SeSQL administration guide

Contents

1 Django administration commands

1.1 syncdb

Django `syncdb` command will create the SeSQL tables that were not created yet. SeSQL **must** be configured before running it.

1.2 createsqltables

This command will output the SQL code to create the various tables required by SeSQL, with `DROP` statements if needed.

Since there is no support for upgrading schemas in SeSQL currently, you may have to use this command and manually edit the output if you do changes in SeSQL configuration.

If you have a lot of content to index into SeSQL at once, you may also try to create the tables without the `CREATE INDEX` statements, insert the data, and then run the `CREATE INDEX` statements.

1.3 sesqlindex

This command takes a class name and an id as parameters, and will reindex into SeSQL the given object.

1.4 sesqlreindex

This command takes only a class name as parameter, and will index into SeSQL all objects that exist in Django tables but that are not yet into SeSQL. It is very useful if you just installed SeSQL on an existing database.

This command may take a long time and use a lot of memory. You can interrupt it and restart it later on, it'll continue where it stopped (objects are reindexed by groups of 1000, only the current group will be restarted).

Note : sesqlreindex will not consider the `SKIP_CONDITION` in its stats, so you may be in a situation where sesqlreindex will constantly claim that it needs to reindex objects, but not change anything when executed. The `SKIP_CONDITION` is only checked when an individual object is being indexed (whatever the source of indexation), not when stats are calculated.

1.5 sesqlasyncreindex

This command will reindex all your content into a new set of tables, allowing for configuration change. It is devised to not impact significantly your production. It is slightly delicate to use, and is fully documented in its own documentation file.

1.6 sesqlupdate

This command will massively reindex a single (or a coupe of) column. It will not index any new object. Typical use is when a new column was added.

1.7 sesqlshortquery

Perform a short query. The first parameter is a Python expression describing a Q object, the second, optional parameter, a Python expression specifying the sort order to use.

Warning : this command uses `eval()` to process the Python expressions, it is therefore unsafe, and should only be used for testing purpose. Never expose this command to untrusted input.

1.8 sesqllongquery

Like the previous command, but performing a long query.

1.9 sesqlfixatedate

A tool to recopy a field to another field when the target is NULL. Useful when a date is only existing in some rows, and you want it to default to another date field, in a one-shot operation.

2 SQL level administration

SeSQL contains three kinds of tables :

- The *master table* : this table doesn't contain any actual content, but can be requested (with regular `SELECT`) to query all indexed content.

- The *data tables* : those tables are defined in the type map, and contain the actual content; they all inherit from the *master table*. Data can be inserted, modified and deleted from those tables.
- The *utility tables* : they are used for features like search history or dependency tracking. Refer to each features for on overview of its tables.