classes.Cage attribute), 17
fixation_solution_id	(lab- bookdb.db.common_classes.BrainExtractionProtocol attribute), 17		HandlingHabituation (class in lab- bookdb.db.common_classes), 21
fixation_solution_volume	(lab- bookdb.db.common_classes.BrainExtractionProtocol attribute), 17		handlingHabituationProtocol (class in lab- bookdb.db.common_classes), 21
fluorescent_microscopy	(lab- bookdb.db.base_classes.Biopsy attribute), 12	I	
fluorescent_microscopy	(lab- bookdb.db.common_classes.BrainBiopsy attribute), 16		id (labbookdb.db.base_classes.Biopsy attribute), 12
FluorescentMicroscopyMeasurement (class in labbookdb.db.common_classes), 20			id (labbookdb.db.base_classes.FMRIScannerSetup at- tribute), 12
flushing_solution	(lab- bookdb.db.common_classes.BrainExtractionProtocol attribute), 17		id (labbookdb.db.base_classes.Genotype attribute), 13
flushing_solution_id	(lab- bookdb.db.common_classes.BrainExtractionProtocol attribute), 17		id (labbookdb.db.base_classes.Irregularity attribute), 13
flushing_solution_volume	(lab- bookdb.db.common_classes.BrainExtractionProtocol attribute), 17		id (labbookdb.db.base_classes.Measurement attribute), 13
FMRIMeasurement (class in lab- bookdb.db.common_classes), 19			id (labbookdb.db.base_classes.MeasurementUnit at- tribute), 13
FMRIScannerSetup (class in lab- bookdb.db.base_classes), 12			id (labbookdb.db.base_classes.Operator attribute), 13
ForcedSwimTestMeasurement (class in lab- bookdb.db.common_classes), 20			id (labbookdb.db.base_classes.OpticFiberImplant at- tribute), 14
frequency (labbookdb.db.base_classes.StimulationEvent attribute), 14			id (labbookdb.db.base_classes.OrthogonalStereotacticTarget attribute), 14
frequency (labbookdb.db.common_classes.TreatmentProtocol attribute), 25			id (labbookdb.db.base_classes.Protocol attribute), 14
full_name (labbookdb.db.base_classes.Operator at- tribute), 13			id (labbookdb.db.base_classes.StimulationEvent at- tribute), 14
further_cages() (in module lab- bookdb.report.tracking), 35			id (labbookdb.db.base_classes.Virus attribute), 15
			id (labbookdb.db.common_classes.AnesthesiaProtocol attribute), 15
			id (labbookdb.db.common_classes.Animal attribute), 16
			id (labbookdb.db.common_classes.AnimalExternalIdentifier attribute), 16
			id (labbookdb.db.common_classes.Arena attribute), 16
			id (labbookdb.db.common_classes.BrainBiopsy at- tribute), 16
			id (labbookdb.db.common_classes.BrainExtractionProtocol attribute), 17
			id (labbookdb.db.common_classes.Cage attribute), 17

