FREE eBook

LEARNING celery

Free unaffiliated eBook created from
Stack Overflow contributors.

#celery

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: celery

It is an unofficial and free celery ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official celery.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with celery

Remarks

"Celery is an asynchronous task queue/job queue based on distributed message passing." – http://www.celeryproject.org/

Celery is great for asychronous and scheduled background tasks. It is commonly used for long-running tasks that are part of a Django or Flask application.

Examples

Installation or Setup

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

To install the latest version using `pip`:

```
$ pip install celery
```

To install using `easy_install`:

```
$ easy_install celery
```

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 # as root
```

Celery + Redis

Installation

Additional dependencies are required for Redis support. Install both Celery and the dependencies in one go using the `celery[redis]` bundle:

```
$ pip install -U celery[redis]
```

Configuration

Configure the location of your Redis database:

```
BROKER_URL = 'redis://localhost:6379/0'
```

The URL should be in the format of:

```
redis://:password@hostname:port/db_number
```

Application

Create the file tasks.py:

```
from celery import Celery

BROKER_URL = 'redis://localhost:6379/0'
app = Celery('tasks', broker=BROKER_URL)

@app.task
def add(x, y):
    return x + y
```

The first argument to `Celery` is the name of the current module. This way names can be automatically generated. The second argument is the `broker` keyword which specifies the URL of the message broker.

Running the celery worker server

Run the worker by executing with the worker argument:

```
$ celery -A tasks worker --loglevel=info
```

Calling the task

To call the task, use the `delay()` method.

```
>>> from tasks import add
>>> add.delay(4, 4)
```

Calling a task returns an AsyncResult instance, which can check the state of the task, wait for the task to finish, or get its return value. (If the task failed, it gets the exception and traceback).

Keeping Results

To keep track of the task's states, Celery needs to store or send the states somewhere. Use Redis as the result backend:

```
BROKER_URL = 'redis://localhost:6379/0'
BACKEND_URL = 'redis://localhost:6379/1'
app = Celery('tasks', broker=BROKER_URL, backend=BACKEND_URL)
```

To read more about result backends please see Result Backends.

Now with the result backend configured, call the task again. This time hold on to the AsyncResult instance returned from the task:

```
>>> result = add.delay(4, 4)
```

The `ready()` method returns whether the task has finished processing or not:

```
>>> result.ready()
False
```

It is possible to wait for the result to complete, but this is rarely used since it turns the asynchronous call into a synchronous one:

```
>>> result.get(timeout=1)
8
```

Based on celery official document

Read Getting started with celery online: https://riptutorial.com/celery/topic/6987/getting-started-with-celery

Chapter 2: Celery monitoring with Flower

Examples

Up and running with Flower

Flower is a web based tool to monitor Celery.

To install Flower, we can use `pip` as follows:

```
pip install flower
```

To run Flower for `proj`:

```
celery –A proj flower
```

Now, flower is accessible in:

```
http://localhost:5555
```

Read Celery monitoring with Flower online: https://riptutorial.com/celery/topic/7477/celery-monitoring-with-flower

Credits

S. No	Chapters	Contributors
1	Getting started with celery	4444, Community, jamjar, jegesh, Majid
2	Celery monitoring with Flower	ettanany