Celery

1 定义

Celery 是一个简单、灵活且可靠的，处理大量消息的分布式系统

它是一个专注于实时处理的任务队列，同时也支持任务调度

中文官网：http://docs.jinkan.org/docs/celery/

在线安装 sudo pip3 install celery

离线安装

```
tar xvfz celery-0.0.0.tar.gz
cd celery-0.0.0
python3 setup.py build
python3 setup.py install
```

2,使用场景

1，任务调度

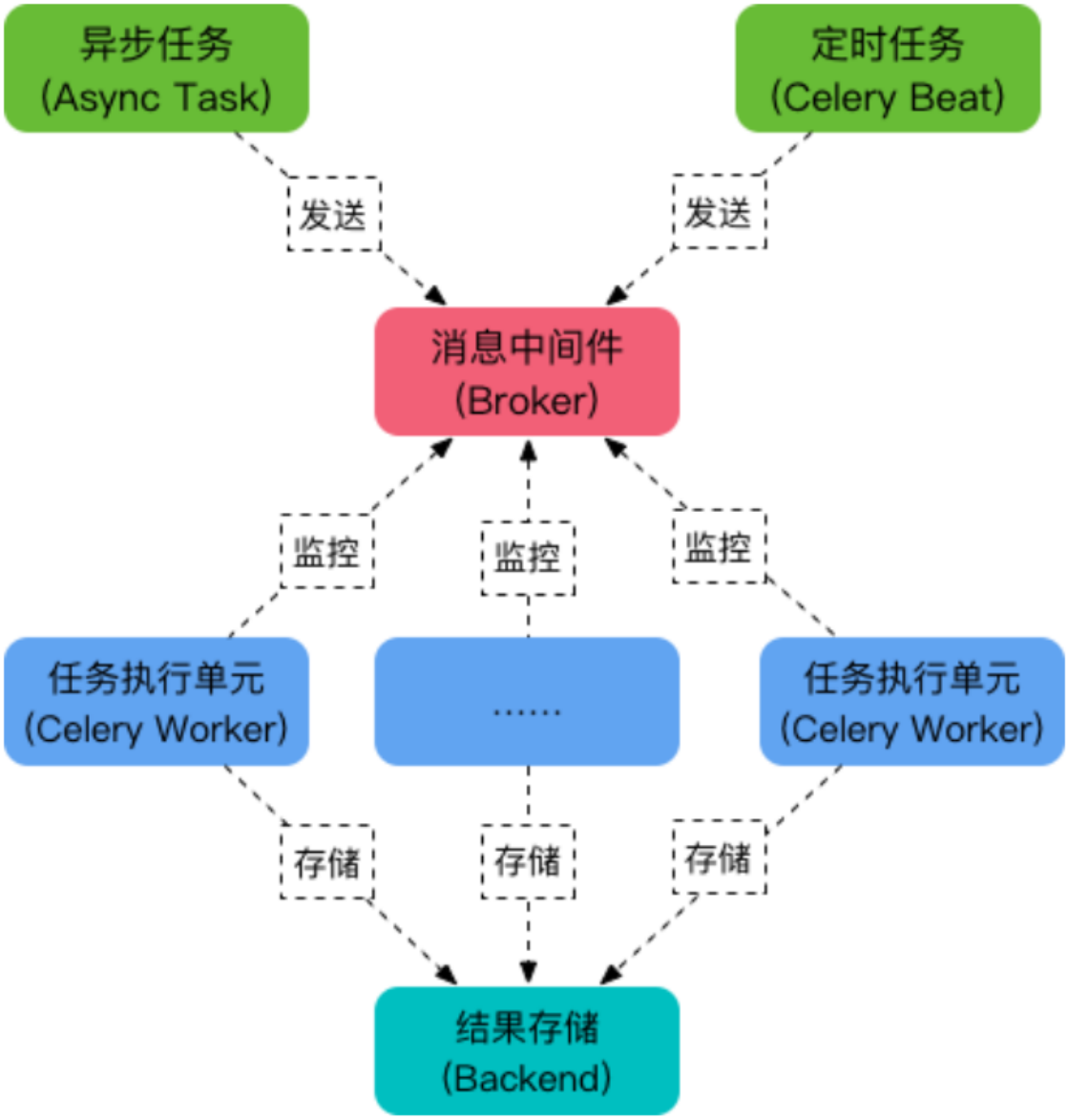
2，避开阻塞操作

名词解释：

broker - 消息传输的中间件，生产者一旦有消息发送，将发至broker；【RQ，redis】

backend - 用于存储消息/任务结果，如果需要跟踪和查询任务状态，则需添加要配置相关

worker - 工作者 - 消费/执行broker中消息/任务的进程



2 使用celery

1, 创建celery应用

```
#创建 tasks.py 文件

from celery import Celery
#初始化celery, 指定broker
app = Celery('guoxiaonao',
broker='redis://:password@127.0.0.1/1')

# 创建任务函数
@app.task
def task_test():
    print("task is running....")

#在该tasks.py同级目录下 执行 celery -A tasks
worker --loglevel=info
```

2,推送任务

在tasks.py文件的同级目录进入 python3 执行 如下代码

```
from tasks import task_test
task_test.delay()
```

3,存储执行结果

Celery提供存储任务执行结果的方案，需借助 redis 或 mysql 或Memcached 等

详情可见 http://docs.celeryproject.org/en/latest/reference/celery.result.html#module-celery.result

```
from celery import Celery


app = Celery('demo',

broker='redis://@127.0.0.1:6379/1'

backend='redis://@127.0.0.1:6379/2',
             )

# 创建任务函数
@app.task
def task_test(a, b):
    print("task is running")
    return a + b
```

3 Django + Celery

1，创建项目+应用

```
#常规命令
django-admin startproject chongci
python manage.py startapp user
```

2，创建celery.py

在settings.py同级目录下 创建 celery.py文件

文件内容如下：

```
from celery import Celery
from django.conf import settings
```

```
import os

# 为celery设置环境变量
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'chongci.settings')

# 创建应用
app = Celery("chongci")
# 配置应用
app.conf.update(
    # 配置broker

 BROKER_URL='redis://:password@127.0.0.1:6379/1',
)
# 设置app自动加载任务
app.autodiscover_tasks(settings.INSTALLED_APPS)
```

3, 在应用模块【user目录下】创建tasks.py文件

文件内容如下:

```
from chongci.celery import app
import time

@app.task
def task_test():
    print("task begin....")
    time.sleep(10)
    print("task over....")
```

4，应用视图编写；内容如下：

```
from django.http import HttpResponse
from .tasks import task_test
import datetime

def test_celery(request):
    task_test.delay()
    now = datetime.datetime.now()
    html = "return at %s"%
(now.strftime('%H:%M:%S'))
    return HttpResponse(html)
```

5，分布式路由下添加 test_celery函数对应路由，此过程略

6，启动django　python3 manage.py runserver

7，创建 celery worker

　在项目路径下，即chongci 下 执行如下

```
celery -A chongci worker -l info
```