
django-aggtrigg Documentation

Release 0.6.1

Rodolphe Quiédeville

December 12, 2014

1	Quick start	3
2	Manage triggers and related objects	5
3	Howto use the new aggregat	7
4	What inside	9
4.1	Contents	9
4.2	Indices and tables	12

Create triggers to do some aggregate and permit to count objects from database without using COUNT() aggregat.
Detailed documentation is in the “docs” directory.

Quick start

1. Add “django_aggtrigg” to your INSTALLED_APPS setting like this:

```
INSTALLED_APPS = (  
    ...  
    'django_aggtrigg',  
)
```

2. Import fields in your models:

```
from django_aggtrigg.models import IntegerTriggerField  
from django_aggtrigg.models import FloatTriggerField
```

3. Configure your fields as is:

```
class Apple(models.Model):  
    indice = IntegerTriggerField(default=0)  
    indice.aggregate_trigger=['count', 'min']  
  
    mark = FloatTriggerField(default=0)  
    mark.aggregate_trigger=['min']
```

By default only the *count* aggregat will be created.

4. Use the new manager on you Model

```
objects = AggTriggManager()
```

Manage triggers and related objects

To create the triggers in the database do:

```
python manage.py aggtrigg_create
```

Dropping triggers is easy as doing:

```
python manage.py aggtrigg_drop | psql -d DATABASE NAME
```

For safety reason the drop command just ouptput on stdout the SQL statements.

To initialize the aggregate table, you can fill it by hand or do:

```
python manage.py aggtrigg_initialize
```

Howto use the new aggregat

Instead of doing a COUNT as the traditionnal way:

```
Apple.objects.filter(indice=42).count()
```

you will do:

```
Apple.objects.optimized_count(indice=42)
```

This is may be less easy, but so much more efficient when you manipulate billions or tuples in your relations.

What inside

The class **apple** was create in the app called **foo** so the default name of the table that contains data will be **foo_apple**, we use the tablename from the Model so if it's changed in **Meta** will still be compliant.

A new table **foo_apple__indice_agg** is created in the same database as **foo_apple**, it will contain the aggregat:

```
foo=# \d foo_apple__indice_agg
Table "public.foo_apple__indice_agg"
  Column    | Type    | Modifiers
-----+-----+-----
 indice     | integer |
 agg_count  | integer |
 agg_min    | integer |
Indexes:
    "foo_apple__indice_agg_indice_idx" btree (indice)
```

4.1 Contents

4.1.1 Install

django-aggtrigg is open-source software, published under BSD license. See [License](#) for details.

If you want to install a development environment, you should go to [Contributing](#) documentation.

Prerequisites

- [Python](#)¹ 2.7, 3.3 or 3.4. Other versions may work, but they are not part of the test suite at the moment.

As a library

In most cases, you will use *django-aggtrigg* as a dependency of another project. In such a case, you should add *django-aggtrigg* in your main project's requirements. Typically in `setup.py`:

```
from setuptools import setup

setup(
    install_requires=[
        'django-aggtrigg',
```

¹ <https://www.python.org/>

```
        # ...
    ]
    # ...
)
```

Then when you install your main project with your favorite package manager (like [pip](#)²), *django-aggtrigg* will automatically be installed.

Standalone

You can install *django-aggtrigg* with your favorite Python package manager. As an example with [pip](#)²:

```
pip install django-aggtrigg
```

Check

Check *django-aggtrigg* has been installed:

```
python -c "import django_aggtrigg;print(django_aggtrigg.__version__)"
```

You should get *django_aggtrigg*'s version.

References

4.1.2 About django-aggtrigg

This section is about the *django-aggtrigg* project itself.

Vision

django-json-dbindex helps you integrate and use complex indexes within a Django project.

License

Copyright (c) 2014, Rodolphe Quiédeville. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of *django-aggtrigg* nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT

² <https://pypi.python.org/pypi/pip/>

SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Authors & contributors

Maintainer: Rodolphe Quiédeville <rodolphe@quiedeville.org>, as a member of the [PeopleDoc](https://github.com/novapost/)³ team: <https://github.com/novapost/>

Developers: <https://github.com/novapost/django-aggtrigg/graphs/contributors>

Notes & references

Changelog

This document describes changes between each past release. For information about future releases, check [milestones](#)⁴ and *Vision*.

0.0.1 (2014-12-01)

Initial release.

- First release of django-aggtrigg, compatibility with PostgreSQL only.

Notes & references

4.1.3 Contributing

This document provides guidelines for people who want to contribute to the *django-aggtrigg* project.

Create tickets

Please use [django-aggtrigg bugtracker](#)⁵ **before** starting some work:

- check if the bug or feature request has already been filed. It may have been answered too!
- else create a new ticket.
- if you plan to contribute, tell us, so that we are given an opportunity to give feedback as soon as possible.
- Then, in your commit messages, reference the ticket with some `refs #TICKET-ID` syntax.

³ <http://www.people-doc.com>

⁴ <https://github.com/novapost/django-aggtrigg/milestones>

⁵ <https://github.com/novapost/django-aggtrigg/issues>

Use topic branches

- Work in branches.
- Please never push in `master` directly.
- Prefix your branch with one the following keyword `feature/` when adding a new feature and `fix/` when working on a fix. You can also add the ticket ID corresponding to the issue to be explicit.
- If you work in a development branch and want to refresh it with changes from master, please `rebase`⁶ or `merge-based rebase`⁷, i.e. do not merge master.

Fork, clone

Clone *django-aggtrigg* repository (adapt to use your own fork):

```
git clone git@github.com:novapost/django-aggtrigg.git
cd django-aggtrigg/
```

Usual actions

The *Makefile* is the reference card for usual actions in development environment:

- Install development toolkit with `pip`⁸: `make develop`.
- Run tests with `tox`⁹: `make test`.
- Build documentation: `make documentation`. It builds `Sphinx`¹⁰ documentation in `var/docs/html/index.html`.
- Release *django-aggtrigg* project with `zest.releaser`¹¹: `make release`.
- Cleanup local repository: `make clean`, `make distclean` and `make maintainer-clean`.

See also `make help`.

Notes & references

4.2 Indices and tables

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

⁶ <http://git-scm.com/book/en/Git-Branching-Rebasing>

⁷ <http://tech.novapost.fr/psycho-rebasing-en.html>

⁸ <https://pypi.python.org/pypi/pip/>

⁹ <https://pypi.python.org/pypi/tox/>

¹⁰ <https://pypi.python.org/pypi/Sphinx/>

¹¹ <https://pypi.python.org/pypi/zest.releaser/>