

---

# **SweFreq Documentation**

*Release development*

**NBIS**

**Jun 27, 2019**



---

## Contents:

---

<b>1</b>	<b>Code</b>	<b>1</b>
1.1	Backend . . . . .	1
<b>2</b>	<b>Documentation</b>	<b>31</b>
2.1	Set up a development system . . . . .	31
2.2	Importing data . . . . .	31
2.3	Merge accounts for Elixir AAI . . . . .	31
	<b>Python Module Index</b>	<b>33</b>
	<b>Index</b>	<b>35</b>



## 1.1 Backend

### 1.1.1 application

```
class ApproveUser (application: tornado.web.Application, request: tor-  
                   nado.httputil.HTTPServerRequest, **kwargs)  
    Bases: handlers.AdminHandler  
    post (dataset, email)  
  
class Collection (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest,  
                  **kwargs)  
    Bases: handlers.UnsafeHandler  
    get (dataset, ds_version=None)  
  
class CountryList (application: tornado.web.Application, request: tor-  
                   nado.httputil.HTTPServerRequest, **kwargs)  
    Bases: handlers.UnsafeHandler  
    country_list  
    get ()  
  
class DatasetFiles (application: tornado.web.Application, request: tor-  
                    nado.httputil.HTTPServerRequest, **kwargs)  
    Bases: handlers.AuthorizedHandler  
    get (dataset, ds_version=None)  
  
class DatasetUsersCurrent (application: tornado.web.Application, request: tor-  
                             nado.httputil.HTTPServerRequest, **kwargs)  
    Bases: handlers.AdminHandler  
    get (dataset)
```

```
class DatasetUsersPending (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.AdminHandler
```

```
    get (dataset)
```

```
class GenerateTemporaryLink (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.AuthorizedHandler
```

```
    post (dataset, ds_version=None)
```

```
class GetDataset (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.UnsafeHandler
```

```
    get (dataset, version=None)
```

```
class GetSchema (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.UnsafeHandler
```

Returns the schema.org, and bioschemas.org, annotation for a given url.

This function behaves quite differently from the rest of the application as the structured data testing tool had trouble catching the schema inject when it went through AngularJS. The solution for now has been to make this very general function that “re-parses” the ‘url’ request parameter to figure out what information to return.

```
    get ()
```

```
class GetUser (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.UnsafeHandler
```

```
    get ()
```

```
class ListDatasetVersions (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.UnsafeHandler
```

```
    get (dataset)
```

```
class ListDatasets (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.UnsafeHandler
```

```
    get ()
```

```
class LogEvent (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.SafeHandler
```

```
    post (dataset, event, target)
```

```
class QuitHandler (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.UnsafeHandler
```

```
    get ()
```

```
class RequestAccess (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.SafeHandler
```

```
    post (dataset)
```

```

class RevokeUser (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest,
                  **kwargs)
    Bases: handlers.AdminHandler

    post (dataset, email)

class SFTPAccess (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest,
                  **kwargs)
    Bases: handlers.SafeHandler

    Creates, or re-enables, sFTP users in the database.

    generate_password (size=12)
        Generates a password of length 'size', comprised of random lowercase and uppercase letters, and numbers.

    get ()
        Returns sFTP credentials for the current user.

    post ()
        Handles generation of new credentials. This function either creates a new set of sftp credentials for a user,
        or updates the old ones with a new password and expiry date.

class ServeLogo (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest,
                  **kwargs)
    Bases: handlers.UnsafeHandler

    get (dataset)

class UserDatasetAccess (application: tornado.web.Application, request: tor-
                          nado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.SafeHandler

    get ()

build_dataset_structure (dataset_version, user=None, dataset=None)

format_bytes (nbytes)

```

### 1.1.2 auth

```

class DeveloperLoginHandler (application: tornado.web.Application, request: tor-
                              nado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.BaseHandler

    get ()

class DeveloperLogoutHandler (application: tornado.web.Application, request: tor-
                                nado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.BaseHandler

    get ()

class ElixirLoginHandler (application: tornado.web.Application, request: tor-
                            nado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.BaseHandler, tornado.auth.OAuth2Mixin

    get ()

    get_user (access_token)

    get_user_token (code)

class ElixirLogoutHandler (application: tornado.web.Application, request: tor-
                             nado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.BaseHandler

```

```
get ()
```

### 1.1.3 db

```
class BaseModel (*args, **kwargs)
```

```
    Bases: peewee.Model
```

```
    DoesNotExist
```

```
        alias of BaseModelDoesNotExist
```

```
    id = <AutoField: BaseModel.id>
```

```
class BeaconCounts (*args, **kwargs)
```

```
    Bases: backend.db.BaseModel
```

```
    DoesNotExist
```

```
        alias of BeaconCountsDoesNotExist
```

```
    callcount = <IntegerField: BeaconCounts.callcount>
```

```
    datasetid = <CharField: BeaconCounts.datasetid>
```

```
    variantcount = <IntegerField: BeaconCounts.variantcount>
```

```
class Collection (*args, **kwargs)
```

```
    Bases: backend.db.BaseModel
```

A collection is a source of data which can be sampled into a SampleSet.

```
    DoesNotExist
```

```
        alias of CollectionDoesNotExist
```

```
    ethnicity = <CharField: Collection.ethnicity>
```

```
    id = <AutoField: Collection.id>
```

```
    name = <CharField: Collection.name>
```

```
    sample_sets
```

```
class Coverage (*args, **kwargs)
```

```
    Bases: backend.db.BaseModel
```

Coverage statistics are pre-calculated for each variant for a given dataset.

The fields show the fraction of a population that reaches the mapping coverages given by the variable names.

ex. **cov20 = 0.994** means that **99.4%** of the population had at a mapping coverage of at least 20 in this position.

```
    DoesNotExist
```

```
        alias of CoverageDoesNotExist
```

```
    chrom = <CharField: Coverage.chrom>
```

```
    coverage = <ArrayField: Coverage.coverage>
```

```
    dataset_version = <ForeignKeyField: Coverage.dataset_version>
```

```
    dataset_version_id = <ForeignKeyField: Coverage.dataset_version>
```

```
    id = <AutoField: Coverage.id>
```

```
    mean = <FloatField: Coverage.mean>
```

```
    median = <FloatField: Coverage.median>
```

```

    pos = <IntegerField: Coverage.pos>
class Dataset (*args, **kwargs)
    Bases: backend.db.BaseModel

    A dataset is part of a study, and usually include a certain population.
    Most studies only have a single dataset, but multiple are allowed.

    DoesNotExist
        alias of DatasetDoesNotExist

    access

    access_current

    access_logs

    access_pending

    avg_seq_depth = <FloatField: Dataset.avg_seq_depth>
    beacon_uri = <CharField: Dataset.beacon_uri>
    browser_uri = <CharField: Dataset.browser_uri>
    current_version

    dataset_size = <IntegerField: Dataset.dataset_size>
    description = <TextField: Dataset.description>
    full_name = <CharField: Dataset.full_name>
    has_image ()

    id = <AutoField: Dataset.id>
    logo

    sample_sets

    seq_center = <CharField: Dataset.seq_center>
    seq_tech = <CharField: Dataset.seq_tech>
    seq_type = <CharField: Dataset.seq_type>
    short_name = <CharField: Dataset.short_name>
    study = <ForeignKeyField: Dataset.study>
    study_id = <ForeignKeyField: Dataset.study>
    versions

class DatasetAccess (*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of DatasetAccessDoesNotExist

    dataset = <ForeignKeyField: DatasetAccess.dataset>
    dataset_id = <ForeignKeyField: DatasetAccess.dataset>
    id = <AutoField: DatasetAccess.id>
    is_admin = <BooleanField: DatasetAccess.is_admin>

