

Celery文档 中文简译

夏永锋

October 18, 2013

Contents

1 入门指南	2
1.1 Celery简介	2
1.1.1 什么是任务队列?	2
1.1.2 我依赖于什么?	2
1.1.3 开始	2
1.1.4 Celery是...	2
1.1.5 特性	3
1.1.6 框架集成	3
1.1.7 快速跳转	3
1.1.8 安装	3
1.2 中间人 (Broker)	3
1.2.1 代理人使用说明	3
1.2.2 代理人概观	3
1.3 Celery学习第一步	3
1.3.1 选择一个代理人	3
1.3.2 安装Celery	3
1.3.3 应用	3
1.3.4 运行Celery工作者服务器	3
1.3.5 调用任务	3
1.3.6 保存结果	3
1.3.7 配置	3
1.3.8 何去何从	3
1.4 接下来	3
1.5 资源	3
2 用户手册	3
2.1 应用	3
2.2 任务	3
2.3 调用任务	3
2.4 规划: 设计工作流	3
2.5 工作者指南	3
2.6 定期任务	3
2.7 HTTP回调任务 (Web钩子)	3
2.8 路由任务	3
2.9 监控和管理指南	3
2.10 安全	3
2.11 优化	3
2.12 并发	3
2.13 信号	3
3 配置与默认值	3
4 集成到Django	3
5 内部原理	3

1 入门指南

1.1 Celery简介

Celery是一个简单、灵活且可靠的分布式系统，用于处理数量巨大的消息，同时提供维护这样一个系统所需要的工具。Celery是一个消息队列，侧重于实时处理，但也支持任务调度。

1.1.1 什么是任务队列？

任务队列是一种在多线程或多机器之间分配任务的机制。任务队列的输入是一个工作单元，也称为一个任务，然后专用的工作者进程持续地监听该队列，看是否有新的工作需要完成。Celery使用一个中间人在客户端（消息生产者）和工作者（消息消费者）之间传递消息。客户端在对列中放入一个消息来开始一个任务，中间人将该消息分发到一个工作者。

一个Celery系统可以包含多个工作者和中间人，以实现高可用和水平扩展。

Celery以Python语言编写，但协议可以使用任何语言来实现。

1.1.2 我依赖于什么？

Celery的使用需要有一个消息中间人来发送和接收消息。RabbitMQ、Redis和MongoDB作为中间人，特性比较完善，但Celery也支持大量其他方案，包括使用SQLite进行本地开发。

Celery可以运行在单机上，也可以运行在多台机器上，甚至横跨多个数据中心。

1.1.3 开始

如果这是你第一次尝试使用Celery，或你使用过以前的版本，但对于Celery3.0是新手，那么你应该阅读一下我们的入门教程：

- [Celery学习第一步](#)
- [接下来](#)

1.1.4 Celery是...

“

- 简单
Celery易于使用和维护，也不需要配置文件。
- 高度可用
- 快速
- 灵活

”

- 1.1.5 特性
- 1.1.6 框架集成
- 1.1.7 快速跳转
- 1.1.8 安装
- 1.2 中间人 (Broker)
- 1.2.1 代理人使用说明
- 1.2.2 代理人概观
- 1.3 Celery学习第一步
- 1.3.1 选择一个代理人
- 1.3.2 安装Celery
- 1.3.3 应用
- 1.3.4 运行Celery工作者服务器
- 1.3.5 调用任务
- 1.3.6 保存结果
- 1.3.7 配置
- 1.3.8 何去何从
- 1.4 接下来
- 1.5 资源
- 2 用户手册
- 2.1 应用
- 2.2 任务
- 2.3 调用任务
- 2.4 规划：设计工作流
- 2.5 工作者指南
- 2.6 定期任务
- 2.7 HTTP回调任务 (Web钩子)
- 2.8 路由任务
- 2.9 监控和管理指南
- 2.10 安全
- 2.11 优化
- 2.12 并发
- 2.13 信号
- 3 配置与默认值
- 4 集成到Django
- 5 内部原理