[id \(labbookdb.db.common_classes.CageStay attribute\), 18](#)
[id \(labbookdb.db.common_classes.DNAExtraction attribute\), 18](#)
[id \(labbookdb.db.common_classes.DNAExtractionProtocol attribute\), 18](#)
[id \(labbookdb.db.common_classes.DrinkingMeasurement attribute\), 19](#)
[id \(labbookdb.db.common_classes.Evaluation attribute\), 19](#)
[id \(labbookdb.db.common_classes.FluorescentMicroscopyMeasurement attribute\), 20](#)
[id \(labbookdb.db.common_classes.FMRIMeasurement attribute\), 20](#)
[id \(labbookdb.db.common_classes.ForcedSwimTestMeasurement attribute\), 20](#)
[id \(labbookdb.db.common_classes.HandlingHabituation attribute\), 21](#)
[id \(labbookdb.db.common_classes.HandlingHabituationProtocol attribute\), 21](#)
[id \(labbookdb.db.common_classes.Incubation attribute\), 21](#)
[id \(labbookdb.db.common_classes.Ingredient attribute\), 22](#)
[id \(labbookdb.db.common_classes.Observation attribute\), 22](#)
[id \(labbookdb.db.common_classes.OpenFieldTestMeasurement attribute\), 22](#)
[id \(labbookdb.db.common_classes.Operation attribute\), 23](#)
[id \(labbookdb.db.common_classes.OpticFiberImplantProtocol attribute\), 23](#)
[id \(labbookdb.db.common_classes.SectioningProtocol attribute\), 23](#)
[id \(labbookdb.db.common_classes.Solution attribute\), 24](#)
[id \(labbookdb.db.common_classes.StimulationProtocol attribute\), 24](#)
[id \(labbookdb.db.common_classes.Substance attribute\), 24](#)
[id \(labbookdb.db.common_classes.SucrosePreferenceMeasurement attribute\), 24](#)
[id \(labbookdb.db.common_classes.Treatment attribute\), 25](#)
[id \(labbookdb.db.common_classes.TreatmentProtocol attribute\), 25](#)
[id \(labbookdb.db.common_classes.VirusInjectionProtocol attribute\), 25](#)
[id \(labbookdb.db.common_classes.WeightMeasurement attribute\), 26](#)
[id_local \(labbookdb.db.common_classes.Cage attribute\), 17](#)
[identifier \(labbookdb.db.common_classes.AnimalExternalIdentifier attribute\), 16](#)
[imaged_wavelength \(labbookdb.db.common_classes.FluorescentMicroscopyMeasurement attribute\), 20](#)
[inactivation \(labbookdb.db.common_classes.DNAExtractionProtocol attribute\), 18](#)
[inactivation_id \(labbookdb.db.common_classes.DNAExtractionProtocol attribute\), 18](#)
[Incubation \(class in labbookdb.db.common_classes\), 21](#)
[individual_picking_up \(labbookdb.db.common_classes.HandlingHabituationProtocol attribute\), 21](#)
[injection \(labbookdb.db.common_classes.AnesthesiaProtocol attribute\), 15](#)
[Ingredient \(class in labbookdb.db.common_classes\), 21](#)
[Instructions\(\) \(in module labbookdb.db.add\), 12](#)
[irregularities \(labbookdb.db.base_classes.Measurement attribute\), 13](#)
[irregularities \(labbookdb.db.common_classes.DrinkingMeasurement attribute\), 19](#)
[irregularities \(labbookdb.db.common_classes.FluorescentMicroscopyMeasurement attribute\), 20](#)
[irregularities \(labbookdb.db.common_classes.FMRIMeasurement attribute\), 20](#)
[irregularities \(labbookdb.db.common_classes.ForcedSwimTestMeasurement attribute\), 20](#)
[irregularities \(labbookdb.db.common_classes.OpenFieldTestMeasurement attribute\), 22](#)
[irregularities \(labbookdb.db.common_classes.Operation attribute\), 23](#)
[irregularities \(labbookdb.db.common_classes.SucrosePreferenceMeasurement attribute\), 24](#)
[irregularities \(labbookdb.db.common_classes.WeightMeasurement attribute\), 26](#)
[Irregularity \(class in labbookdb.db.base_classes\), 13](#)
[L](#)
[labbookdb \(module\), 39](#)
[labbookdb.cli \(module\), 38](#)
[labbookdb.db \(module\), 28](#)
[labbookdb.db.add \(module\), 10](#)