```

```
user = <ForeignKeyField: DatasetAccess.user>
user_id = <ForeignKeyField: DatasetAccess.user>
wants_newsletter = <BooleanField: DatasetAccess.wants_newsletter>

class DatasetAccessCurrent(*args, **kwargs)
    Bases: backend.db.DatasetAccess

    DoesNotExist
        alias of DatasetAccessCurrentDoesNotExist

    access_requested = <DateTimeField: DatasetAccessCurrent.access_requested>
    dataset = <ForeignKeyField: DatasetAccessCurrent.dataset>
    dataset_id = <ForeignKeyField: DatasetAccessCurrent.dataset>
    has_access = <IntegerField: DatasetAccessCurrent.has_access>
    id = <AutoField: DatasetAccessCurrent.id>
    is_admin = <BooleanField: DatasetAccessCurrent.is_admin>
    user = <ForeignKeyField: DatasetAccessCurrent.user>
    user_id = <ForeignKeyField: DatasetAccessCurrent.user>
    wants_newsletter = <BooleanField: DatasetAccessCurrent.wants_newsletter>

class DatasetAccessPending(*args, **kwargs)
    Bases: backend.db.DatasetAccess

    DoesNotExist
        alias of DatasetAccessPendingDoesNotExist

    access_requested = <DateTimeField: DatasetAccessPending.access_requested>
    dataset = <ForeignKeyField: DatasetAccessPending.dataset>
    dataset_id = <ForeignKeyField: DatasetAccessPending.dataset>
    has_access = <IntegerField: DatasetAccessPending.has_access>
    id = <AutoField: DatasetAccessPending.id>
    is_admin = <BooleanField: DatasetAccessPending.is_admin>
    user = <ForeignKeyField: DatasetAccessPending.user>
    user_id = <ForeignKeyField: DatasetAccessPending.user>
    wants_newsletter = <BooleanField: DatasetAccessPending.wants_newsletter>

class DatasetFile(*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of DatasetFileDoesNotExist

    dataset_version = <ForeignKeyField: DatasetFile.dataset_version>
    dataset_version_id = <ForeignKeyField: DatasetFile.dataset_version>
    download_logs
    file_size = <IntegerField: DatasetFile.file_size>
    id = <AutoField: DatasetFile.id>
```

```
name = <CharField: DatasetFile.name>
uri = <CharField: DatasetFile.uri>
class DatasetLogo(*args, **kwargs)
  Bases: backend.db.BaseModel
  DoesNotExist
    alias of DatasetLogoDoesNotExist
  data = <BlobField: DatasetLogo.data>
  dataset = <ForeignKeyField: DatasetLogo.dataset>
  dataset_id = <ForeignKeyField: DatasetLogo.dataset>
  id = <AutoField: DatasetLogo.id>
  mimetype = <CharField: DatasetLogo.mimetype>
class DatasetVersion(*args, **kwargs)
  Bases: backend.db.BaseModel
  DoesNotExist
    alias of DatasetVersionDoesNotExist
  available_from = <DateTimeField: DatasetVersion.available_from>
  beacon_access = <EnumField: DatasetVersion.beacon_access>
  consent_logs
  coverage_levels = <ArrayField: DatasetVersion.coverage_levels>
  coverage_set
  data_contact_link = <CharField: DatasetVersion.data_contact_link>
  data_contact_name = <CharField: DatasetVersion.data_contact_name>
  dataset = <ForeignKeyField: DatasetVersion.dataset>
  dataset_id = <ForeignKeyField: DatasetVersion.dataset>
  description = <TextField: DatasetVersion.description>
  file_access = <EnumField: DatasetVersion.file_access>
  files
  id = <AutoField: DatasetVersion.id>
  link_hashes
  mate
  metrics_set
  num_variants = <IntegerField: DatasetVersion.num_variants>
  portal_avail = <BooleanField: DatasetVersion.portal_avail>
  ref_doi = <CharField: DatasetVersion.ref_doi>
  reference_set = <ForeignKeyField: DatasetVersion.reference_set>
  reference_set_id = <ForeignKeyField: DatasetVersion.reference_set>
  terms = <TextField: DatasetVersion.terms>
```

```
variants

version = <CharField: DatasetVersion.version>

class DatasetVersionCurrent(*args, **kwargs)
    Bases: backend.db.DatasetVersion

    DoesNotExist
        alias of DatasetVersionCurrentDoesNotExist

    available_from = <DateTimeField: DatasetVersionCurrent.available_from>
    beacon_access = <EnumField: DatasetVersionCurrent.beacon_access>
    coverage_levels = <ArrayField: DatasetVersionCurrent.coverage_levels>
    data_contact_link = <CharField: DatasetVersionCurrent.data_contact_link>
    data_contact_name = <CharField: DatasetVersionCurrent.data_contact_name>
    dataset = <ForeignKeyField: DatasetVersionCurrent.dataset>
    dataset_id = <ForeignKeyField: DatasetVersionCurrent.dataset>
    description = <TextField: DatasetVersionCurrent.description>
    file_access = <EnumField: DatasetVersionCurrent.file_access>
    id = <AutoField: DatasetVersionCurrent.id>
    num_variants = <IntegerField: DatasetVersionCurrent.num_variants>
    portal_avail = <BooleanField: DatasetVersionCurrent.portal_avail>
    ref_doi = <CharField: DatasetVersionCurrent.ref_doi>
    reference_set = <ForeignKeyField: DatasetVersionCurrent.reference_set>
    reference_set_id = <ForeignKeyField: DatasetVersionCurrent.reference_set>
    terms = <TextField: DatasetVersionCurrent.terms>
    version = <CharField: DatasetVersionCurrent.version>

class EnumField(choices=None, *args, **kwargs)
    Bases: peewee.Field

    db_field = 'string'

    db_value(value)

    python_value(value)

class Feature(*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of FeatureDoesNotExist

    chrom = <CharField: Feature.chrom>
    feature_type = <CharField: Feature.feature_type>
    gene = <ForeignKeyField: Feature.gene>
    gene_id = <ForeignKeyField: Feature.gene>
    id = <AutoField: Feature.id>
    start = <IntegerField: Feature.start>
```

```

    stop = <IntegerField: Feature.stop>
    strand = <EnumField: Feature.strand>
    transcript = <ForeignKeyField: Feature.transcript>
    transcript_id = <ForeignKeyField: Feature.transcript>
class Gene(*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of GeneDoesNotExist

    canonical_transcript = <CharField: Gene.canonical_transcript>
    chrom = <CharField: Gene.chrom>
    exons
    full_name = <CharField: Gene.full_name>
    gene_id = <CharField: Gene.gene_id>
    id = <AutoField: Gene.id>
    name = <CharField: Gene.name>
    other_names
    reference_set = <ForeignKeyField: Gene.reference_set>
    reference_set_id = <ForeignKeyField: Gene.reference_set>
    start = <IntegerField: Gene.start>
    stop = <IntegerField: Gene.stop>
    strand = <EnumField: Gene.strand>
    transcripts
    variants

class GeneOtherNames(*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of GeneOtherNamesDoesNotExist

    gene = <ForeignKeyField: GeneOtherNames.gene>
    gene_id = <ForeignKeyField: GeneOtherNames.gene>
    id = <AutoField: GeneOtherNames.id>
    name = <CharField: GeneOtherNames.name>

class Linkhash(*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of LinkhashDoesNotExist

    dataset_version = <ForeignKeyField: Linkhash.dataset_version>
    dataset_version_id = <ForeignKeyField: Linkhash.dataset_version>
    expires_on = <DateTimeField: Linkhash.expires_on>

```

```
hash = <CharField: Linkhash.hash>
id = <AutoField: Linkhash.id>
user = <ForeignKeyField: Linkhash.user>
user_id = <ForeignKeyField: Linkhash.user>

class Metrics(*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of MetricsDoesNotExist

    dataset_version = <ForeignKeyField: Metrics.dataset_version>
    dataset_version_id = <ForeignKeyField: Metrics.dataset_version>
    hist = <ArrayField: Metrics.hist>
    id = <AutoField: Metrics.id>
    metric = <CharField: Metrics.metric>
    mids = <ArrayField: Metrics.mids>
```

```
class ReferenceSet(*args, **kwargs)
    Bases: backend.db.BaseModel
```

The gencode, ensembl, dbNSFP and omim data are combined to fill out the Gene, Transcript and Feature tables. DbSNP data is separate, and may be shared between reference sets, so it uses a foreign key instead.

```
DoesNotExist
    alias of ReferenceSetDoesNotExist

current_version
dataset_versions
dbnsfp_version = <CharField: ReferenceSet.dbnsfp_version>
ensembl_version = <CharField: ReferenceSet.ensembl_version>
gencode_version = <CharField: ReferenceSet.gencode_version>
genes
id = <AutoField: ReferenceSet.id>
name = <CharField: ReferenceSet.name>
reference_build = <CharField: ReferenceSet.reference_build>
```

```
class SFTPUser(*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of SFTPUserDoesNotExist

    account_expires = <DateTimeField: SFTPUser.account_expires>
    id = <AutoField: SFTPUser.id>
    password_hash = <CharField: SFTPUser.password_hash>
    user = <ForeignKeyField: SFTPUser.user>
    user_id = <ForeignKeyField: SFTPUser.user>
```

```

    user_name = <CharField: SFTPUser.user_name>
    user_uid = <IntegerField: SFTPUser.user_uid>
class SampleSet (*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of SampleSetDoesNotExist
    collection = <ForeignKeyField: SampleSet.collection>
    collection_id = <ForeignKeyField: SampleSet.collection>
    dataset = <ForeignKeyField: SampleSet.dataset>
    dataset_id = <ForeignKeyField: SampleSet.dataset>
    id = <AutoField: SampleSet.id>
    phenotype = <CharField: SampleSet.phenotype>
    sample_size = <IntegerField: SampleSet.sample_size>
class Study (*args, **kwargs)
    Bases: backend.db.BaseModel

    A study is a scientific study with a PI and a description, and may include one or more datasets.

    DoesNotExist
        alias of StudyDoesNotExist
    contact_email = <CharField: Study.contact_email>
    contact_name = <CharField: Study.contact_name>
    datasets
    description = <TextField: Study.description>
    id = <AutoField: Study.id>
    pi_email = <CharField: Study.pi_email>
    pi_name = <CharField: Study.pi_name>
    publication_date = <DateTimeField: Study.publication_date>
    ref_doi = <CharField: Study.ref_doi>
    title = <CharField: Study.title>
class Transcript (*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of TranscriptDoesNotExist
    chrom = <CharField: Transcript.chrom>
    gene = <ForeignKeyField: Transcript.gene>
    gene_id = <ForeignKeyField: Transcript.gene>
    id = <AutoField: Transcript.id>
    mim_annotation = <CharField: Transcript.mim_annotation>
    mim_gene_accession = <IntegerField: Transcript.mim_gene_accession>

```

```
start = <IntegerField: Transcript.start>
stop = <IntegerField: Transcript.stop>
strand = <EnumField: Transcript.strand>
transcript_id = <CharField: Transcript.transcript_id>
transcripts
variants
class User(*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of UserDoesNotExist

    access_current
    access_logs
    access_pending
    affiliation = <CharField: User.affiliation>
    consent_logs
    country = <CharField: User.country>
    dataset_access
    download_logs
    email = <CharField: User.email>
    has_access(dataset, ds_version=None)
        Check whether user has permission to access a dataset

        Parameters
            • dataset (Database) – peewee Database object
            • ds_version (str) – the dataset version

        Returns allowed to access

        Return type bool

    has_requested_access(dataset)
    id = <AutoField: User.id>
    identity = <CharField: User.identity>
    identity_type = <EnumField: User.identity_type>
    is_admin(dataset)
    link_hashes
    name = <CharField: User.name>
    sftp_user

class UserAccessLog(*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of UserAccessLogDoesNotExist
```

```

    action = <EnumField:  UserAccessLog.action>
    dataset = <ForeignKeyField:  UserAccessLog.dataset>
    dataset_id = <ForeignKeyField:  UserAccessLog.dataset>
    id = <AutoField:  UserAccessLog.id>
    ts = <DateTimeField:  UserAccessLog.ts>
    user = <ForeignKeyField:  UserAccessLog.user>
    user_id = <ForeignKeyField:  UserAccessLog.user>
class UserConsentLog (*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of UserConsentLogDoesNotExist

    dataset_version = <ForeignKeyField:  UserConsentLog.dataset_version>
    dataset_version_id = <ForeignKeyField:  UserConsentLog.dataset_version>
    id = <AutoField:  UserConsentLog.id>
    ts = <DateTimeField:  UserConsentLog.ts>
    user = <ForeignKeyField:  UserConsentLog.user>
    user_id = <ForeignKeyField:  UserConsentLog.user>
class UserDownloadLog (*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of UserDownloadLogDoesNotExist

    dataset_file = <ForeignKeyField:  UserDownloadLog.dataset_file>
    dataset_file_id = <ForeignKeyField:  UserDownloadLog.dataset_file>
    id = <AutoField:  UserDownloadLog.id>
    ts = <DateTimeField:  UserDownloadLog.ts>
    user = <ForeignKeyField:  UserDownloadLog.user>
    user_id = <ForeignKeyField:  UserDownloadLog.user>
class Variant (*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of VariantDoesNotExist

    allele_count = <IntegerField:  Variant.allele_count>
    allele_freq = <FloatField:  Variant.allele_freq>
    allele_num = <IntegerField:  Variant.allele_num>
    alt = <CharField:  Variant.alt>
    chrom = <CharField:  Variant.chrom>
    dataset_version = <ForeignKeyField:  Variant.dataset_version>
    dataset_version_id = <ForeignKeyField:  Variant.dataset_version>

