
celery-haystack Documentation

Release 0.9

Jannis Leidel

Sep 27, 2017

Contents

1	Requirements	3
2	Installation	5
3	Usage	7
3.1	Haystack 1.X	7
3.2	Haystack 2.X	7
4	Thanks	9
5	Issues	11
5.1	Changelog	11

This Django app allows you to utilize Celery for automatically updating and deleting objects in a [Haystack](#) search index.

CHAPTER 1

Requirements

- Django 1.4+
- Haystack 1.2.X or 2.X
- Celery 3.X

You also need to install your choice of one of the supported search engines for Haystack and one of the supported backends for Celery.

Use your favorite Python package manager to install the app from PyPI, e.g.:

```
pip install celery-haystack
```

By default a few dependencies will automatically be installed:

- [django-appconf](#) – An app to gracefully handle application settings.
- [django-celery-transactions](#) – An app that “holds on to Celery tasks until the current database transaction is committed, avoiding potential race conditions as described in [Celery’s user guide](#).”

Haystack 1.X

1. Add 'celery_haystack' to the INSTALLED_APPS setting

```
INSTALLED_APPS = [  
    # ..  
    'celery_haystack',  
]
```

2. Alter all of your SearchIndex subclasses to inherit from celery_haystack.indexes.CelerySearchIndex

```
from haystack import site, indexes  
from celery_haystack.indexes import CelerySearchIndex  
from myapp.models import Note  
  
class NoteIndex(CelerySearchIndex):  
    text = indexes.CharField(document=True, model_attr='content')  
  
site.register(Note, NoteIndex)
```

3. Ensure your Celery instance is running.

Haystack 2.X

1. Add 'celery_haystack' to the INSTALLED_APPS setting

```
INSTALLED_APPS = [  
    # ..  
    'celery_haystack',  
]
```

2. Enable the celery-haystack signal processor in the settings

```
HAYSTACK_SIGNAL_PROCESSOR = 'celery_haystack.signals.CelerySignalProcessor'
```

3. Alter all of your SearchIndex subclasses to inherit from celery_haystack.indexes.CelerySearchIndex and haystack.indexes.Indexable

```
from haystack import indexes
from celery_haystack.indexes import CelerySearchIndex
from myapp.models import Note

class NoteIndex(CelerySearchIndex, indexes.Indexable):
    text = indexes.CharField(document=True, model_attr='content')

    def get_model(self):
        return Note
```

4. Ensure your Celery instance is running.

CHAPTER 4

Thanks

This app is a blatant rip-off of Daniel Lindsley's [queued_search](#) app but uses Ask Solem Hoel's [Celery](#) instead of the equally awesome [queues](#) library by Matt Croyden.

Please use the [Github issue tracker](#) for any bug reports or feature requests.

Contents:

Changelog

v0.9 (2015-06-13)

- Moved to Haystack GitHub org: <https://github.com/django-haystack/celery-haystack>
- Fix handling the default Haystack backend alias, making it a list.
- Added `CELERY_HAYSTACK_QUEUE` setting to define which Celery queue to use.
- Added `CELERY_HAYSTACK_COUNTDOWN` setting to define when to start the indexing task after initially creating it.
- Stop returning after after enqueueing in the Haystack router to support multiple routers.
- Optionally support using `django-transaction-hooks` for improved transaction handling.
- Instantiate update task class correctly.
- Use Celery's task logger utility function.

v0.8 (2014-07-31)

- Fix bug when using multiple Haystack indexes
- Fixed merge bug where primary key of object was cast to int
- Add compatibility for Python 3.3, 3.4, Celery 3.X

v0.7.2 (2013-03-23)

- Fixed import time issue with Haystack 2.X.
- Minor fixes to the README.
- Made signal processor compatible for subclassing for easier extensibility.

v0.7.1 (2013-03-09)

- Fixed an installation issues with d2to1.

v0.7 (2013-03-09)

- **Backwards incompatible** change to support the new signal processor API in Haystack 2.X. To upgrade simply add this to your settings:

```
HAYSTACK_SIGNAL_PROCESSOR = 'celery_haystack.signals.CelerySignalProcessor'
```

Many thanks to Stefan Wehrmeyer for the help.

- Simplified index class implementation.
- Support multiple indexes in the task. Thanks, Stefan Wehrmeyer.
- Use the exception handler of the task logger instead of the error handler when catching an exception.
- Switched to [d2to1](#) for handling package metadata.

v0.6.2 (2012-06-28)

- Fixed AttributeError in settings handling.

v0.6.1 (2012-06-27)

- Fixed logging setup.

v0.6 (2012-06-27)

- **backwards incompatible change**

Added support for [django-celery-transactions](#) to make sure the tasks are respecting Django's transaction management. It holds on to Celery tasks until the current database transaction is committed, avoiding potential race conditions as described in [Celery's user guide](#).

This is **enabled by default** but can be disabled in case you want to manually manage the transactions:

```
CELERY_HAYSTACK_TRANSACTION_SAFE = False
```

- Refactored the error handling to always return a message about what happened in every step of the index interaction. Raises exception about misconfiguration and wrong parameters quicker.
- Improved support for multiple search indexes as implemented by Haystack 2.X. Many thanks to Germán M. Bravo (Kronuz).

v0.5 (2012-05-23)

- Moved repository to personal account again: <http://github.com/jezdez/celery-haystack>
- Removed versiontools dependency again.
- Moved to using Travis-CI for tests: <http://travis-ci.org/jezdez/celery-haystack>

v0.4 (2011-09-17)

- Fixed bug which caused the deletion of an item to not happen correctly. Please rebuild your Haystack indexes using the `rebuild_index` management command.
- Added initial Sphinx documentation: <http://celery-haystack.rtd.org>
- Revamped the tests to test the search results, not only queuing.

v0.3.1 (2011-08-22)

- Minor bugfix in new appconf support code.

v0.3 (2011-08-22)

- Moved configuration defaults handling to `django-appconf`.
- Fixed issue that occurred when retrying a task.

v0.2.1 (2011-08-05)

- Fixed typo in exception message handling.

v0.2 (2011-08-04)

- Added support for Haystack 1.2.X.
- Properly stop indexing if instance couldn't be found.
- Forced Celery task config values to be of the correct type.

v0.1.2 (2011-07-29) and v0.1.3 (2011-08-01)

- Removed stale print statement.

v0.1.1 (2011-07-29)

- Fixed packaging issue (added manifest template).

v0.1 (2011-07-29)

- Initial release.