### 设计说明

此任务框架支持集群，采用redis控制一个任务只能在一台服务器上执行，集群的主要作用为了出现一台宕机的情况下不影响任务

采用redis订阅和发布进行任务的控制，订阅一个频道的任务线程在收到订阅信息后会执行相应的指令。数据协议 参考下文 redis订阅协议

JobManagePubSub：实现了redis的订阅接口，用来接收发布消息的redis数据

实现 ScheduleJobsOperate 接口可以加载数据库中配置的任务  
AbstractJob：抽象的任务，添加新的任务可以继承这个抽象类

SubscriberCenterImpl：订阅控制中心，用来添加订阅

TaskManager：初始化定时任务，并且控制定时任务

### Redis订阅协议

通过redis频道项订阅此频道的线程发布如下数据：

{  
    "id":"1",  
    "executeParams":"{"executeTime":1492676032822,"params":{}}",  
    "method":"runAJobNow"  
}

Id:配置在数据库的任务id

executeParams：执行参数：

executeTime：传递的任务执行时间

params：需要传递的其他的个性化参数

method：调用的方法

getAllJob：获取所有任务信息（数据会保存在redis中）

{  
    "id":"1",  
    "method":"getAllJob"  
}

addJob：添加一个任务

{  
    "id":"1",  
    "method":"addJob",  
    "oldJobGroup":"testGroup",  
    "oldJobName":"test1ddd"  
}

runAJobNow：立即运行一个任务

{  
    "id":"1",  
    "executeParams":"{"executeTime":1492676032822,"params":{}}",  
    "method":"runAJobNow"  
}

pauseJob：暂停一个任务

{  
    "id":"1",  
    "method":"pauseJob"  
}

resumeJob：恢复一个任务

{  
    "id":"1",  
    "method":"resumeJob"  
}

deleteJob删除一个任务

{  
    "id":"1",  
    "method":"deleteJob"  
}

### 测试例子

**public** **class** RedisPublish {

Jedis jedis=**new** Jedis("127.0.0.1",6379);

**private** String channelName="testManage";

@Test

**public** **void** runAJobNow(){

Map<String,String> params=**new** HashMap<String, String>();

params.put("id", "1");

params.put("method", "runAJobNow");

ExecuteParams executeParams =**new** ExecuteParams();

Calendar instance = Calendar.*getInstance*();

instance.add(Calendar.***MONTH***, 1);

executeParams.setExecuteTime(instance.getTime());

params.put("executeParams", JSON.*toJSONString*(executeParams));

System.***out***.println(JSON.*toJSONString*(params));

jedis.publish(channelName, JSON.*toJSONString*(params));

}

}