grano Documentation

Release 0.5

Code for Africa

December 14, 2014

Contents

This site only contains documentation related to the internal API of grano. If you want to learn more about the project, we recommend you visit the project home page.

Contents:

CHAPTER 1

Bulk data loading API

1.1 Loader

class grano.logic.loader.**Loader**(*project_slug*, *source_url=None*, *project_label=None*, *project_settings=None*, *account=None*, *ignore_errors=True*)

A loader is a factory object that can be used to make entities and relations in the database. It will perform some validation and handle database transactions.

make_entity(*schema*, *source_url=None*)

Create an entity loader, i.e. a construction helper for entities.

Parameters

- **schema** – The schema that the entity should be associated with.
- **source_url** – A URL which will be made the default source for all properties defined on this entity.

Returns EntityLoader

make_relation(*schema*, *source*, *target*, *source_url=None*)

Create a relation loader, i.e. a construction helper for relations.

Parameters

- **schema** – A schema name for the relation.
- **source** – An EntityLoader which has been used to construct the source entity.
- **target** – A second EntityLoader which has been used to construct the target entity. Cannot be identical to the source.
- **source_url** – A URL which will be made the default source for all properties defined on this entity.

Returns RelationLoader

persist()

Save the created entiites and relations, i.e. commit the database transaction.

1.2 EntityLoader

class grano.logic.loader.**EntityLoader**(*loader*, *schema*, *source_url=None*)

A factory object for entities, used to set the schemata and properties for an entity.

save()
Save the entity to the database. Do this only once, after all properties have been set.

set(*name*, *value*, *source_url=None*)
Set the value of a given property, optionally by attributing a source URL.

Parameters

- **name** – The property name. This must be defined as part of one of the schemata that the entity or relation is associated with.
- **value** – The value to be set for this property. If it is `None`, the property will not be set, but existing values of will be marked as inactive.
- **source_url** – A URL which will be set as the origin of this information.

unique(*name*, *only_active=True*)
Define a unique field for this entity or relation. Each unique key will be used to decide whether a record already exists and can be updated, or whether a new one must be created.

Parameters

- **name** – The property name of the unique field.
- **only_active** – If set to `False`, the check will include all historic values of the property as well as the current value.

1.3 RelationLoader

class grano.logic.loader.**RelationLoader**(*loader*, *schema*, *source*, *target*, *source_url=None*)
A factory object for relations, used to construct a relation by setting its schema, source entity, target entity and a set of properties.

save()
Save the relation to the database. Do this only once, after all properties have been set.

set(*name*, *value*, *source_url=None*)
Set the value of a given property, optionally by attributing a source URL.

Parameters

- **name** – The property name. This must be defined as part of one of the schemata that the entity or relation is associated with.
- **value** – The value to be set for this property. If it is `None`, the property will not be set, but existing values of will be marked as inactive.
- **source_url** – A URL which will be set as the origin of this information.

unique(*name*, *only_active=True*)
Define a unique field for this entity or relation. Each unique key will be used to decide whether a record already exists and can be updated, or whether a new one must be created.

Parameters

- **name** – The property name of the unique field.
- **only_active** – If set to `False`, the check will include all historic values of the property as well as the current value.

CHAPTER 2

Plugin extension points

2.1 System startup

class grano.interface.**Startup**

This interface will be called when grano is started and allows plugins to register additional functionality such as flask views.

configure(*manager*)

Run this on startup.

2.2 Entity changes

class grano.interface.**EntityChangeProcessor**

An entity change processor gets notified whenever there is a change to an entity so that it can perform related actions.

This may happen out of band (ie. on a queue or batch job), thus changes may not be applied immediately.

entity_changed(*entity_id*, *operation*)

Notify the plugin that an entity has changed. The plugin will only receive the ID and must query for the object itself.

2.3 Relation changes

class grano.interface.**RelationChangeProcessor**

A relation change processor gets notified whenever there is a change to a relation so that it can perform related actions.

This may happen out of band (ie. on a queue or batch job), thus changes may not be applied immediately.

relation_changed(*relation_id*, *operation*)

Notify the plugin that a relation has changed. The plugin will only receive the ID and must query for the object itself.

2.4 Project changes

class grano.interface.**ProjectChangeProcessor**

A project change processor gets notified whenever there is a change to a project's settings so that it can perform related actions.

This may happen out of band (ie. on a queue or batch job), thus changes may not be applied immediately.

project_changed(*project_slug*, *operation*)

Notify the plugin that a project has changed. The plugin will only receive the ID and must query for the object itself.

CHAPTER 3

Indices and tables

- *genindex*
- *modindex*
- *search*

C

E

L

M

P

R

S

U