```

```
filter_string = <CharField: Variant.filter_string>
genes
hom_count = <IntegerField: Variant.hom_count>
id = <AutoField: Variant.id>
orig_alt_alleles = <ArrayField: Variant.orig_alt_alleles>
pos = <IntegerField: Variant.pos>
quality_metrics = <BinaryJSONField: Variant.quality_metrics>
ref = <CharField: Variant.ref>
rsid = <IntegerField: Variant.rsid>
site_quality = <FloatField: Variant.site_quality>
transcripts
variant_id = <CharField: Variant.variant_id>
vep_annotations = <BinaryJSONField: Variant.vep_annotations>

class VariantGenes(*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of VariantGenesDoesNotExist

    gene = <ForeignKeyField: VariantGenes.gene>
    gene_id = <ForeignKeyField: VariantGenes.gene>
    id = <AutoField: VariantGenes.id>
    variant = <ForeignKeyField: VariantGenes.variant>
    variant_id = <ForeignKeyField: VariantGenes.variant>

class VariantMate(*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of VariantMateDoesNotExist

    allele_count = <IntegerField: VariantMate.allele_count>
    allele_freq = <FloatField: VariantMate.allele_freq>
    allele_num = <IntegerField: VariantMate.allele_num>
    alt = <CharField: VariantMate.alt>
    chrom = <CharField: VariantMate.chrom>
    chrom_id = <CharField: VariantMate.chrom_id>
    dataset_version = <ForeignKeyField: VariantMate.dataset_version>
    dataset_version_id = <ForeignKeyField: VariantMate.dataset_version>
    id = <AutoField: VariantMate.id>
    mate_chrom = <CharField: VariantMate.mate_chrom>
    mate_id = <CharField: VariantMate.mate_id>
```

```

mate_start = <IntegerField: VariantMate.mate_start>
pos = <IntegerField: VariantMate.pos>
ref = <CharField: VariantMate.ref>
variant_id = <CharField: VariantMate.variant_id>
class VariantTranscripts (*args, **kwargs)
    Bases: backend.db.BaseModel

    DoesNotExist
        alias of VariantTranscriptsDoesNotExist
    id = <AutoField: VariantTranscripts.id>
    transcript = <ForeignKeyField: VariantTranscripts.transcript>
    transcript_id = <ForeignKeyField: VariantTranscripts.transcript>
    variant = <ForeignKeyField: VariantTranscripts.variant>
    variant_id = <ForeignKeyField: VariantTranscripts.variant>

build_dict_from_row (row)

get_admin_datasets (user)
    Get a list of datasets where user is admin

    Parameters user (User) – Peewee User object for the user of interest

    Returns

    Return type DataSetAccess

get_dataset (dataset: str)
    Given dataset name get Dataset

    Parameters dataset (str) – short name of the dataset

    Returns the corresponding DatasetVersion entry

    Return type Dataset

get_dataset_version (dataset: str, version: str = None)
    Given dataset get DatasetVersion

    Parameters dataset (str) – short name of the dataset

    Returns the corresponding DatasetVersion entry

    Return type DatasetVersion

get_next_free_uid ()
    Get the next free uid >= 10000 and > than the current uids from the sftp_user table in the db.

    Returns the next free uid

    Return type int

```

### 1.1.4 handlers

```

class AdminHandler (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
    Bases: backend.handlers.SafeHandler

```

**prepare ()**

This method is called before any other method. Having the decorator `@tornado.web.authenticated` here implies that all the Handlers that inherit from this one are going to require authentication in all their methods.

```
class AngularTemplate (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
```

Bases: `backend.handlers.UnsafeHandler`

**get (path)****initialize (path)**

```
class AuthorizedHandler (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
```

Bases: `backend.handlers.SafeHandler`

**prepare ()**

This method is called before any other method. Having the decorator `@tornado.web.authenticated` here implies that all the Handlers that inherit from this one are going to require authentication in all their methods.

```
class AuthorizedStaticNginxFileHandler (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
```

Bases: `backend.handlers.AuthorizedHandler`, `backend.handlers.BaseStaticNginxFileHandler`

Serve static files for authenticated users from the nginx frontend

Requires a “path” argument in constructor which should be the root of the nginx frontend where the files can be found. Then configure the nginx frontend something like this:

```
location <path> {
    internal;
    alias <location of files>;
}
```

```
class BaseHandler (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
```

Bases: `tornado.web.RequestHandler`

Base Handler. Handlers should not inherit from this class directly but from either `SafeHandler` or `UnsafeHandler` to make security status explicit.

**get\_current\_user ()**

Override to determine the current user from, e.g., a cookie.

This method may not be a coroutine.

**on\_finish ()**

Called after the end of a request.

Override this method to perform cleanup, logging, etc. This method is a counterpart to `prepare`. `on_finish` may not produce any output, as it is called after the response has been sent to the client.

**prepare ()**

Called at the beginning of a request before `get/post/etc`.

Override this method to perform common initialization regardless of the request method.

Asynchronous support: Use `async def` or decorate this method with `.gen.coroutine` to make it asynchronous. If this method returns an `Awaitable` execution will not proceed until the `Awaitable` is done.

New in version 3.1: Asynchronous support.

**set\_user\_msg** (*msg*, *level='info'*)

This function sets the user message cookie. The system takes four default levels, 'success', 'info', 'warning', and 'error'. Messages set to other levels will be defaulted to 'info'.

**write** (*chunk*)

Writes the given chunk to the output buffer.

To write the output to the network, use the *flush()* method below.

If the given chunk is a dictionary, we write it as JSON and set the Content-Type of the response to be `application/json`. (if you want to send JSON as a different Content-Type, call `set_header` after calling `write()`).

Note that lists are not converted to JSON because of a potential cross-site security vulnerability. All JSON output should be wrapped in a dictionary. More details at <http://haacked.com/archive/2009/06/25/json-hijacking.aspx/> and <https://github.com/facebook/tornado/issues/1009>

**write\_error** (*status\_code*, *\*\*kwargs*)

Overwrites `write_error` method to have custom error pages. [http://tornado.readthedocs.org/en/latest/web.html#tornado.web.RequestHandler.write\\_error](http://tornado.readthedocs.org/en/latest/web.html#tornado.web.RequestHandler.write_error)

**class BaseStaticNginxFileHandler** (*application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, \*\*kwargs*)

Bases: `backend.handlers.UnsafeHandler`

Serve static files for users from the nginx frontend

Requires a `path` argument in constructor which should be the root of the nginx frontend where the files can be found. Then configure the nginx frontend something like this:

```
location <path> {
    internal;
    alias <location of files>;
}
```

**get** (*dataset, file, ds\_version=None, user=None*)

**initialize** (*path*)

**class SafeHandler** (*application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, \*\*kwargs*)

Bases: `backend.handlers.BaseHandler`

All handlers that need authentication and authorization should inherit from this class.

**prepare** ()

This method is called before any other method. Having the decorator `@tornado.web.authenticated` here implies that all the Handlers that inherit from this one are going to require authentication in all their methods.

**class SafeStaticFileHandler** (*application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, \*\*kwargs*)

Bases: `tornado.web.StaticFileHandler, backend.handlers.SafeHandler`

Serve static files for logged in users

**class TemporaryStaticNginxFileHandler** (*application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, \*\*kwargs*)

Bases: `backend.handlers.BaseStaticNginxFileHandler`

**get** (*dataset, ds\_version, hash\_value, file*)

```
class UnsafeHandler (application: tornado.web.Application, request: tor-
                    nado.httputil.HTTPServerRequest, **kwargs)
    Bases: backend.handlers.BaseHandler
```

### 1.1.5 route

```
class Application (settings)
    Bases: tornado.web.Application
```

### 1.1.6 settings

### 1.1.7 test

```
class RequestTests (methodName='runTest')
    Bases: unittest.case.TestCase

    HOST = 'http://localhost:4000'

    assertHTTPCode (path, code=200, method='get', *args, **kwargs)

    cookies

    destroySession ()

    get (path, *args, **kwargs)

    getUrl (path)

    login_user (user)

    newSession ()

    post (path, *args, **kwargs)

    session
```

```
class TestAdminAccess (methodName='runTest')
    Bases: backend.test.RequestTests

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

    test_admin_is_admin ()

    test_admin_list_users ()

    test_admin_list_users_get_data ()

    test_admin_list_users_only_own_project_1 ()

    test_admin_list_users_only_own_project_2 ()

    test_login_admin ()
```

```
class TestEndpoints (methodName='runTest')
    Bases: backend.test.RequestTests

    test_create_temporary_link ()

    test_dataset_collection ()
```

```
test_dataset_logo ()
test_datasets ()
test_get_one_version ()
test_get_users_current ()
test_get_users_pending ()
test_get_versions ()
test_list_files ()
test_one_dataset ()
test_request_access ()
test_user_datasets ()
test_user_me ()

class TestLoggedInUser (methodName='runTest')
    Bases: backend.test.RequestTests

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

    testLoggedInFiles ()

    testLoggedInTempLinkGet ()

    testLoggedInTempLinkPost ()

    testLoggedInTempLinkPostXSRF1 ()

    testLoggedInTempLinkPostXSRF2 ()

class TestRequestAccess (methodName='runTest')
    Bases: backend.test.RequestTests

    USER = 'e1'

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    tearDown ()
        Hook method for deconstructing the test fixture after testing it.

    test_get_xsrf_token ()

    test_login ()

    test_request_access_correctly ()

    test_request_access_with_get ()

    test_request_access_with_wrong_xsrf_1 ()

    test_request_access_with_wrong_xsrf_2 ()

    test_request_access_without_xsrf ()

class TestUserManagement (methodName='runTest')
    Bases: backend.test.RequestTests
```

```
setUp ()
    Hook method for setting up the test fixture before exercising it.

tearDown ()
    Hook method for deconstructing the test fixture after testing it.

test_admin_approve_user ()

test_full_user_roundabout ()

test_recently_approved_user_can_list_files ()

test_recently_revoked_user_cant_list_files ()
```

