
Celery Tracker Documentation

Release 0.3.2

Tatsuo Ikeda

March 13, 2015

1	Installation	3
1.1	celeryconfig and Django settings module.	3
2	Usage	5
2.1	With Celery Extention	5
2.2	With Django	5
3	Plugins	7
3.1	Configuration file	7
3.2	Reference	8
4	Tracking	9
4.1	Reference	9
5	Customize	11
5.1	Customize plugin	11
6	Change History	13
6.1	0.3.2 (2012-12-31)	13
6.2	0.2.5 (2012-09-12)	13
6.3	0.2.4 (2012-09-11)	13
7	Celery Tracker	15
7.1	ZabbixPlugin	15
7.2	FluentPlugin	15
7.3	ReceivePlugin	15
7.4	Customize	15
8	Setup	17
8.1	Examples	17
9	License	19
10	Indices and tables	21

Contents:

Installation

Setup

```
$ pip install celery-tracker
```

1.1 celeryconfig and Django settings module.

Create celeryconfig.py, Or edit to settings.py

```
# Broker - amqp,
BROKER_URL = "amqp://guest@127.0.0.1/"

# Path to file storage. (default: memory), e.g. /tmp/tracker.db
CELERY_TRACKER_STORAGE = "/tmp/tracker.db"

# Log Level
CELERY_TRACKER_LOG_LEVEL = "INFO"

# Plugins
CELERY_TRACKER_PLUGINS = {
    "fluent": {
        "class": "tracker.plugins.fluent.FluentPlugin",
        "verbose": 0,
        "interval": 20,
        "tag": "celery.tracker",
        "host": "127.0.0.1",
        "port": 24224
    },
    "zabbix": {
        "class": "tracker.plugins.zabbix.ZabbixPlugin",
        "verbose": 0,
        "interval": 20,
        "tag": "celery.tracker",
        "host": "127.0.0.1",
        "port": 10051,
        "metrics": [
            {"host": "celery-agent"},
        ]
    },
}
```

```
"receive": {
    "class": "tracker.plugins.receive.ReceivePlugin",
    "verbose": 0,
    "tag": "celery.tracker",
    "host": "0.0.0.0",
    "port": 27015,
},
#"logging": {
#    "class": "tracker.plugins.logging.LoggingPlugin",
#    "tag": "celery.tracker",
#    "interval": 10,
#    "verbose": True
#},
}
```

Usage

Start commands

```
$ tracker
tracker 0.2.3 is starting.
Configuration ->
  . broker -> amqp://guest@127.0.0.1:5672//
  . webserver -> http://localhost:12201
tracker has started.
```

2.1 With Celery Extention

```
$ celery tracker --config=celeryconfig
tracker 0.3.1 is starting.
Configuration ->
  . broker -> amqp://guest@127.0.0.1:5672//
  . webserver -> http://localhost:12201
tracker has started.
```

2.2 With Django

```
$ python manage.py celery tracker --settings=project.settings --pythonpath=~/.virtualenvs/project/
```

Plugins

Booting plugins

- Zabbix - Zabbix threading.Thread
- Fluent - Flutnd threading.Thread
- Receive - Receive threading.Thread

3.1 Configuration file

```
CELERY_TRACKER_PLUGINS = {
    "fluent": {
        "class": "tracker.plugins.fluent.FluentPlugin",
        "verbose": 0,
        "interval": 20,
        "tag": "celery.tracker",
        "host": "127.0.0.1",
        "port": 24224
    },
    "zabbix": {
        "class": "tracker.plugins.zabbix.ZabbixPlugin",
        "verbose": 0,
        "interval": 20,
        "tag": "celery.tracker",
        "host": "127.0.0.1",
        "port": 10051,
        "metrics": [
            {"host": "celery-agent"},
        ]
    },
    "receive": {
        "class": "tracker.plugins.receive.ReceivePlugin",
        "verbose": 0,
        "tag": "celery.tracker",
        "host": "0.0.0.0",
        "port": 27015,
    },
    # "logging": {
    #     "class": "tracker.plugins.logging.LoggingPlugin",
    #     "tag": "celery.tracker",
    #     "interval": 10,
    #     "verbose": True
    # }
```

```
}  
    # },
```

3.2 Reference

3.2.1 Base

3.2.2 Zabbix

3.2.3 Fluent

3.2.4 Receive

Tracking

4.1 Reference

4.1.1 Storage

4.1.2 State

Customize

If you send your original message. See below.

5.1 Customize plugin

Inherit BasePlugin, Define a MyPlugin class.

```
"""
Customize Plugin
"""

from tracker.plugins.base import BasePlugin
import MySender

class MyPlugin(BasePlugin):
    """ Plugin for Mine. """

    def __init__(self, host, port, **kwargs):
        super(MyPlugin, self).__init__(**kwargs)

        self.host = host
        self.port = port
        self.sender = MySender(
            tag=self.tag, host=self.host, port=self.port)

    def running(self):
        """ implements method """
        event = self.storage.event("my") # CELERY_TRACKER_PLUGINS key.

        workers_average = event and event["workers_average"]
        self.sender.send(workers_average)

        tasks_average = event and event["tasks_average"]
        self.sender.send(tasks_average)

        self._logging(workers_average)
        self._logging(tasks_average)
        if self.verbose > 0:
            self._logging(event, "debug")

    def _logging(self, event, level="info"):
        logger = getattr(self.logger, level)
```

```
logger("MyPlugin/%s:%s tag: %s event: %r " % (
    self.host, self.port, self.tag, event))
```

And set the MyPlugin into configuration file. [@celeryconfig.py, @settings.py]

```
# plugins
CELERY_TRACKER_PLUGINS = {
    "fluent": {
        "class": "tracker.plugins.fluent.FluentPlugin",
        "verbose": 0,
        "interval": 20,
        "tag": "celery.tracker",
        "host": "127.0.0.1",
        "port": 24224
    },
    "zabbix": {
        "class": "tracker.plugins.zabbix.ZabbixPlugin",
        "verbose": 0,
        "interval": 20,
        "tag": "celery.tracker",
        "host": "127.0.0.1",
        "port": 10051,
        "metrics": [
            {"host": "celery-agent"},
        ]
    },
    "my": {
        "class": "path.to.MyPlugin",
        "verbose": 0
        "interval": 15,
        "tag": "celery.tracker",
        "host": "127.0.0.1",
        "port": 10101,
    },
}
```

Change History:

Change History

6.1 0.3.2 (2012-12-31)

- Add Tornado application.

6.2 0.2.5 (2012-09-12)

- PEP386-compliant

6.3 0.2.4 (2012-09-11)

- First release.

Source(README):

Celery Tracker

Receive/Sending event tracking data for the [Celery](#).

7.1 ZabbixPlugin

Send

Receive

7.2 FluentPlugin

Send

Receive

7.3 ReceivePlugin

JSON response, e.g. at <http://127.0.0.1:27015>

7.4 Customize

[Customize Documentation - celery-tracker.rtfd.org](#)

Setup

PyPI

```
$ pip install celery-tracker
```

GitHub

github.com/ikeikeikeike/celery-tracker

8.1 Examples

For more information, please see the [Example](#)

License

MIT License

Indices and tables

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