aristeinberg / celery Public

forked from celery/celery

Code Pull requests Actions Projects Wiki Security In

master 32 branches 145 tags Go to file Code

This branch is up to date with master. Contribute

607567d on Apr 28, 2013 8,119 commits

celery	Fixes tests	8 years ago
docs	New signals: a...	8 years ago
exa...	TaskProducer ...	8 years ago
extra	removed cento...	8 years ago
funt...	New signals: a...	8 years ago
requ...	Now depends ...	8 years ago
.cov...	Skip pypy stuff.	8 years ago
.giti...	Adds .coverag...	8 years ago
.trav...	Travis: Only ge...	8 years ago
CON...	Rendered CON...	8 years ago
CON...	Add Brian Bout...	8 years ago
Cha...	Changelog cos...	8 years ago
LICE...	LICENSE: Fixe...	8 years ago

About

Distributed Task Queue (development branch)

celeryproject.org

Readme

View license

0 stars

1 watching

4.2k forks

Releases

145 tags

Packages

No packages published

MAN...	Distribution cl...	8 years ago
REA...	Master branch...	8 years ago
TODO	Distribution cl...	8 years ago
pave...	releaseok buil...	8 years ago
setu...	Now depends ...	8 years ago
setu...	[3.2] Requires ...	8 years ago
tox.ini	[3.2] Requires ...	8 years ago

README.rst

celery - Distributed Task Queue

Version:	3.2.0a1 (Cipater)
Web:	http://celeryproject.c
Download:	http://pypi.python.or
Source:	http://github.com/cel
Keywords:	task queue, job queu asynchronous, async amqp, redis, python, queue, distributed

--

What is a Task Queue?

Task queues are used as a mechanism to distribute work across threads or machines.

A task queue's input is a unit of work, called a task, dedicated worker processes then constantly monitor the queue for new work to perform.

Celery communicates via messages, usually using a broker to mediate between clients and workers. To initiate a task a client puts a message on the queue, the broker then delivers the message to a worker.

A Celery system can consist of multiple workers and brokers, giving way to high availability and horizontal scaling.

Celery is a library written in Python, but the protocol can be implemented in any language. So far there's RCelery for the Ruby programming language, and a PHP client, but language interoperability can also be achieved by using webhooks.

What do I need?

Celery version 3.0 runs on,

- Python (2.5, 2.6, 2.7, 3.2, 3.3)

- PyPy (1.8, 1.9)
- Jython (2.5, 2.7).

This is the last version to support Python 2.5, and from Celery 3.1, Python 2.6 or later is required. The last version to support Python 2.4 was Celery series 2.2.

Celery is usually used with a message broker to send and receive messages. The RabbitMQ, Redis transports are feature complete, but there's also experimental support for a myriad of other solutions, including using SQLite for local development.

Celery can run on a single machine, on multiple machines, or even across datacenters.

Get Started

If this is the first time you're trying to use Celery, or you are new to Celery 3.0 coming from previous versions then you should read our getting started tutorials:

- First steps with Celery

  > Tutorial teaching you the bare minimum needed to get started with Celery.

- Next steps

> A more complete overview, showing more features.

Celery is...

- **Simple**

  > Celery is easy to use and maintain, and does *not need configuration files*.
  >
  > It has an active, friendly community you can talk to for support, including a mailing-list and and an IRC channel.
  >
  > Here's one of the simplest applications you can make:

```
from celery import Cel

app = Celery('hello',

@app.task
def hello():
    return 'hello worl
```

- **Highly Available**

  > Workers and clients will automatically retry in the event of connection loss or failure, and some brokers support HA in way of *Master/Master* or *Master/Slave* replication.

- **Fast**

  > A single Celery process can process millions of tasks a minute, with sub-millisecond round-trip latency (using RabbitMQ, py-librabbitmq, and optimized settings).

- **Flexible**

  > Almost every part of *Celery* can be extended or used on its own, Custom pool implementations, serializers, compression schemes, logging, schedulers, consumers, producers, autoscalers, broker transports and much more.

It supports...

- **Message Transports**

  - RabbitMQ, Redis,
  - MongoDB (experimental), Amazon SQS (experimental),
  - CouchDB (experimental), SQLAlchemy (experimental),

    - Django ORM (experimental), IronMQ
    - and more...

