可通过配置多个任务监听器，在任务执行前和执行后执行监听的方法。监听器分为每台作业节点均执行和分布式场景中仅单一节点执行2种。

1. 每台作业节点均执行的监听

若作业处理作业服务器的文件，处理完成后删除文件，可考虑使用每个节点均执行清理任务。此类型任务实现简单，且无需考虑全局分布式任务是否完成，请尽量使用此类型监听器。

步骤：

* 定义监听器

public class MyElasticJobListener implements ElasticJobListener {

@Override

public void beforeJobExecuted(ShardingContexts shardingContexts) {

// do something ...

}

@Override

public void afterJobExecuted(ShardingContexts shardingContexts) {

// do something ...

}

}

* 将监听器作为参数传入JobScheduler

public class JobMain {

public static void main(String[] args) {

new JobScheduler(createRegistryCenter(), createJobConfiguration(), new MyElasticJobListener()).init();

}

private static CoordinatorRegistryCenter createRegistryCenter() {

CoordinatorRegistryCenter regCenter = new ZookeeperRegistryCenter(new ZookeeperConfiguration("zk\_host:2181", "elastic-job-demo"));

regCenter.init();

return regCenter;

}

private static LiteJobConfiguration createJobConfiguration() {

// 创建作业配置

...

}

}

2. 分布式场景中仅单一节点执行的监听

若作业处理数据库数据，处理完成后只需一个节点完成数据清理任务即可。此类型任务处理复杂，需同步分布式环境下作业的状态同步，提供了超时设置来避免作业不同步导致的死锁，请谨慎使用。

步骤：

* 定义监听器

public class TestDistributeOnceElasticJobListener extends AbstractDistributeOnceElasticJobListener {

public TestDistributeOnceElasticJobListener(long startTimeoutMills, long completeTimeoutMills) {

super(startTimeoutMills, completeTimeoutMills);

}

@Override

public void doBeforeJobExecutedAtLastStarted(ShardingContexts shardingContexts) {

// do something ...

}

@Override

public void doAfterJobExecutedAtLastCompleted(ShardingContexts shardingContexts) {

// do something ...

}

}

* 将监听器作为参数传入JobScheduler

public class JobMain {

public static void main(String[] args) {

long startTimeoutMills = 5000L;

long completeTimeoutMills = 10000L;

new JobScheduler(createRegistryCenter(), createJobConfiguration(), new MyDistributeOnceElasticJobListener(startTimeoutMills, completeTimeoutMills)).init();

}

private static CoordinatorRegistryCenter createRegistryCenter() {

CoordinatorRegistryCenter regCenter = new ZookeeperRegistryCenter(new ZookeeperConfiguration("zk\_host:2181", "elastic-job-demo"));

regCenter.init();

return regCenter;

}

private static LiteJobConfiguration createJobConfiguration() {

// 创建作业配置

...

}

}