labbookdb.db.base_classes (module), 12
 labbookdb.db.common_classes (module), 15
 labbookdb.db.query (module), 26
 labbookdb.db.utils (module), 28
 labbookdb.decorators (module), 39
 labbookdb.evaluate (module), 28
 labbookdb.introspection (module), 28
 labbookdb.report (module), 38
 labbookdb.report.development (module), 29
 labbookdb.report.formatting (module), 29
 labbookdb.report.processing (module), 29
 labbookdb.report.protocolize (module), 30
 labbookdb.report.selection (module), 30
 labbookdb.report.tracking (module), 33
 labbookdb.report.utilities (module), 37
 labbookdb.tests (module), 38
 labbookdb.tests.test_basic (module), 38
 leftright (labbookdb.db.base_classes.OrthogonalStereoImageTarget attribute), 14
 length (labbookdb.db.base_classes.OpticFiberImplant attribute), 14
 license (labbookdb.db.common_classes.Animal attribute), 16
 light_source (labbookdb.db.common_classes.FluorescentMicroscopyMeasurement attribute), 20
 load_session() (in module labbookdb.db.add), 12
 load_session() (in module labbookdb.db.query), 28
 location (labbookdb.db.common_classes.Cage attribute), 17
 long_code (labbookdb.db.base_classes.OpticFiberImplant attribute), 14
 long_name (labbookdb.db.base_classes.MeasurementUnit attribute), 13
 long_name (labbookdb.db.common_classes.Substance attribute), 24
 lysis (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 18
 lysis_buffer (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 18
 lysis_buffer_id (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 18
 lysis_buffer_volume (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 18
 lysis_id (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 18
 make_identifier_short_form() (in module labbookdb.report.utilities), 37
 manufacturer (labbookdb.db.base_classes.OpticFiberImplant attribute), 14
 manufacturer_code (labbookdb.db.base_classes.OpticFiberImplant attribute), 14
 mass_unit (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 18
 mass_unit_id (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 19
 maximal_severity (labbookdb.db.common_classes.Animal attribute), 16
 Measurement (class in labbookdb.db.base_classes), 13
 measurement_id (labbookdb.db.common_classes.Evaluation attribute), 19
 measurements (labbookdb.db.common_classes.Animal attribute), 16
 measurements (labbookdb.db.common_classes.Arena attribute), 16
 measurements (labbookdb.db.common_classes.Cage attribute), 17
 MeasurementUnit (class in labbookdb.db.base_classes), 13
 movement (labbookdb.db.common_classes.Incubation attribute), 21
N
 name (labbookdb.db.base_classes.Protocol attribute), 14
 name (labbookdb.db.common_classes.AnesthesiaProtocol attribute), 15
 name (labbookdb.db.common_classes.Arena attribute), 16
 name (labbookdb.db.common_classes.BrainExtractionProtocol attribute), 17
 name (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 19
 name (labbookdb.db.common_classes.HandlingHabituationProtocol attribute), 21
 name (labbookdb.db.common_classes.OpticFiberImplantProtocol attribute), 23
 name (labbookdb.db.common_classes.SectioningProtocol attribute), 23
 name (labbookdb.db.common_classes.Solution attribute), 24
 name (labbookdb.db.common_classes.StimulationProtocol attribute), 24
 name (labbookdb.db.common_classes.Substance attribute), 24

M

main() (in module labbookdb.cli), 38
 maintenance (labbookdb.db.common_classes.AnesthesiaProtocol attribute), 15

name (*labbookdb.db.common_classes.TreatmentProtocol* attribute), 25

name (*labbookdb.db.common_classes.VirusInjectionProtocol* attribute), 25

numerical_aperture (*labbookdb.db.base_classes.OpticFiberImplant* attribute), 14

O

Observation (class in *labbookdb.db.common_classes*), 22

observations (*labbookdb.db.common_classes.Animal* attribute), 16

onset (*labbookdb.db.base_classes.StimulationEvent* attribute), 14

OpenFieldTestMeasurement (class in *labbookdb.db.common_classes*), 22

Operation (class in *labbookdb.db.common_classes*), 22

operations (*labbookdb.db.common_classes.Animal* attribute), 16

Operator (class in *labbookdb.db.base_classes*), 13

operator (*labbookdb.db.base_classes.Measurement* attribute), 13

operator (*labbookdb.db.common_classes.DrinkingMeasurement* attribute), 19

operator (*labbookdb.db.common_classes.FluorescentMicroscopyMeasurement* attribute), 20

operator (*labbookdb.db.common_classes.FMRIMeasurement* attribute), 20

operator (*labbookdb.db.common_classes.ForcedSwimTestMeasurement* attribute), 20

operator (*labbookdb.db.common_classes.Observation* attribute), 22

operator (*labbookdb.db.common_classes.OpenFieldTestMeasurement* attribute), 22

operator (*labbookdb.db.common_classes.Operation* attribute), 23

operator (*labbookdb.db.common_classes.SucrosePreferenceMeasurement* attribute), 24