## 1.1.8 Variant browser

### browser\_handlers

Request handlers for the variant browser.

```
class Autocomplete (application: tornado.web.Application, request: tor-
                    nado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.UnsafeHandler
```

Provide autocompletion for protein names based on current query.

```
get (dataset: str, query: str, ds_version: str = None)
    Provide autocompletion for protein names based on current query.
```

#### Parameters

- **dataset** (*str*) – dataset short name
- **query** (*str*) – query
- **ds\_version** (*str*) – dataset version

```
class Download (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest,
                **kwargs)
    Bases: handlers.UnsafeHandler
```

Download variants in CSV format.

```
get (dataset: str, datatype: str, item: str, ds_version: str = None, filter_type: str = None)
    Download variants in CSV format.
```

Will filter the variants if filter\_type is provided.

#### Parameters

- **dataset** (*str*) – dataset short name
- **datatype** (*str*) – type of data
- **item** (*str*) – query item
- **ds\_version** (*str*) – dataset version
- **filter\_type** (*str*) – type of filter to apply

```
class GetCoverage (application: tornado.web.Application, request: tor-
                   nado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.UnsafeHandler
```

Retrieve coverage.

**get** (*dataset: str, datatype: str, item: str, ds\_version: str = None*)  
Retrieve coverage.

#### Parameters

- **dataset** (*str*) – dataset short name
- **datatype** (*str*) – type of data
- **item** (*str*) – query item
- **ds\_version** (*str*) – dataset version

**class GetCoveragePos** (*application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, \*\*kwargs*)

Bases: `handlers.UnsafeHandler`

Retrieve coverage range.

**get** (*dataset: str, datatype: str, item: str, ds\_version: str = None*)  
Retrieve coverage range.

#### Parameters

- **dataset** (*str*) – dataset short name
- **datatype** (*str*) – type of data
- **item** (*str*) – query item
- **ds\_version** (*str*) – dataset version

**class GetGene** (*application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, \*\*kwargs*)

Bases: `handlers.UnsafeHandler`

Request information about a gene.

**get** (*dataset: str, gene: str, ds\_version: str = None*)  
Request information about a gene.

#### Parameters

- **dataset** (*str*) – short name of the dataset
- **gene** (*str*) – the gene id
- **ds\_version** (*str*) – dataset version

**class GetRegion** (*application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, \*\*kwargs*)

Bases: `handlers.UnsafeHandler`

Request information about genes in a region.

**get** (*dataset: str, region: str, ds\_version: str = None*)  
Request information about genes in a region.

#### Parameters

- **dataset** (*str*) – short name of the dataset
- **region** (*str*) – the region in the format chr-startpos-endpos
- **ds\_version** (*str*) – dataset version

```
class GetTranscript (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.UnsafeHandler
```

Request information about a transcript.

```
get (dataset: str, transcript: str, ds_version: str = None)
    Request information about a transcript.
```

#### Parameters

- **dataset** (*str*) – short name of the dataset
- **transcript** (*str*) – the transcript id

**Returns** transcript (transcript and exons), gene (gene information)

**Return type** dict

```
class GetVariant (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.UnsafeHandler
```

Request information about a gene.

```
get (dataset: str, variant: str, ds_version: str = None)
    Request information about a gene.
```

#### Parameters

- **dataset** (*str*) – short name of the dataset
- **variant** (*str*) – variant in the format chrom-pos-ref-alt

```
class GetVariants (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.UnsafeHandler
```

Retrieve variants.

```
get (dataset: str, datatype: str, item: str, ds_version: str = None)
    Retrieve variants.
```

#### Parameters

- **dataset** (*str*) – short name of the dataset
- **datatype** (*str*) – gene, region, or transcript
- **item** (*str*) – item to query

```
class Search (application: tornado.web.Application, request: tornado.httputil.HTTPServerRequest, **kwargs)
    Bases: handlers.UnsafeHandler
```

Perform a search for the wanted object.

```
get (dataset: str, query: str, ds_version: str = None)
    Perform a search for the wanted object.
```

#### Parameters

- **dataset** (*str*) – short name of the dataset
- **query** (*str*) – search query

## lookups

Lookups for a PostgreSQL database with genomic data.

Lookup functions for the variant browser.

**autocomplete** (*dataset: str, query: str, ds\_version: str = None*)

Provide autocomplete suggestions based on the query.

### Parameters

- **dataset** (*str*) – short name of dataset
- **query** (*str*) – the query to compare to the available gene names
- **ds\_version** (*str*) – the dataset version

**Returns** A list of genes names whose beginning matches the query

**Return type** `list`

**get\_awesomebar\_result** (*dataset: str, query: str, ds\_version: str = None*)

Parse the search input.

Datatype is one of:

- *gene*
- *transcript*
- *variant*
- *dbsnp\_variant\_set*
- *region*

Identifier is one of:

- ensembl ID for gene
- variant ID string for variant (eg. 1-1000-A-T)
- region ID string for region (eg. 1-1000-2000)

Follow these steps:

- if query is an ensembl ID, return it
- if a gene symbol, return that gene's ensembl ID
- if an RSID, return that variant's string

### Parameters

- **dataset** (*str*) – short name of dataset
- **query** (*str*) – the search query
- **ds\_version** (*str*) – the dataset version

**Returns** (datatype, identifier)

**Return type** `tuple`

**get\_coverage\_for\_bases** (*dataset: str, chrom: str, start\_pos: int, end\_pos: int = None, ds\_version: str = None*)

Get the coverage for the list of bases given by start\_pos->end\_pos, inclusive.

### Parameters

- **dataset** (*str*) – short name for the dataset
- **chrom** (*str*) – chromosome
- **start\_pos** (*int*) – first position of interest
- **end\_pos** (*int*) – last position of interest; if None it will be set to start\_pos
- **ds\_version** (*str*) – version of the dataset

**Returns** coverage dicts for the region of interest. None if failed

**Return type** list

**get\_coverage\_for\_transcript** (*dataset: str, chrom: str, start\_pos: int, end\_pos: int = None, ds\_version: str = None*)

Get the coverage for the list of bases given by start\_pos->end\_pos, inclusive.

**Parameters**

- **dataset** (*str*) – short name for the dataset
- **chrom** (*str*) – chromosome
- **start\_pos** (*int*) – first position of interest
- **end\_pos** (*int*) – last position of interest; if None it will be set to start\_pos
- **ds\_version** (*str*) – version of the dataset

**Returns** coverage dicts for the region of interest

**Return type** list

**get\_exons\_in\_transcript** (*dataset: str, transcript\_id: str, ds\_version=None*)

Retrieve exons associated with the given transcript id.

**Parameters**

- **dataset** (*str*) – short name of the dataset
- **transcript\_id** (*str*) – the id of the transcript
- **ds\_version** (*str*) – dataset version

**Returns** dicts with values for each exon sorted by start position

**Return type** list

**get\_gene** (*dataset: str, gene\_id: str, ds\_version: str = None*)

Retrieve gene by gene id.

**Parameters**

- **dataset** (*str*) – short name of the dataset
- **gene\_id** (*str*) – the id of the gene
- **ds\_version** (*str*) – dataset version

**Returns** values for the gene; None if not found

**Return type** dict

**get\_gene\_by\_dbid** (*gene\_dbid: str*)

Retrieve gene by gene database id.

**Parameters** **gene\_dbid** (*str*) – the database id of the gene

**Returns** values for the gene; empty if not found

**Return type** dict

**get\_gene\_by\_name** (*dataset: str, gene\_name: str, ds\_version=None*)

Retrieve gene by gene\_name.

**Parameters**

- **dataset** (*str*) – short name of the dataset
- **gene\_name** (*str*) – the id of the gene
- **ds\_version** (*str*) – dataset version

**Returns** values for the gene; empty if not found

**Return type** dict

**get\_genes\_in\_region** (*dataset: str, chrom: str, start\_pos: int, stop\_pos: int, ds\_version: str = None*)

Retrieve genes located within a region.

**Parameters**

- **dataset** (*str*) – short name of the dataset
- **chrom** (*str*) – chromosome name
- **start\_pos** (*int*) – start of region
- **stop\_pos** (*int*) – end of region
- **ds\_version** (*str*) – dataset version

**Returns** values for the gene; empty if not found

**Return type** dict

**get\_raw\_variant** (*dataset: str, pos: int, chrom: str, ref: str, alt: str, ds\_version: str = None*)

Retrieve variant by position and change.

**Parameters**

- **dataset** (*str*) – short name of the reference set
- **pos** (*int*) – position of the variant
- **chrom** (*str*) – name of the chromosome
- **ref** (*str*) – reference sequence
- **alt** (*str*) – variant sequence
- **ds\_version** (*str*) – dataset version

**Returns** values for the variant; None if not found

**Return type** dict

**get\_transcript** (*dataset: str, transcript\_id: str, ds\_version: str = None*)

Retrieve transcript by transcript id.

Also includes exons as ['exons']

**Parameters**

- **dataset** (*str*) – short name of the dataset
- **transcript\_id** (*str*) – the id of the transcript
- **ds\_version** (*str*) – dataset version

**Returns** values for the transcript, including exons; None if not found

**Return type** dict

**get\_transcripts\_in\_gene** (*dataset: str, gene\_id: str, ds\_version: str = None*)

Get the transcripts associated with a gene.

