django$_{c}elery_{r}esults Documentation

Release 1.0.4

Ask Solem

Nov 13, 2018

Contents

Version 1.0.4

Web http://django-celery-results.readthedocs.io/

Download http://pypi.python.org/pypi/django-celery-results

Source http://github.com/celery/django-celery-results

Keywords django, celery, database, results

CHAPTER 1

About

This extension enables you to store Celery task results using the Django ORM.

It defines a single model (`django_celery_results.models.TaskResult`) used to store task results, and you can query this database table like any other Django model.

CHAPTER 2

Installing

The installation instructions for this extension is available from the Celery documentation:

http://docs.celeryproject.org/en/latest/django/first-steps-with-django.html#django-celery-results-using-the-django-orm-cache-as-a-resu

CHAPTER 3

Contents

3.1 Copyright

django-celery-results User Manual

by Ask Solem

Note: While the django-celery-results *documentation* is offered under the Creative Commons *Attribution-ShareAlike 4.0 International* license the django-celery-results *software* is offered under the BSD License (3 Clause)

3.2 API Reference

Release 1.0

Date Nov 13, 2018

3.2.1 `django_celery_results.backends`

class `django_celery_results.backends.`**`CacheBackend`**(**args*, ***kwargs*)

Backend using the Django cache framework to store task metadata.

cache_backend

decode(*data*)

delete(*key*)

encode(*data*)

get(*key*)

set(*key*, *value*)

class django_celery_results.backends.**DatabaseBackend**(*app*, *serializer=None*, *max_cached_results=None*, *accept=None*, *expires=None*, *expires_type=None*, *url=None*, ***kwargs*)

The Django database backend, using models to store task state.

TaskModel
alias of *django_celery_results.models.TaskResult*

cleanup()
Delete expired metadata.

decode_content(*obj*, *content*)

encode_content(*data*)

subpolling_interval = 0.5

3.2.2 django_celery_results.backends.database

class django_celery_results.backends.database.**DatabaseBackend**(*app*, *serializer=None*, *max_cached_results=None*, *accept=None*, *expires=None*, *expires_type=None*, *url=None*, ***kwargs*)

The Django database backend, using models to store task state.

TaskModel
alias of *django_celery_results.models.TaskResult*

cleanup()
Delete expired metadata.

decode_content(*obj*, *content*)

encode_content(*data*)

subpolling_interval = 0.5

3.2.3 django_celery_results.backends.cache

Celery cache backend using the Django Cache Framework.

class django_celery_results.backends.cache.**CacheBackend**(**args*, ***kwargs*)
Backend using the Django cache framework to store task metadata.

cache_backend

decode(*data*)

delete(*key*)

encode(*data*)

get(*key*)

set(*key*, *value*)

3.2.4 django_celery_results.models

Database models.

class django_celery_results.models.**TaskResult**(**args*, ***kwargs*)

Task result/status.

exception DoesNotExist

exception MultipleObjectsReturned

as_dict()

content_encoding

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

content_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

date_done

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_next_by_date_done(***morekwargs*)

get_previous_by_date_done(***morekwargs*)

get_status_display(***morekwargs*)

hidden

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

meta

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django_celery_results.managers.TaskResultManager object>

result

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

status

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

task_args
A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

task_id
A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

task_kwargs
A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

task_name
A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

traceback
A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

3.2.5 django_celery_results.managers

Model managers.

class django_celery_results.managers.**TaskResultManager**
Manager for `celery.models.TaskResult` models.

connection_for_read()

connection_for_write()

current_engine()

delete_expired(*expires*)
Delete all expired results.

get_all_expired(*expires*)
Get all expired task results.

get_task(*task_id*)
Get result for task by `task_id`.

Keyword Arguments **exception_retry_count** (*int*) – How many times to retry by transaction rollback on exception. This could happen in a race condition if another worker is trying to create the same task. The default is to retry once.

store_result(***kwargs*)
Store the result and status of a task.

Parameters

- **content_type** (*str*) – Mime-type of result and meta content.
- **content_encoding** (*str*) – Type of encoding (e.g. binary/utf-8).
- **task_id** (*str*) – Id of task.
- **task_name** (*str*) – Celery task name.
- **task_args** (*str*) – Task arguments.
- **task_kwargs** (*str*) – Task kwargs.

- **result** (*str*) – The serialized return value of the task, or an exception instance raised by the task.
- **status** (*str*) – Task status. See `celery.states` for a list of possible status values.

Keyword Arguments

- **traceback** (*str*) – The traceback string taken at the point of exception (only passed if the task failed).
- **meta** (*str*) – Serialized result meta data (this contains e.g. children).
- **exception_retry_count** (*int*) – How many times to retry by transaction rollback on exception. This could happen in a race condition if another worker is trying to create the same task. The default is to retry twice.

warn_if_repeatable_read()

exception django_celery_results.managers.**TxIsolationWarning**
Warning emitted if the transaction isolation level is suboptimal.

django_celery_results.managers.**transaction_retry**(*max_retries=1*)
Decorate a function to retry database operations.

For functions doing database operations, adding retrying if the operation fails.

Keyword Arguments **max_retries** (*int*) – Maximum number of retries. Default one retry.

3.2.6 django_celery_results.utils

Utilities.

django_celery_results.utils.**now**()
Return the current date and time.

3.3 Change history

3.3.1 1.0.4

release-date 2018-11-12 19:00 p.m. UTC+2:00

release-by Omer Katz

1.0.3 is broken. Use 1.0.4

- Revert renaming label as it is a breaking change.

3.3.2 1.0.3

release-date 2018-11-12 18:00 p.m. UTC+2:00

release-by Omer Katz

- Revert renaming label as it is a breaking change.

3.3.3 1.0.2

release-date 2018-11-12 18:00 p.m. UTC+2:00

release-by Omer Katz

- **Store task name, args, kwargs as part of the task results in database.** Contributed by :github_user: *wardal*.
- **Admin screen changes - task name filter, search on task_name, task_id, status.** Contributed by :github_user: *jaylynch*.
- Added default_app_config.
- Added missing migration.
- Fix MySQL max length issue.
- Drop support for Django<1.11.

3.3.4 1.0.1

release-date 2016-11-07 02:00 p.m. PST

release-by Ask Solem

- Migrations were not being installed as part of the distribution (Issue #4).
- Now includes simple task result admin interface.

  Contributed by @zeezdev.
- Now depends on Celery 4.0.0.

3.3.5 1.0.0

release-date 2016-09-08 03:19 p.m. PDT

release-by Ask Solem

- Initial release

3.4 Glossary

term Description of term

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

d

Index

H

I

M

N

O

R

S

T

W