operator (*labbookdb.db.common_classes.WeightMeasurement* attribute), 26

operator_id (*labbookdb.db.base_classes.Measurement* attribute), 13

operator_id (*labbookdb.db.common_classes.DrinkingMeasurement* attribute), 19

operator_id (*labbookdb.db.common_classes.FluorescentMicroscopyMeasurement* attribute), 20

operator_id (*labbookdb.db.common_classes.FMRIMeasurement* attribute), 20

operator_id (*labbookdb.db.common_classes.ForcedSwimTestMeasurement* attribute), 20

operator_id (*labbookdb.db.common_classes.Observation* attribute), 22

operator_id (*labbookdb.db.common_classes.OpenFieldTestMeasurement* attribute), 22

operator_id (*labbookdb.db.common_classes.Operation* attribute), 23

operator_id (*labbookdb.db.common_classes.SucrosePreferenceMeasurement* attribute), 24

operator_id (*labbookdb.db.common_classes.WeightMeasurement* attribute), 26

optic_fiber_implant (*labbookdb.db.common_classes.OpticFiberImplantProtocol* attribute), 23

optic_fiber_implant_id (*labbookdb.db.common_classes.OpticFiberImplantProtocol* attribute), 23

OpticFiberImplant (class in *labbookdb.db.base_classes*), 13

OpticFiberImplantProtocol (class in *labbookdb.db.common_classes*), 23

OrthogonalStereotacticTarget (class in *labbookdb.db.base_classes*), 14

overview() (in module *labbookdb.report.tracking*), 35

P

parameterized() (in module *labbookdb.report.selection*), 33

patch (*labbookdb.db.common_classes.Evaluation* attribute), 19

perfusion_flow (*labbookdb.db.common_classes.BrainExtractionProtocol* attribute), 17

perfusion_system (*labbookdb.db.common_classes.BrainExtractionProtocol* attribute), 17

physiology (*labbookdb.db.common_classes.Observation* attribute), 22

pitch (*labbookdb.db.base_classes.OrthogonalStereotacticTarget* attribute), 14

plasmid_summary (*labbookdb.db.base_classes.Virus* attribute), 15

plottable_sucrosepreference_df() (in module *labbookdb.report.formatting*), 29

plottable_sums() (in module *labbookdb.report.formatting*), 29

post_extraction_fixation_time (*labbookdb.db.common_classes.BrainExtractionProtocol* attribute), 17

post_extraction_fixation_time (*labbookdb.db.base_classes.OrthogonalStereotacticTarget* attribute), 14

[print_document\(\)](#) (in module `labbookdb.report.protocolize`), 30
[protease](#) (`labbookdb.db.common_classes.DNAExtractionProtocol` attribute), 19
[protease_id](#) (`labbookdb.db.common_classes.DNAExtractionProtocol` attribute), 19
[protease_volume](#) (`labbookdb.db.common_classes.DNAExtractionProtocol` attribute), 19
[Protocol](#) (class in `labbookdb.db.base_classes`), 14
[protocol](#) (`labbookdb.db.common_classes.DNAExtractionProtocol` attribute), 18
[protocol](#) (`labbookdb.db.common_classes.HandlingHabituationProtocol` attribute), 21
[protocol](#) (`labbookdb.db.common_classes.TreatmentProtocol` attribute), 25
[protocol_id](#) (`labbookdb.db.common_classes.DNAExtractionProtocol` attribute), 18
[protocol_id](#) (`labbookdb.db.common_classes.HandlingHabituationProtocol` attribute), 21
[protocol_id](#) (`labbookdb.db.common_classes.TreatmentProtocol` attribute), 25
[protocols](#) (`labbookdb.db.common_classes.Operation` attribute), 23
[pubchem_sid](#) (`labbookdb.db.common_classes.Substance` attribute), 24
[pulse_width](#) (`labbookdb.db.base_classes.StimulationEvent` attribute), 14

Q

[qualitative_dates\(\)](#) (in module `labbookdb.report.tracking`), 36
[qualitative_depth_reference](#) (`labbookdb.db.base_classes.OrthogonalStereotacticTarget` attribute), 14