**Parameters**

- **dataset** (*str*) – short name of the reference set
- **gene\_id** (*str*) – id of the gene
- **ds\_version** (*str*) – dataset version

**Returns** transcripts (dict) associated with the gene; empty if no hits

**Return type** list

**get\_transcripts\_in\_gene\_by\_dbid** (*gene\_dbid: int*)

Get the transcripts associated with a gene.

**Parameters** **gene\_dbid** (*int*) – database id of the gene

**Returns** transcripts (dict) associated with the gene; empty if no hits

**Return type** list

**get\_variant** (*dataset: str, pos: int, chrom: str, ref: str, alt: str, ds\_version: str = None*)

Retrieve variant by position and change.

**Parameters**

- **dataset** (*str*) – short name of the dataset
- **pos** (*int*) – position of the variant
- **chrom** (*str*) – name of the chromosome
- **ref** (*str*) – reference sequence
- **alt** (*str*) – variant sequence
- **ds\_version** (*str*) – version of the dataset

**Returns** values for the variant; None if not found

**Return type** dict

**get\_variants\_by\_rsid** (*dataset: str, rsid: str, ds\_version: str = None*)

Retrieve variants by their associated rsid.

**Parameters**

- **dataset** (*str*) – short name of dataset
- **rsid** (*str*) – rsid of the variant (starting with rs)
- **ds\_version** (*str*) – version of the dataset

**Returns** variants as dict; no hits returns None

**Return type** list

**get\_variants\_in\_gene** (*dataset: str, gene\_id: str, ds\_version: str = None*)

Retrieve variants present inside a gene.

**Parameters**

- **dataset** (*str*) – short name of the dataset

- **gene\_id** (*str*) – id of the gene
- **ds\_version** (*str*) – version of the dataset

**Returns** values for the variants

**Return type** *list*

**get\_variants\_in\_region** (*dataset: str, chrom: str, start\_pos: int, end\_pos: int, ds\_version: str = None*)

Variants that overlap a region.

**Parameters**

- **dataset** (*str*) – short name of the dataset
- **chrom** (*str*) – name of the chromosom
- **start\_pos** (*int*) – start of the region
- **end\_pos** (*int*) – start of the region
- **ds\_version** (*str*) – version of the dataset

**Returns** variant dicts, None if no hits

**Return type** *list*

**get\_variants\_in\_transcript** (*dataset: str, transcript\_id: str, ds\_version: str = None*)

Retrieve variants inside a transcript.

**Parameters**

- **dataset** (*str*) – short name of the dataset
- **transcript\_id** (*str*) – id of the transcript (ENST)
- **ds\_version** (*str*) – version of the dataset

**Returns** values for the variant; None if not found

**Return type** *dict*

## route

Routing definitions

## utils

Utility functions for lookups and browser\_handlers.

**add\_consequence\_to\_variant** (*variant: dict*)

Add information about variant consequence to a variant.

**Parameters** **variant** (*dict*) – variant information

**add\_consequence\_to\_variants** (*variant\_list: list*)

Add information about variant consequence to multiple variants.

**Parameters**

- **variant\_list** (*list*) – list of variants
- **datatype** (*str*) – type of data
- **item** (*str*) – query item

**annotation\_severity** (*annotation: dict*)

Evaluate severity of the consequences; “bigger is more important”.

**Parameters** **annotation** (*dict*) – vep\_annotation from a variant

**Returns** severity score

**Return type** *float*

**get\_coverage** (*dataset: str, datatype: str, item: str, ds\_version: str = None*)

Retrieve coverage for a gene/region/transcript.

**Parameters**

- **dataset** (*str*) – short name of the dataset
- **datatype** (*str*) – type of “region” (gene/region/transcript)
- **item** (*str*) – the datatype item to look up
- **ds\_version** (*str*) – the dataset version

**Returns** start, stop, coverage list

**Return type** *dict*

**get\_coverage\_pos** (*dataset: str, datatype: str, item: str, ds\_version: str = None*)

Retrieve coverage range.

**Parameters**

- **dataset** (*str*) – short name of the dataset
- **datatype** (*str*) – type of “region” (gene/region/transcript)
- **item** (*str*) – the datatype item to look up

**Returns** start, stop, chrom

**Return type** *dict*

**get\_flags\_from\_variant** (*variant: dict*)

Get flags from variant.

Checks for: \* MNP \* LoF (loss of function)

**Parameters** **variant** (*dict*) – a variant

**Returns** flags for the variant

**Return type** *list*

**get\_proper\_hgvs** (*annotation: dict*)

Get HGVS for change, either at transcript or protein level.

**Parameters** **annotation** (*dict*) – VEP annotation with HGVS information

**Returns** variant effect at aa level in HGVS format (p.), None if parsing fails

**Return type** *str*

**get\_protein\_hgvs** (*annotation*)

Aa changes in HGVS format.

**Parameters** **annotation** (*dict*) – VEP annotation with HGVS information

**Returns** variant effect at aa level in HGVS format (p.), None if parsing fails

**Return type** *str*

**get\_transcript\_hgvs** (*annotation: dict*)

Nucleotide change in HGVS format.

**Parameters** **annotation** (*dict*) – VEP annotation with HGVS information

**Returns** variant effect at nucleotide level in HGVS format (c.), None if parsing fails

**Return type** *str*

**get\_variant\_list** (*dataset: str, datatype: str, item: str, ds\_version: str = None*)

Retrieve variants for a datatype.

**Parameters**

- **dataset** (*str*) – dataset short name
- **datatype** (*str*) – type of data
- **item** (*str*) – query item
- **ds\_version** (*str*) – dataset version

**Returns** {variants:list, headers:list}

**Return type** *dict*

**is\_region\_too\_large** (*start: int, stop: int*)

Evaluate whether the size of a region is larger than maximum query.

**Parameters**

- **start** (*int*) – Start position of the region
- **stop** (*int*) – End position of the region

**Returns** True if too large

**Return type** *bool*

**order\_vep\_by\_csq** (*annotation\_list: list*)

Will add “major\_consequence” to each annotation and order by severity.

**Parameters** **annotation\_list** (*list*) – VEP annotations (as dict)

**Returns** annotations ordered by major consequence severity

**Return type** *list*

**parse\_dataset** (*dataset: str, ds\_version: str = None*)

Check/parse if the dataset name is in the beacon form (*reference:dataset:version*).

**Parameters**

- **dataset** (*str*) – short name of the dataset
- **ds\_version** (*str*) – the dataset version

**Returns** (dataset, version)

**Return type** *tuple*

**parse\_region** (*region: str*)

Parse a region with either one or two positions

**Parameters** **region** (*str*) – region, e.g. 3-100-200 or 3-100

**Returns** (chrom, start, pos)

**Return type** *tuple*

**remove\_extraneous\_information** (*variant: dict*)

Remove information that is not used in the frontend from a variant.

**Parameters** **variant** (*dict*) – variant data from database

**remove\_extraneous\_vep\_annotations** (*annotation\_list: list*)

Remove annotations with low-impact consequences (less than intron variant).

**Parameters** **annotation\_list** (*list*) – VEP annotations (as dict)

**Returns** VEP annotations with higher impact

**Return type** *list*

**worst\_csq\_from\_csq** (*csq: str*)

Find worst consequence in a possibly &-filled consequence string.

**Parameters** **csq** (*str*) – string of consequences, seperated with & (if multiple)

**Returns** the worst consequence

**Return type** *str*

**worst\_csq\_from\_list** (*csq\_list: list*)

Choose the worst consequence.

**Parameters** **csq\_list** (*list*) – list of consequences

**Returns** the worst consequence

**Return type** *str*

**worst\_csq\_index** (*csq\_list: list*)

Find the index of the worst consequence.

Corresponds to the lowest value (index) from CSQ\_ORDER\_DICT.

**Parameters** **csq\_list** (*list*) – consequences

**Returns** index in CSQ\_ODER\_DICT of the worst consequence

**Return type** *int*

**worst\_csq\_with\_vep** (*annotation\_list: list*)

Choose the vep annotation with the most severe consequence.

Add a "major\_consequence" field for that annotation.

**Parameters** **annotation\_list** (*list*) – VEP annotations

**Returns** the annotation with the most severe consequence

**Return type** *dict*

## 2.1 Set up a development system

In order to set up a minimal database system for development:

1. Install docker (and docker-compose in case it's not included in the installation)
2. Start the server and database:

```
` $ docker-compose up `
```

3. Add test data:

```
` $ psql -h localhost -U postgres swefreq -f test/data/browser_test_data.sql `
```

The test data contains all data required for the browser tests.

## 2.2 Importing data

The data import system can be found at *scripts/importer*, and helper scripts are available in *scripts/*.

## 2.3 Merge accounts for Elixir AAI

It is possible to maintain the same dataset permissions with other logins (e.g. you are a dataset admin with your institutional account and want to be able to login with your ORCID account and still be an admin).

To merge the accounts:

1. Log in to the Perun Identity consolidator at <https://perun.elixir-czech.cz/fed/gui> with your account with admin access.
2. Go to the authentication tab and click *identity consolidator >>*.

