Kostenloses eBook

LERNEN

celery

Free unaffiliated eBook created from
Stack Overflow contributors.

#celery

Inhaltsverzeichnis

Über

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

Kapitel 1: Erste Schritte mit Sellerie

Bemerkungen

"Celery ist eine Warteschlange für asynchrone Aufgaben, die auf der Weitergabe von Nachrichten basiert." - http://www.celeryproject.org/

Sellerie eignet sich hervorragend für asynchrone und geplante Hintergrundaufgaben. Es wird normalerweise für lang andauernde Aufgaben verwendet, die Teil einer Django- oder Flask-Anwendung sind.

Examples

Installation oder Setup

Sie können Celery entweder über den Python Package Index (PyPI) oder über die Quelle installieren.

So installieren Sie die neueste Version mit `pip` :

```
$ pip install celery
```

So installieren Sie mit `easy_install` :

```
$ easy_install celery
```

Herunterladen und Installieren von der Quelle

Laden Sie die neueste Version von Celery von http://pypi.python.org/pypi/celery/ herunter.

Sie können es folgendermaßen installieren:

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

Sellerie + Redis

Installation

Für die Unterstützung von Redis sind zusätzliche Abhängigkeiten erforderlich. Installieren Sie sowohl Sellerie als auch die Abhängigkeiten mit dem `celery[redis]` :

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

Aufbau

Konfigurieren Sie den Speicherort Ihrer Redis-Datenbank:

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

Die URL sollte folgendes Format haben:

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

Anwendung

Erstellen Sie die Datei task.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
```

Das erste Argument für `Celery` ist der Name des aktuellen Moduls. Auf diese Weise können Namen automatisch generiert werden. Das zweite Argument ist das `broker` Schlüsselwort, das die URL des Nachrichtenbrokers angibt.

Den Sellerieserver-Server ausführen

Führen Sie den Worker aus, indem Sie ihn mit dem Worker-Argument ausführen:

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

Aufruf der Aufgabe

Verwenden Sie zum Aufrufen der Aufgabe die `delay()` -Methode.

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

Beim Aufrufen einer Aufgabe wird eine AsyncResult- Instanz zurückgegeben, die den Status der

Aufgabe prüfen, warten kann, bis die Aufgabe abgeschlossen ist, oder ihren Rückgabewert abrufen. (Wenn der Task fehlgeschlagen ist, erhält er die Ausnahme und das Traceback).

Ergebnisse halten

Um den Status der Aufgabe zu verfolgen, muss Celery die Zustände irgendwo speichern oder senden. Verwenden Sie Redis als Ergebnis-Backend:

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

Weitere Informationen zu den Ergebnis-Backends finden Sie unter Ergebnis-Backends .

Rufen Sie die Task jetzt mit dem konfigurierten Ergebnis-Backend erneut auf. Halten Sie sich diesmal an die von der Task zurückgegebene AsyncResult- Instanz:

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

Die `ready()` -Methode gibt zurück, ob die Task die Verarbeitung abgeschlossen hat oder nicht:

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

Sie können abwarten, bis das Ergebnis abgeschlossen ist. Dies wird jedoch selten verwendet, da der asynchrone Aufruf zu einem synchronen Aufruf wird:

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

Basierend auf offiziellem Sellerie-Dokument

Erste Schritte mit Sellerie online lesen: https://riptutorial.com/de/celery/topic/6987/erste-schritte-mit-sellerie

Kapitel 2: Sellerieüberwachung mit Blume

Examples

Mit Flower loslegen

Flower ist ein webbasiertes Tool zur Überwachung von Sellerie.

Um Flower zu installieren, können wir `pip` wie folgt verwenden:

```
pip install flower
```

Um Flower für `proj` :

```
celery –A proj flower
```

Nun ist die Blume zugänglich in:

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

Sellerieüberwachung mit Blume online lesen:
https://riptutorial.com/de/celery/topic/7477/sellerieuberwachung-mit-blume

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit Sellerie	4444, Community, jamjar, jegesh, Majid
2	Sellerieüberwachung mit Blume	ettanany