R

[rate](#) (`labbookdb.db.common_classes.TreatmentProtocol` attribute), 25
[rate_unit](#) (`labbookdb.db.common_classes.TreatmentProtocol` attribute), 25
[rate_unit_id](#) (`labbookdb.db.common_classes.TreatmentProtocol` attribute), 25
[recording_bracket](#) (`labbookdb.db.common_classes.ForcedSwimTestMeasurement` attribute), 21
[recovery_bolus](#) (`labbookdb.db.common_classes.AnesthesiaProtocol` attribute), 15
[reference](#) (`labbookdb.db.base_classes.OrthogonalStereotacticTarget` attribute), 14

[reference_date](#) (`labbookdb.db.common_classes.DrinkingMeasurement` attribute), 19
[reference_date](#) (`labbookdb.db.common_classes.SucrosePreferenceMeasurement` attribute), 24
[relativize_dates\(\)](#) (in module `labbookdb.report.utilities`), 38
[report_animals\(\)](#) (`labbookdb.db.common_classes.CageStay` method), 18
[resonator](#) (`labbookdb.db.base_classes.FMRIScannerSetup` attribute), 12
[respiration](#) (`labbookdb.db.common_classes.AnesthesiaProtocol` attribute), 15
[revolutions_per_minute](#) (`labbookdb.db.common_classes.Incubation` attribute), 21
[roll](#) (`labbookdb.db.base_classes.OrthogonalStereotacticTarget` attribute), 14
[rounded_days\(\)](#) (in module `labbookdb.report.processing`), 29
[route](#) (`labbookdb.db.common_classes.TreatmentProtocol` attribute), 25

S

[sample_location](#) (`labbookdb.db.base_classes.Biopsy` attribute), 12
[sample_location](#) (`labbookdb.db.common_classes.BrainBiopsy` attribute), 17
[sample_mass](#) (`labbookdb.db.common_classes.DNAExtractionProtocol` attribute), 19
[scanner](#) (`labbookdb.db.base_classes.FMRIScannerSetup` attribute), 12
[scanner_setup](#) (`labbookdb.db.common_classes.FMRIMeasurement` attribute), 20
[scanner_setup_id](#) (`labbookdb.db.common_classes.FMRIMeasurement` attribute), 20
[sectioning_protocol](#) (`labbookdb.db.common_classes.BrainBiopsy` attribute), 17
[sectioning_protocol_id](#) (`labbookdb.db.common_classes.BrainBiopsy` attribute), 17
[SectioningProtocol](#) (class in `labbookdb.db.common_classes`), 23
[session_duration](#) (`labbookdb.db.common_classes.HandlingHabituationProtocol` attribute), 21