3. Log in with your second account.
  - genindex

### b

- `backend.application`, 1
- `backend.auth`, 3
- `backend.db`, 4
- `backend.handlers`, 15
- `backend.modules.browser.browser_handlers`,  
20
- `backend.modules.browser.lookups`, 23
- `backend.modules.browser.route`, 27
- `backend.modules.browser.utils`, 27
- `backend.route`, 18
- `backend.settings`, 18
- `backend.test`, 18
- `browser`, 20



## A

access (*Dataset attribute*), 5  
 access\_current (*Dataset attribute*), 5  
 access\_current (*User attribute*), 12  
 access\_logs (*Dataset attribute*), 5  
 access\_logs (*User attribute*), 12  
 access\_pending (*Dataset attribute*), 5  
 access\_pending (*User attribute*), 12  
 access\_requested (*DatasetAccessCurrent attribute*), 6  
 access\_requested (*DatasetAccessPending attribute*), 6  
 account\_expires (*SFTPUser attribute*), 10  
 action (*UserAccessLog attribute*), 12  
 add\_consequence\_to\_variant() (in module *backend.modules.browser.utils*), 27  
 add\_consequence\_to\_variants() (in module *backend.modules.browser.utils*), 27  
 AdminHandler (class in *backend.handlers*), 15  
 affiliation (*User attribute*), 12  
 allele\_count (*Variant attribute*), 13  
 allele\_count (*VariantMate attribute*), 14  
 allele\_freq (*Variant attribute*), 13  
 allele\_freq (*VariantMate attribute*), 14  
 allele\_num (*Variant attribute*), 13  
 allele\_num (*VariantMate attribute*), 14  
 alt (*Variant attribute*), 13  
 alt (*VariantMate attribute*), 14  
 AngularTemplate (class in *backend.handlers*), 16  
 annotation\_severity() (in module *backend.modules.browser.utils*), 27  
 Application (class in *backend.route*), 18  
 ApproveUser (class in *backend.application*), 1  
 assertHTTPCode() (*RequestTests method*), 18  
 AuthorizedHandler (class in *backend.handlers*), 16  
 AuthorizedStaticNginxFileHandler (class in *backend.handlers*), 16  
 Autocomplete (class in *backend.modules.browser.browser\_handlers*),

20

autocomplete() (in module *backend.modules.browser.lookups*), 23  
 available\_from (*DatasetVersion attribute*), 7  
 available\_from (*DatasetVersionCurrent attribute*), 8  
 avg\_seq\_depth (*Dataset attribute*), 5

## B

backend.application (module), 1  
 backend.auth (module), 3  
 backend.db (module), 4  
 backend.handlers (module), 15  
 backend.modules.browser.browser\_handlers (module), 20  
 backend.modules.browser.lookups (module), 23  
 backend.modules.browser.route (module), 27  
 backend.modules.browser.utils (module), 27  
 backend.route (module), 18  
 backend.settings (module), 18  
 backend.test (module), 18  
 BaseHandler (class in *backend.handlers*), 16  
 BaseModel (class in *backend.db*), 4  
 BaseStaticNginxFileHandler (class in *backend.handlers*), 17  
 beacon\_access (*DatasetVersion attribute*), 7  
 beacon\_access (*DatasetVersionCurrent attribute*), 8  
 beacon\_uri (*Dataset attribute*), 5  
 BeaconCounts (class in *backend.db*), 4  
 browser (module), 20  
 browser\_uri (*Dataset attribute*), 5  
 build\_dataset\_structure() (in module *backend.application*), 3  
 build\_dict\_from\_row() (in module *backend.db*), 15

C

callcount (*BeaconCounts attribute*), 4  
 canonical\_transcript (*Gene attribute*), 9

chrom (*Coverage attribute*), 4  
 chrom (*Feature attribute*), 8  
 chrom (*Gene attribute*), 9  
 chrom (*Transcript attribute*), 11  
 chrom (*Variant attribute*), 13  
 chrom (*VariantMate attribute*), 14  
 chrom\_id (*VariantMate attribute*), 14  
 Collection (*class in backend.application*), 1  
 Collection (*class in backend.db*), 4  
 collection (*SampleSet attribute*), 11  
 collection\_id (*SampleSet attribute*), 11  
 consent\_logs (*DatasetVersion attribute*), 7  
 consent\_logs (*User attribute*), 12  
 contact\_email (*Study attribute*), 11  
 contact\_name (*Study attribute*), 11  
 cookies (*RequestTests attribute*), 18  
 country (*User attribute*), 12  
 country\_list (*CountryList attribute*), 1  
 CountryList (*class in backend.application*), 1  
 Coverage (*class in backend.db*), 4  
 coverage (*Coverage attribute*), 4  
 coverage\_levels (*DatasetVersion attribute*), 7  
 coverage\_levels (*DatasetVersionCurrent attribute*), 8  
 coverage\_set (*DatasetVersion attribute*), 7  
 current\_version (*Dataset attribute*), 5  
 current\_version (*ReferenceSet attribute*), 10

## D

data (*DatasetLogo attribute*), 7  
 data\_contact\_link (*DatasetVersion attribute*), 7  
 data\_contact\_link (*DatasetVersionCurrent attribute*), 8  
 data\_contact\_name (*DatasetVersion attribute*), 7  
 data\_contact\_name (*DatasetVersionCurrent attribute*), 8  
 Dataset (*class in backend.db*), 5  
 dataset (*DatasetAccess attribute*), 5  
 dataset (*DatasetAccessCurrent attribute*), 6  
 dataset (*DatasetAccessPending attribute*), 6  
 dataset (*DatasetLogo attribute*), 7  
 dataset (*DatasetVersion attribute*), 7  
 dataset (*DatasetVersionCurrent attribute*), 8  
 dataset (*SampleSet attribute*), 11  
 dataset (*UserAccessLog attribute*), 13  
 dataset\_access (*User attribute*), 12  
 dataset\_file (*UserDownloadLog attribute*), 13  
 dataset\_file\_id (*UserDownloadLog attribute*), 13  
 dataset\_id (*DatasetAccess attribute*), 5  
 dataset\_id (*DatasetAccessCurrent attribute*), 6  
 dataset\_id (*DatasetAccessPending attribute*), 6  
 dataset\_id (*DatasetLogo attribute*), 7  
 dataset\_id (*DatasetVersion attribute*), 7  
 dataset\_id (*DatasetVersionCurrent attribute*), 8

dataset\_id (*SampleSet attribute*), 11  
 dataset\_id (*UserAccessLog attribute*), 13  
 dataset\_size (*Dataset attribute*), 5  
 dataset\_version (*Coverage attribute*), 4  
 dataset\_version (*DatasetFile attribute*), 6  
 dataset\_version (*Linkhash attribute*), 9  
 dataset\_version (*Metrics attribute*), 10  
 dataset\_version (*UserConsentLog attribute*), 13  
 dataset\_version (*Variant attribute*), 13  
 dataset\_version (*VariantMate attribute*), 14  
 dataset\_version\_id (*Coverage attribute*), 4  
 dataset\_version\_id (*DatasetFile attribute*), 6  
 dataset\_version\_id (*Linkhash attribute*), 9  
 dataset\_version\_id (*Metrics attribute*), 10  
 dataset\_version\_id (*UserConsentLog attribute*), 13  
 dataset\_version\_id (*Variant attribute*), 13  
 dataset\_version\_id (*VariantMate attribute*), 14  
 dataset\_versions (*ReferenceSet attribute*), 10  
 DatasetAccess (*class in backend.db*), 5  
 DatasetAccessCurrent (*class in backend.db*), 6  
 DatasetAccessPending (*class in backend.db*), 6  
 DatasetFile (*class in backend.db*), 6  
 DatasetFiles (*class in backend.application*), 1  
 datasetid (*BeaconCounts attribute*), 4  
 DatasetLogo (*class in backend.db*), 7  
 datasets (*Study attribute*), 11  
 DatasetUsersCurrent (*class in backend.application*), 1  
 DatasetUsersPending (*class in backend.application*), 1  
 DatasetVersion (*class in backend.db*), 7  
 DatasetVersionCurrent (*class in backend.db*), 8  
 db\_field (*EnumField attribute*), 8  
 db\_value () (*EnumField method*), 8  
 dbnsfp\_version (*ReferenceSet attribute*), 10  
 description (*Dataset attribute*), 5  
 description (*DatasetVersion attribute*), 7  
 description (*DatasetVersionCurrent attribute*), 8  
 description (*Study attribute*), 11  
 destroySession () (*RequestTests method*), 18  
 DeveloperLoginHandler (*class in backend.auth*), 3  
 DeveloperLogoutHandler (*class in backend.auth*), 3  
 DoesNotExist (*BaseModel attribute*), 4  
 DoesNotExist (*BeaconCounts attribute*), 4  
 DoesNotExist (*Collection attribute*), 4  
 DoesNotExist (*Coverage attribute*), 4  
 DoesNotExist (*Dataset attribute*), 5  
 DoesNotExist (*DatasetAccess attribute*), 5  
 DoesNotExist (*DatasetAccessCurrent attribute*), 6  
 DoesNotExist (*DatasetAccessPending attribute*), 6  
 DoesNotExist (*DatasetFile attribute*), 6  
 DoesNotExist (*DatasetLogo attribute*), 7