- **Concurrency**

    - Prefork, Eventlet, gevent, threads/single threaded

- **Result Stores**

    - AMQP, Redis
    - memcached, MongoDB
    - SQLAlchemy, Django ORM
    - Apache Cassandra, IronCache

- **Serialization**

    - *pickle*, *json*, *yaml*, *msgpack*.
    - *zlib*, *bzip2* compression.
    - Cryptographic message signing.

Framework Integration

> Celery is easy to integrate with web frameworks, some of which even have integration packages:
>
> | | |
> |---|---|
> | Django | not needed |
> | Pyramid | pyramid_celery |
> | Pylons | celery-pylons |
> | Flask | not needed |
> | web2py | web2py-celery |
> | Tornado | tornado-celery |
>
> The integration packages are not strictly necessary, but they can make development easier, and sometimes they add important hooks like closing database connections at `fork`.

Documentation

The latest documentation with user guides, tutorials and API reference is hosted at Read The Docs.

Installation

You can install Celery either via the Python Package Index (PyPI) or from source.

To install using pip,:

```
$ pip install -U Celery
```

To install using easy_install,:

```
$ easy_install -U Celery
```

Bundles

Celery also defines a group of bundles that can be used to install Celery and the dependencies for a given feature.

You can specify these in your requirements or on the `pip` comand-line by using brackets. Multiple bundles can be specified by separating them by commas.

```
$ pip install celery[librabbit
```

```
$ pip install celery[librabbit
```

The following bundles are available:

Serializers

celery[auth]:	for using the auth serializer.
celery[msgpack]:	for using the msgpack serializer.
celery[yaml]:	for using the yaml

	serializer.

Concurrency

celery[eventlet]:	for using the eventlet pool.
celery[gevent]:	for using the gevent pool.
celery[threads]:	for using the thread pool.

Transports and Backends

celery[librabbitmq]:	for using th librabbitmc library.
celery[redis]:	for using R as a messa transport c a result backend.
celery[mongodb]:	for using MongoDB message transport (*experimer* or as a resu backend (*supportec*

celery[sqs]:	for using Amazon SQ as a messa transport (*experimer*
celery[memcache]:	for using memcache as a result backend.
celery[cassandra]:	for using Apache Cassandra a result backend.
celery[couchdb]:	for using CouchDB a message transport (*experimer*
celery[couchbase]:	for using CouchBase a result backend.
celery[beanstalk]:	for using Beanstalk a message transport (*experimer*
celery[zookeeper]:	for using Zookeeper a message transport.

celery[zeromq]:	for using ZeroMQ as message transport (*experimer*
celery[sqlalchemy]:	for using SQLAlchen as a messa transport (*experimer* or as a resu backend (*supportec*
celery[pyro]:	for using th Pyro4 message transport (*experimer*
celery[slmq]:	for using th SoftLayer Message Queue transport (*experimer*

Downloading and installing from source

Download the latest version of Celery from http://pypi.python.org/pypi/celery/

You can install it by doing the following,:

```
$ tar xvfz celery-0.0.0.tar.gz
$ cd celery-0.0.0
$ python setup.py build
# python setup.py install
```

The last command must be executed as a privileged user if you are not currently using a virtualenv.

Using the development version

With pip

The Celery development version also requires the development versions of `kombu`, `amqp` and `billiard`.

You can install the latest snapshot of these using the following pip commands:

```
$ pip install https://github.c
$ pip install https://github.c
$ pip install https://github.c
$ pip install https://github.c
```

With git

Please the Contributing section.

Getting Help

Mailing list

For discussions about the usage, development, and future of celery, please join the celery-users mailing list.

IRC

Come chat with us on IRC. The **#celery** channel is located at the Freenode network.

Bug tracker

If you have any suggestions, bug reports or annoyances please report them to our issue tracker at http://github.com/celery/celery/issues/

Wiki

http://wiki.github.com/celery/celery/

Contributing

Development of celery happens at Github: http://github.com/celery/celery

You are highly encouraged to participate in the development of celery. If you don't like Github (for some reason) you're welcome to send regular patches.

Be sure to also read the Contributing to Celery section in the documentation.

License

This software is licensed under the New BSD License. See the `LICENSE` file in the top distribution directory for the full license text.