severity (<i>labbookdb.db.common_classes.Observation</i> attribute), 22	stereotactic_target (<i>labbookdb.db.common_classes.VirusInjectionProtocol</i> attribute), 25
sex (<i>labbookdb.db.common_classes.Animal</i> attribute), 16	stereotactic_target_id (<i>labbookdb.db.common_classes.OpticFiberImplantProtocol</i> attribute), 23
shape (<i>labbookdb.db.common_classes.Arena</i> attribute), 16	stereotactic_target_id (<i>labbookdb.db.common_classes.VirusInjectionProtocol</i> attribute), 26
single_caged (<i>labbookdb.db.common_classes.CageStay</i> attribute), 18	stimulation_protocol() (in module <i>labbookdb.report.selection</i>), 33
siunitx (<i>labbookdb.db.base_classes.MeasurementUnit</i> attribute), 13	stimulation_wavelength (<i>labbookdb.db.common_classes.FluorescentMicroscopyMeasurement</i> attribute), 20
slice_thickness (<i>labbookdb.db.common_classes.SectioningProtocol</i> attribute), 23	StimulationEvent (class in <i>labbookdb.db.base_classes</i>), 14
Solution (class in <i>labbookdb.db.common_classes</i>), 23	StimulationProtocol (class in <i>labbookdb.db.common_classes</i>), 24
solution (<i>labbookdb.db.common_classes.TreatmentProtocol</i> attribute), 25	stimulations (<i>labbookdb.db.common_classes.FMRIMeasurement</i> attribute), 20
solution_id (<i>labbookdb.db.common_classes.TreatmentProtocol</i> attribute), 25	storage_solution (<i>labbookdb.db.common_classes.BrainExtractionProtocol</i> attribute), 17
source (<i>labbookdb.db.base_classes.Virus</i> attribute), 15	storage_solution_id (<i>labbookdb.db.common_classes.BrainExtractionProtocol</i> attribute), 17
source (<i>labbookdb.db.common_classes.DNAExtraction</i> attribute), 18	storage_solution_volume (<i>labbookdb.db.common_classes.BrainExtractionProtocol</i> attribute), 17
source_id (<i>labbookdb.db.common_classes.DNAExtraction</i> attribute), 18	storage_time (<i>labbookdb.db.common_classes.BrainExtractionProtocol</i> attribute), 17
start_amount (<i>labbookdb.db.common_classes.DrinkingMeasurement</i> attribute), 19	strength (<i>labbookdb.db.base_classes.StimulationEvent</i> attribute), 15
start_bregma_distance (<i>labbookdb.db.common_classes.SectioningProtocol</i> attribute), 23	Substance (class in <i>labbookdb.db.common_classes</i>), 24
start_date (<i>labbookdb.db.base_classes.Biopsy</i> attribute), 12	substance (<i>labbookdb.db.common_classes.Ingredient</i> attribute), 22
start_date (<i>labbookdb.db.common_classes.BrainBiopsy</i> attribute), 17	substance_id (<i>labbookdb.db.common_classes.Ingredient</i> attribute), 22
start_date (<i>labbookdb.db.common_classes.CageStay</i> attribute), 18	sucrose_bottle_position (<i>labbookdb.db.common_classes.SucrosePreferenceMeasurement</i> attribute), 24
start_date (<i>labbookdb.db.common_classes.Treatment</i> attribute), 25	sucrose_concentration (<i>labbookdb.db.common_classes.SucrosePreferenceMeasurement</i> attribute), 24
start_depth (<i>labbookdb.db.common_classes.SectioningProtocol</i> attribute), 23	sucrose_end_amount (<i>labbookdb.db.common_classes.SucrosePreferenceMeasurement</i> attribute), 25
start_interaural_distance (<i>labbookdb.db.common_classes.SectioningProtocol</i> attribute), 23	sucrose_start_amount (<i>labbookdb.db.common_classes.SucrosePreferenceMeasurement</i> attribute), 25
start_lambda_distance (<i>labbookdb.db.common_classes.SectioningProtocol</i> attribute), 23	
start_midline_distance (<i>labbookdb.db.common_classes.SectioningProtocol</i> attribute), 23	
stays (<i>labbookdb.db.common_classes.Cage</i> attribute), 17	
stereotactic_target (<i>labbookdb.db.common_classes.OpticFiberImplantProtocol</i> attribute), 23	

SucrosePreferenceMeasurement (class in labbookdb.db.common_classes), 24

superoinferior (labbookdb.db.base_classes.OrthogonalStereotacticTarget attribute), 14

supplier (labbookdb.db.common_classes.Solution attribute), 24

supplier (labbookdb.db.common_classes.Substance attribute), 24

supplier_product_code (labbookdb.db.common_classes.Solution attribute), 24

supplier_product_code (labbookdb.db.common_classes.Substance attribute), 24

support (labbookdb.db.base_classes.FMRIScannerSetup attribute), 13

system (labbookdb.db.common_classes.SectioningProtocol attribute), 23

T

target (labbookdb.db.base_classes.StimulationEvent attribute), 15

temperature (labbookdb.db.common_classes.FMRIMeasurement attribute), 20

temperature (labbookdb.db.common_classes.ForcedSwimTestMeasurement attribute), 21

temperature (labbookdb.db.common_classes.Incubation attribute), 21

temperature_unit (labbookdb.db.common_classes.Incubation attribute), 21

temperature_unit_id (labbookdb.db.common_classes.Incubation attribute), 21

test_load() (in module labbookdb.tests.test_basic), 38

timedelta_sums() (in module labbookdb.report.processing), 29

timedeltas() (in module labbookdb.report.processing), 29

timetable() (in module labbookdb.report.selection), 33

transmittance (labbookdb.db.base_classes.OpticFiberImplant attribute), 14

transparent_tube (labbookdb.db.common_classes.HandlingHabituationProtocol attribute), 21