- DoesNotExist (*DatasetVersion* attribute), 7
- DoesNotExist (*DatasetVersionCurrent* attribute), 8
- DoesNotExist (*Feature* attribute), 8
- DoesNotExist (*Gene* attribute), 9
- DoesNotExist (*GeneOtherNames* attribute), 9
- DoesNotExist (*Linkhash* attribute), 9
- DoesNotExist (*Metrics* attribute), 10
- DoesNotExist (*ReferenceSet* attribute), 10
- DoesNotExist (*SampleSet* attribute), 11
- DoesNotExist (*SFTPUser* attribute), 10
- DoesNotExist (*Study* attribute), 11
- DoesNotExist (*Transcript* attribute), 11
- DoesNotExist (*User* attribute), 12
- DoesNotExist (*UserAccessLog* attribute), 12
- DoesNotExist (*UserConsentLog* attribute), 13
- DoesNotExist (*UserDownloadLog* attribute), 13
- DoesNotExist (*Variant* attribute), 13
- DoesNotExist (*VariantGenes* attribute), 14
- DoesNotExist (*VariantMate* attribute), 14
- DoesNotExist (*VariantTranscripts* attribute), 15
- Download (class in *backend.modules.browser.browser\_handlers*), 20
- download\_logs (*DatasetFile* attribute), 6
- download\_logs (*User* attribute), 12
- ## E
- ElixirLoginHandler (class in *backend.auth*), 3
- ElixirLogoutHandler (class in *backend.auth*), 3
- email (*User* attribute), 12
- ensembl\_version (*ReferenceSet* attribute), 10
- EnumField (class in *backend.db*), 8
- ethnicity (*Collection* attribute), 4
- exons (*Gene* attribute), 9
- expires\_on (*Linkhash* attribute), 9
- ## F
- Feature (class in *backend.db*), 8
- feature\_type (*Feature* attribute), 8
- file\_access (*DatasetVersion* attribute), 7
- file\_access (*DatasetVersionCurrent* attribute), 8
- file\_size (*DatasetFile* attribute), 6
- files (*DatasetVersion* attribute), 7
- filter\_string (*Variant* attribute), 13
- format\_bytes () (in module *backend.application*), 3
- full\_name (*Dataset* attribute), 5
- full\_name (*Gene* attribute), 9
- ## G
- gencode\_version (*ReferenceSet* attribute), 10
- Gene (class in *backend.db*), 9
- gene (*Feature* attribute), 8
- gene (*GeneOtherNames* attribute), 9
- gene (*Transcript* attribute), 11
- gene (*VariantGenes* attribute), 14
- gene\_id (*Feature* attribute), 8
- gene\_id (*Gene* attribute), 9
- gene\_id (*GeneOtherNames* attribute), 9
- gene\_id (*Transcript* attribute), 11
- gene\_id (*VariantGenes* attribute), 14
- GeneOtherNames (class in *backend.db*), 9
- generate\_password () (*SFTPAccess* method), 3
- GenerateTemporaryLink (class in *backend.application*), 2
- genes (*ReferenceSet* attribute), 10
- genes (*Variant* attribute), 14
- get () (*AngularTemplate* method), 16
- get () (*Autocomplete* method), 20
- get () (*BaseStaticNginxFileHandler* method), 17
- get () (*Collection* method), 1
- get () (*CountryList* method), 1
- get () (*DatasetFiles* method), 1
- get () (*DatasetUsersCurrent* method), 1
- get () (*DatasetUsersPending* method), 2
- get () (*DeveloperLoginHandler* method), 3
- get () (*DeveloperLogoutHandler* method), 3
- get () (*Download* method), 20
- get () (*ElixirLoginHandler* method), 3
- get () (*ElixirLogoutHandler* method), 3
- get () (*GetCoverage* method), 20
- get () (*GetCoveragePos* method), 21
- get () (*GetDataset* method), 2
- get () (*GetGene* method), 21
- get () (*GetRegion* method), 21
- get () (*GetSchema* method), 2
- get () (*GetTranscript* method), 22
- get () (*GetUser* method), 2
- get () (*GetVariant* method), 22
- get () (*GetVariants* method), 22
- get () (*ListDatasets* method), 2
- get () (*ListDatasetVersions* method), 2
- get () (*QuitHandler* method), 2
- get () (*RequestTests* method), 18
- get () (*Search* method), 22
- get () (*ServeLogo* method), 3
- get () (*SFTPAccess* method), 3
- get () (*TemporaryStaticNginxFileHandler* method), 17
- get () (*UserDatasetAccess* method), 3
- get\_admin\_datasets () (in module *backend.db*), 15
- get\_awesomebar\_result () (in module *backend.modules.browser.lookups*), 23
- get\_coverage () (in module *backend.modules.browser.utils*), 28
- get\_coverage\_for\_bases () (in module *backend.modules.browser.lookups*), 23
- get\_coverage\_for\_transcript () (in module *backend.modules.browser.lookups*), 24

- get\_coverage\_pos() (in module *backend.modules.browser.utils*), 28
- get\_current\_user() (*BaseHandler* method), 16
- get\_dataset() (in module *backend.db*), 15
- get\_dataset\_version() (in module *backend.db*), 15
- get\_exons\_in\_transcript() (in module *backend.modules.browser.lookups*), 24
- get\_flags\_from\_variant() (in module *backend.modules.browser.utils*), 28
- get\_gene() (in module *backend.modules.browser.lookups*), 24
- get\_gene\_by\_dbid() (in module *backend.modules.browser.lookups*), 24
- get\_gene\_by\_name() (in module *backend.modules.browser.lookups*), 25
- get\_genes\_in\_region() (in module *backend.modules.browser.lookups*), 25
- get\_next\_free\_uid() (in module *backend.db*), 15
- get\_proper\_hgvs() (in module *backend.modules.browser.utils*), 28
- get\_protein\_hgvs() (in module *backend.modules.browser.utils*), 28
- get\_raw\_variant() (in module *backend.modules.browser.lookups*), 25
- get\_transcript() (in module *backend.modules.browser.lookups*), 25
- get\_transcript\_hgvs() (in module *backend.modules.browser.utils*), 28
- get\_transcripts\_in\_gene() (in module *backend.modules.browser.lookups*), 26
- get\_transcripts\_in\_gene\_by\_dbid() (in module *backend.modules.browser.lookups*), 26
- get\_user() (*ElixirLoginHandler* method), 3
- get\_user\_token() (*ElixirLoginHandler* method), 3
- get\_variant() (in module *backend.modules.browser.lookups*), 26
- get\_variant\_list() (in module *backend.modules.browser.utils*), 29
- get\_variants\_by\_rsid() (in module *backend.modules.browser.lookups*), 26
- get\_variants\_in\_gene() (in module *backend.modules.browser.lookups*), 26
- get\_variants\_in\_region() (in module *backend.modules.browser.lookups*), 27
- get\_variants\_in\_transcript() (in module *backend.modules.browser.lookups*), 27
- GetCoverage (class in *backend.modules.browser.browser\_handlers*), 20
- GetCoveragePos (class in *backend.modules.browser.browser\_handlers*), 21
- GetDataset (class in *backend.application*), 2
- GetGene (class in *backend.modules.browser.browser\_handlers*), 21
- GetRegion (class in *backend.modules.browser.browser\_handlers*), 21
- GetSchema (class in *backend.application*), 2
- GetTranscript (class in *backend.modules.browser.browser\_handlers*), 21
- getUrl() (*RequestTests* method), 18
- GetUser (class in *backend.application*), 2
- GetVariant (class in *backend.modules.browser.browser\_handlers*), 22
- GetVariants (class in *backend.modules.browser.browser\_handlers*), 22
- ## H
- has\_access (*DatasetAccessCurrent* attribute), 6
- has\_access (*DatasetAccessPending* attribute), 6
- has\_access() (*User* method), 12
- has\_image() (*Dataset* method), 5
- has\_requested\_access() (*User* method), 12
- hash (*Linkhash* attribute), 9
- hist (*Metrics* attribute), 10
- hom\_count (*Variant* attribute), 14
- HOST (*RequestTests* attribute), 18
- ## I
- id (*BaseModel* attribute), 4
- id (*Collection* attribute), 4
- id (*Coverage* attribute), 4
- id (*Dataset* attribute), 5
- id (*DatasetAccess* attribute), 5
- id (*DatasetAccessCurrent* attribute), 6
- id (*DatasetAccessPending* attribute), 6
- id (*DatasetFile* attribute), 6
- id (*DatasetLogo* attribute), 7
- id (*DatasetVersion* attribute), 7
- id (*DatasetVersionCurrent* attribute), 8
- id (*Feature* attribute), 8
- id (*Gene* attribute), 9
- id (*GeneOtherNames* attribute), 9
- id (*Linkhash* attribute), 10
- id (*Metrics* attribute), 10
- id (*ReferenceSet* attribute), 10
- id (*SampleSet* attribute), 11
- id (*SFTPUser* attribute), 10
- id (*Study* attribute), 11
- id (*Transcript* attribute), 11
- id (*User* attribute), 12
- id (*UserAccessLog* attribute), 13

id (*UserConsentLog* attribute), 13  
 id (*UserDownloadLog* attribute), 13  
 id (*Variant* attribute), 14  
 id (*VariantGenes* attribute), 14  
 id (*VariantMate* attribute), 14  
 id (*VariantTranscripts* attribute), 15  
 identity (*User* attribute), 12  
 identity\_type (*User* attribute), 12  
 initialize() (*AngularTemplate* method), 16  
 initialize() (*BaseStaticNginxFileHandler* method), 17  
 is\_admin (*DatasetAccess* attribute), 5  
 is\_admin (*DatasetAccessCurrent* attribute), 6  
 is\_admin (*DatasetAccessPending* attribute), 6  
 is\_admin() (*User* method), 12  
 is\_region\_too\_large() (in module *back-end.modules.browser.utils*), 29

## L

link\_hashes (*DatasetVersion* attribute), 7  
 link\_hashes (*User* attribute), 12  
 Linkhash (class in *backend.db*), 9  
 ListDatasets (class in *backend.application*), 2  
 ListDatasetVersions (class in *back-end.application*), 2  
 LogEvent (class in *backend.application*), 2  
 login\_user() (*RequestTests* method), 18  
 logo (*Dataset* attribute), 5

## M

mate (*DatasetVersion* attribute), 7  
 mate\_chrom (*VariantMate* attribute), 14  
 mate\_id (*VariantMate* attribute), 14  
 mate\_start (*VariantMate* attribute), 14  
 mean (*Coverage* attribute), 4  
 median (*Coverage* attribute), 4  
 metric (*Metrics* attribute), 10  
 Metrics (class in *backend.db*), 10  
 metrics\_set (*DatasetVersion* attribute), 7  
 mids (*Metrics* attribute), 10  
 mim\_annotation (*Transcript* attribute), 11  
 mim\_gene\_accession (*Transcript* attribute), 11  
 mimetype (*DatasetLogo* attribute), 7

## N

name (*Collection* attribute), 4  
 name (*DatasetFile* attribute), 6  
 name (*Gene* attribute), 9  
 name (*GeneOtherNames* attribute), 9  
 name (*ReferenceSet* attribute), 10  
 name (*User* attribute), 12  
 newSession() (*RequestTests* method), 18  
 num\_variants (*DatasetVersion* attribute), 7  
 num\_variants (*DatasetVersionCurrent* attribute), 8

## O

on\_finish() (*BaseHandler* method), 16  
 order\_vep\_by\_csq() (in module *back-end.modules.browser.utils*), 29  
 orig\_alt\_alleles (*Variant* attribute), 14  
 other\_names (*Gene* attribute), 9

