Бесплатная электронная книга

учусь

celery

Free unaffiliated eBook created from
Stack Overflow contributors.

#celery

Около

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

глава 1: Начало работы с сельдереем

замечания

«Сельдерей - это асинхронная очередь задач / заданий на основе распределенной передачи сообщений». - http://www.celeryproject.org/

Сельдерей отлично подходит для асинхронных и запланированных фоновых задач. Он обычно используется для длительных задач, которые являются частью приложения Django или Flask.

Examples

Установка или настройка

Вы можете установить Celery либо через Python Package Index (PyPI), либо из источника.

Чтобы установить последнюю версию с помощью `pip` :

```
$ pip install celery
```

Для установки с помощью `easy_install` :

```
$ easy_install celery
```

Загрузка и установка из источника

Загрузите последнюю версию Celery от http://pypi.python.org/pypi/celery/

Вы можете установить его, выполнив следующие действия:

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

Сельдерей + Редис

Монтаж

Для поддержки Redis требуются дополнительные зависимости. Установите как сельдерей, так и зависимости за один раз, используя комплект `celery[redis]` :

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

конфигурация

Настройте расположение базы данных Redis:

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

URL-адрес должен быть в формате:

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

заявка

Создайте файл 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
```

Первый аргумент для `Celery` - это имя текущего модуля. Таким образом, имена могут быть автоматически сгенерированы. Второй аргумент - ключевое слово `broker` которое указывает URL-адрес брокера сообщений.

Запуск сервера работника сельдерея

Запустите работника, выполнив его с помощью аргумента worker:

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

Вызов задачи

Чтобы вызвать задачу, используйте метод `delay()` .

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

Вызов задачи возвращает экземпляр AsyncResult , который может проверить состояние задачи, дождаться завершения задачи или получить ее возвращаемое значение. (Если задача не удалась, она получает исключение и трассировку).

Ведение результатов

Чтобы отслеживать состояния задачи, сельдерей должен где-то хранить или отправлять состояния. Используйте Redis в качестве результата:

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

Подробнее о бэкэндах результатов см. В разделе « Результаты» .

Теперь с настройкой бэкэнда результата снова вызовите задачу. На этот раз удерживайте экземпляр AsyncResult, возвращенный из задачи:

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

Метод `ready()` возвращает завершение обработки задачи или нет:

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

Можно дождаться завершения результата, но это редко используется, поскольку он превращает асинхронный вызов в синхронный:

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

На основе официального документа сельдерея

Прочитайте Начало работы с сельдереем онлайн: https://riptutorial.com/ru/celery/topic/6987/начало-работы-с-сельдереем

глава 2: Контроль сельдерея с цветком

Examples

Вверх и бег с цветами

Цветок - это веб-инструмент для мониторинга сельдерея.

Чтобы установить Flower, мы можем использовать `pip` следующим образом:

```
pip install flower
```

Чтобы запустить Flower for `proj` :

```
celery -A proj flower
```

Теперь цветок доступен в:

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

Прочитайте Контроль сельдерея с цветком онлайн: https://riptutorial.com/ru/celery/topic/7477/контроль-сельдерея-с-цветком

кредиты

S. No	Главы	Contributors
1	Начало работы с сельдереем	4444, Community, jamjar, jegesh, Majid
2	Контроль сельдерея с цветком	ettanany