Treatment (class in labbookdb.db.common_classes), 25

treatment_group() (in module labbookdb.report.tracking), 36

TreatmentProtocol (class in labbookdb.db.common_classes), 25

treatments (labbookdb.db.common_classes.Animal attribute), 16

treatments (labbookdb.db.common_classes.Cage attribute), 17

trial_type (labbookdb.db.base_classes.StimulationEvent attribute), 15

type (labbookdb.db.base_classes.Biopsy attribute), 12

type (labbookdb.db.base_classes.Measurement attribute), 13

type (labbookdb.db.base_classes.Protocol attribute), 14

type (labbookdb.db.common_classes.AnesthesiaProtocol attribute), 15

type (labbookdb.db.common_classes.BrainBiopsy attribute), 17

type (labbookdb.db.common_classes.BrainExtractionProtocol attribute), 17

type (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 19

type (labbookdb.db.common_classes.DrinkingMeasurement attribute), 19

type (labbookdb.db.common_classes.FluorescentMicroscopyMeasurement attribute), 20

type (labbookdb.db.common_classes.FMRIMeasurement attribute), 20

type (labbookdb.db.common_classes.ForcedSwimTestMeasurement attribute), 21

type (labbookdb.db.common_classes.HandlingHabituationProtocol attribute), 21

type (labbookdb.db.common_classes.OpenFieldTestMeasurement attribute), 22

type (labbookdb.db.common_classes.OpticFiberImplantProtocol attribute), 23

type (labbookdb.db.common_classes.SectioningProtocol attribute), 23

type (labbookdb.db.common_classes.SucrosePreferenceMeasurement attribute), 25

type (labbookdb.db.common_classes.TreatmentProtocol attribute), 25

type (labbookdb.db.common_classes.VirusInjectionProtocol attribute), 26

type (labbookdb.db.common_classes.WeightMeasurement attribute), 26

U

unit (labbookdb.db.base_classes.StimulationEvent attribute), 15

unit (labbookdb.db.common_classes.Observation attribute), 22

unit_id (labbookdb.db.base_classes.StimulationEvent attribute), 15

unit_id (labbookdb.db.common_classes.Observation attribute), 22

V

`validate_email()` (labbookdb.db.base_classes.Operator method), 13

`value` (labbookdb.db.common_classes.Observation attribute), 22

`Virus` (class in labbookdb.db.base_classes), 15

`virus` (labbookdb.db.common_classes.VirusInjectionProtocol attribute), 26

`virus_diffusion_time` (labbookdb.db.common_classes.VirusInjectionProtocol attribute), 26

`virus_id` (labbookdb.db.common_classes.VirusInjectionProtocol attribute), 26

`virus_injection_speed` (labbookdb.db.common_classes.VirusInjectionProtocol attribute), 26

`VirusInjectionProtocol` (class in labbookdb.db.common_classes), 25

`volume_unit` (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 19

`volume_unit_id` (labbookdb.db.common_classes.DNAExtractionProtocol attribute), 19

`yaw` (labbookdb.db.base_classes.OrthogonalStereotacticTarget attribute), 14

Z

`z_dim` (labbookdb.db.common_classes.Arena attribute), 16

`zygosity` (labbookdb.db.base_classes.Genotype attribute), 13

W

`wall_color` (labbookdb.db.common_classes.Arena attribute), 16

`water_end_amount` (labbookdb.db.common_classes.SucrosePreferenceMeasurement attribute), 25

`water_start_amount` (labbookdb.db.common_classes.SucrosePreferenceMeasurement attribute), 25

`wavelength` (labbookdb.db.base_classes.StimulationEvent attribute), 15

`weight` (labbookdb.db.common_classes.WeightMeasurement attribute), 26

`weight_unit` (labbookdb.db.common_classes.WeightMeasurement attribute), 26

`weight_unit_id` (labbookdb.db.common_classes.WeightMeasurement attribute), 26

`WeightMeasurement` (class in labbookdb.db.common_classes), 26

X

`x_dim` (labbookdb.db.common_classes.Arena attribute), 16

Y

`y_dim` (labbookdb.db.common_classes.Arena attribute), 16