## P

parse\_dataset() (in module *back-end.modules.browser.utils*), 29  
 parse\_region() (in module *back-end.modules.browser.utils*), 29  
 password\_hash (*SFTPUser* attribute), 10  
 phenotype (*SampleSet* attribute), 11  
 pi\_email (*Study* attribute), 11  
 pi\_name (*Study* attribute), 11  
 portal\_avail (*DatasetVersion* attribute), 7  
 portal\_avail (*DatasetVersionCurrent* attribute), 8  
 pos (*Coverage* attribute), 4  
 pos (*Variant* attribute), 14  
 pos (*VariantMate* attribute), 15  
 post() (*ApproveUser* method), 1  
 post() (*GenerateTemporaryLink* method), 2  
 post() (*LogEvent* method), 2  
 post() (*RequestAccess* method), 2  
 post() (*RequestTests* method), 18  
 post() (*RevokeUser* method), 3  
 post() (*SFTPAccess* method), 3  
 prepare() (*AdminHandler* method), 15  
 prepare() (*AuthorizedHandler* method), 16  
 prepare() (*BaseHandler* method), 16  
 prepare() (*SafeHandler* method), 17  
 publication\_date (*Study* attribute), 11  
 python\_value() (*EnumField* method), 8

## Q

quality\_metrics (*Variant* attribute), 14  
 QuitHandler (class in *backend.application*), 2

## R

ref (*Variant* attribute), 14  
 ref (*VariantMate* attribute), 15  
 ref\_doi (*DatasetVersion* attribute), 7  
 ref\_doi (*DatasetVersionCurrent* attribute), 8  
 ref\_doi (*Study* attribute), 11  
 reference\_build (*ReferenceSet* attribute), 10  
 reference\_set (*DatasetVersion* attribute), 7  
 reference\_set (*DatasetVersionCurrent* attribute), 8  
 reference\_set (*Gene* attribute), 9  
 reference\_set\_id (*DatasetVersion* attribute), 7  
 reference\_set\_id (*DatasetVersionCurrent* attribute), 8  
 reference\_set\_id (*Gene* attribute), 9

ReferenceSet (class in backend.db), 10  
 remove\_extraneous\_information() (in module backend.modules.browser.utils), 29  
 remove\_extraneous\_vep\_annotations() (in module backend.modules.browser.utils), 30  
 RequestAccess (class in backend.application), 2  
 RequestTests (class in backend.test), 18  
 RevokeUser (class in backend.application), 2  
 rsid (Variant attribute), 14

## S

SafeHandler (class in backend.handlers), 17  
 SafeStaticFileHandler (class in backend.handlers), 17  
 sample\_sets (Collection attribute), 4  
 sample\_sets (Dataset attribute), 5  
 sample\_size (SampleSet attribute), 11  
 SampleSet (class in backend.db), 11  
 Search (class in backend.modules.browser.browser\_handlers), 22  
 seq\_center (Dataset attribute), 5  
 seq\_tech (Dataset attribute), 5  
 seq\_type (Dataset attribute), 5  
 ServeLogo (class in backend.application), 3  
 session (RequestTests attribute), 18  
 set\_user\_msg() (BaseHandler method), 17  
 setUp() (TestAdminAccess method), 18  
 setUp() (TestLoggedInUser method), 19  
 setUp() (TestRequestAccess method), 19  
 setUp() (TestUserManagement method), 19  
 sftp\_user (User attribute), 12  
 SFTPAccess (class in backend.application), 3  
 SFTPUser (class in backend.db), 10  
 short\_name (Dataset attribute), 5  
 site\_quality (Variant attribute), 14  
 start (Feature attribute), 8  
 start (Gene attribute), 9  
 start (Transcript attribute), 11  
 stop (Feature attribute), 8  
 stop (Gene attribute), 9  
 stop (Transcript attribute), 12  
 strand (Feature attribute), 9  
 strand (Gene attribute), 9  
 strand (Transcript attribute), 12  
 Study (class in backend.db), 11  
 study (Dataset attribute), 5  
 study\_id (Dataset attribute), 5

## T

tearDown() (TestAdminAccess method), 18  
 tearDown() (TestLoggedInUser method), 19  
 tearDown() (TestRequestAccess method), 19  
 tearDown() (TestUserManagement method), 20

TemporaryStaticNginxFileHandler (class in backend.handlers), 17  
 terms (DatasetVersion attribute), 7  
 terms (DatasetVersionCurrent attribute), 8  
 test\_admin\_approve\_user() (TestUserManagement method), 20  
 test\_admin\_is\_admin() (TestAdminAccess method), 18  
 test\_admin\_list\_users() (TestAdminAccess method), 18  
 test\_admin\_list\_users\_get\_data() (TestAdminAccess method), 18  
 test\_admin\_list\_users\_only\_own\_project\_1() (TestAdminAccess method), 18  
 test\_admin\_list\_users\_only\_own\_project\_2() (TestAdminAccess method), 18  
 test\_create\_temporary\_link() (TestEndpoints method), 18  
 test\_dataset\_collection() (TestEndpoints method), 18  
 test\_dataset\_logo() (TestEndpoints method), 18  
 test\_datasets() (TestEndpoints method), 19  
 test\_full\_user\_roundabout() (TestUserManagement method), 20  
 test\_get\_one\_version() (TestEndpoints method), 19  
 test\_get\_users\_current() (TestEndpoints method), 19  
 test\_get\_users\_pending() (TestEndpoints method), 19  
 test\_get\_versions() (TestEndpoints method), 19  
 test\_get\_xsrftoken() (TestRequestAccess method), 19  
 test\_list\_files() (TestEndpoints method), 19  
 test\_login() (TestRequestAccess method), 19  
 test\_login\_admin() (TestAdminAccess method), 18  
 test\_one\_dataset() (TestEndpoints method), 19  
 test\_recently\_approved\_user\_can\_list\_files() (TestUserManagement method), 20  
 test\_recently\_revoked\_user\_cant\_list\_files() (TestUserManagement method), 20  
 test\_request\_access() (TestEndpoints method), 19  
 test\_request\_access\_correctly() (TestRequestAccess method), 19  
 test\_request\_access\_with\_get() (TestRequestAccess method), 19  
 test\_request\_access\_with\_wrong\_xsrftoken\_1() (TestRequestAccess method), 19  
 test\_request\_access\_with\_wrong\_xsrftoken\_2() (TestRequestAccess method), 19  
 test\_request\_access\_without\_xsrftoken() (TestRequestAccess method), 19

- test\_user\_datasets() (*TestEndpoints method*), 19
- test\_user\_me() (*TestEndpoints method*), 19
- TestAdminAccess (*class in backend.test*), 18
- TestEndpoints (*class in backend.test*), 18
- testLoggedInFiles() (*TestLoggedInUser method*), 19
- testLoggedInTempLinkGet() (*TestLoggedInUser method*), 19
- testLoggedInTempLinkPost() (*TestLoggedInUser method*), 19
- testLoggedInTempLinkPostXSRF1() (*TestLoggedInUser method*), 19
- testLoggedInTempLinkPostXSRF2() (*TestLoggedInUser method*), 19
- TestLoggedInUser (*class in backend.test*), 19
- TestRequestAccess (*class in backend.test*), 19
- TestUserManagement (*class in backend.test*), 19
- title (*Study attribute*), 11
- Transcript (*class in backend.db*), 11
- transcript (*Feature attribute*), 9
- transcript (*VariantTranscripts attribute*), 15
- transcript\_id (*Feature attribute*), 9
- transcript\_id (*Transcript attribute*), 12
- transcript\_id (*VariantTranscripts attribute*), 15
- transcripts (*Gene attribute*), 9
- transcripts (*Transcript attribute*), 12
- transcripts (*Variant attribute*), 14
- ts (*UserAccessLog attribute*), 13
- ts (*UserConsentLog attribute*), 13
- ts (*UserDownloadLog attribute*), 13
- ## U
- UnsafeHandler (*class in backend.handlers*), 17
- uri (*DatasetFile attribute*), 7
- User (*class in backend.db*), 12
- user (*DatasetAccess attribute*), 5
- user (*DatasetAccessCurrent attribute*), 6
- user (*DatasetAccessPending attribute*), 6
- user (*Linkhash attribute*), 10
- user (*SFTPUser attribute*), 10
- USER (*TestRequestAccess attribute*), 19
- user (*UserAccessLog attribute*), 13
- user (*UserConsentLog attribute*), 13
- user (*UserDownloadLog attribute*), 13
- user\_id (*DatasetAccess attribute*), 6
- user\_id (*DatasetAccessCurrent attribute*), 6
- user\_id (*DatasetAccessPending attribute*), 6
- user\_id (*Linkhash attribute*), 10
- user\_id (*SFTPUser attribute*), 10
- user\_id (*UserAccessLog attribute*), 13
- user\_id (*UserConsentLog attribute*), 13
- user\_id (*UserDownloadLog attribute*), 13
- user\_name (*SFTPUser attribute*), 10
- user\_uid (*SFTPUser attribute*), 11
- UserAccessLog (*class in backend.db*), 12
- UserConsentLog (*class in backend.db*), 13
- UserDatasetAccess (*class in backend.application*), 3
- UserDownloadLog (*class in backend.db*), 13
- ## V
- Variant (*class in backend.db*), 13
- variant (*VariantGenes attribute*), 14
- variant (*VariantTranscripts attribute*), 15
- variant\_id (*Variant attribute*), 14
- variant\_id (*VariantGenes attribute*), 14
- variant\_id (*VariantMate attribute*), 15
- variant\_id (*VariantTranscripts attribute*), 15
- variantcount (*BeaconCounts attribute*), 4
- VariantGenes (*class in backend.db*), 14
- VariantMate (*class in backend.db*), 14
- variants (*DatasetVersion attribute*), 7
- variants (*Gene attribute*), 9
- variants (*Transcript attribute*), 12
- VariantTranscripts (*class in backend.db*), 15
- vep\_annotations (*Variant attribute*), 14
- version (*DatasetVersion attribute*), 8
- version (*DatasetVersionCurrent attribute*), 8
- versions (*Dataset attribute*), 5
- ## W
- wants\_newsletter (*DatasetAccess attribute*), 6
- wants\_newsletter (*DatasetAccessCurrent attribute*), 6
- wants\_newsletter (*DatasetAccessPending attribute*), 6
- worst\_csq\_from\_csq() (*in module backend.modules.browser.utils*), 30
- worst\_csq\_from\_list() (*in module backend.modules.browser.utils*), 30
- worst\_csq\_index() (*in module backend.modules.browser.utils*), 30
- worst\_csq\_with\_vep() (*in module backend.modules.browser.utils*), 30
- write() (*BaseHandler method*), 17
- write\_error() (*BaseHandler method*), 17