

SpringBatch从入门到实战(十一)：事务控制_springbatch 执行一次提交事务-CSDN博客

blog.csdn.net/vbirdbest/article/details/131808505

SpringBatch从入门到实战(十一)：事务控制

原创

风流少年 于 2023-07-19 15:18:37 发布

阅读量1.1k 收藏 1

点赞数 2

分类专栏：SpringBatch从入门到实战 文章标签：springbatch

版权



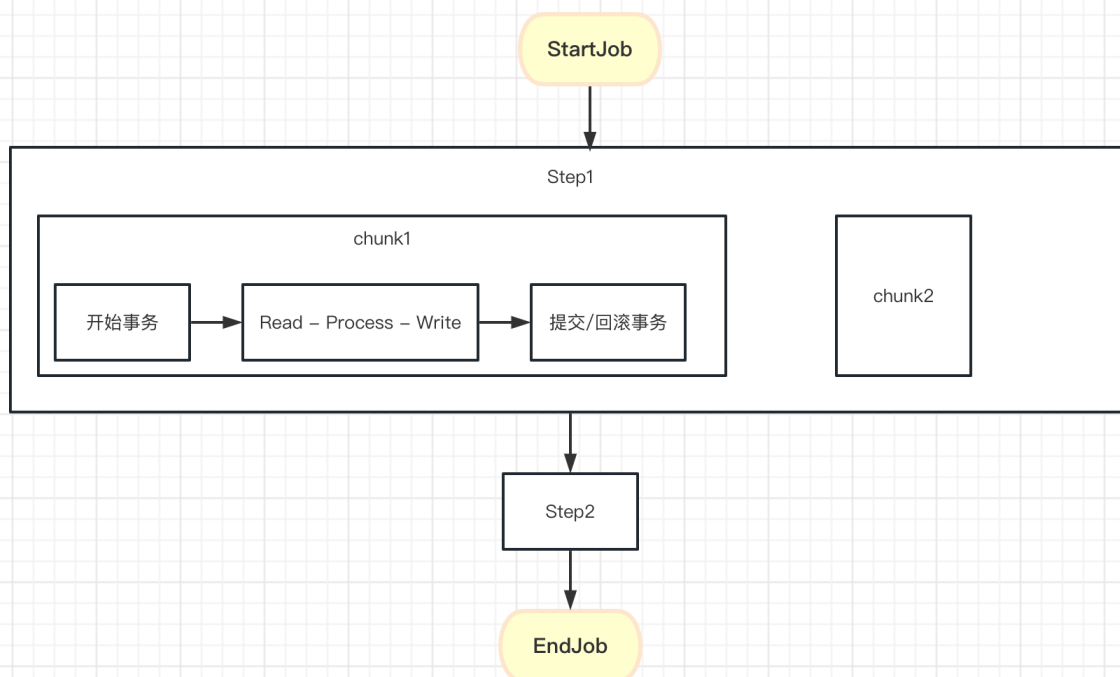
SpringBatch从入门到实战 专栏收录该内容

14 篇文章 19 订阅

订阅专栏

一：事务模型

- Step之间事务独立。
- Step划分成多个Chunk执行，Chunk事务批次独立，互不影响。
- Chunk开始启动一个事务，Chunk结束时提交或者回滚事务。



CSDN @风流少年

二：事务回滚控制

- 默认情况下，无论是设置了重试retry，还是跳过skip，只要从Writer抛出一个异常都会导致事务回滚。如果设置了skip机制，那么在Reader中抛出异常不会导致回滚。
- 有些从Writer抛出一个异常并不需要回滚数据，noRollback属性为Step提供了不必进行事务回滚的异常配置。

```
@Bean
public Step step() {
    return stepBuilderFactory.get("step1")
        .<Integer, Integer>chunk(2)
        .reader(itemReader())
        .processor(itemProcessor())
        .writer(itemWriter())
        .faultTolerant()
        .noRollback(ValidationException.class) // 不必回滚的异常
        .build();
}
```

三：事务数据读取的缓存

一个步骤Step分为Reader、Processor、Writer三个阶段，默认情况下如果错误不是发生在Reader阶段，那么就没有必要去重新读一次数据（框架会缓存起来）。但是某些场景下需要Reader部分也需要重新执行，比如Reader是从一个JMS队列中消费消息，当发生回滚时代表消费失败，重试的时候再从队列上拉取，才能确保JMS的逻辑完整性，这个场景可以使用readerIsTransactionalQueue来配置数据重读。

```
@Bean
public Step step() {
    return stepBuilderFactory.get("step1")
        .<Integer, Integer>chunk(2)
        .reader(itemReader())
        .processor(itemProcessor())
        .writer(itemWriter())
        .readerIsTransactionalQueue()
        .build();
}
```

四：事务属性

事务的属性包括隔离级别()、传播方式 () 以及过期时间(timeout).

```
@Bean
public Step step() {
    DefaultTransactionAttribute attribute = new DefaultTransactionAttribute();
    attribute.setPropagationBehavior(Propagation.REQUIRED.value());
    attribute.setIsolationLevel(Isolation.DEFAULT.value());
    attribute.setTimeout(30);

    return stepBuilderFactory.get("step1")
        .<Integer, Integer>chunk(2)
        .reader(itemReader())
        .processor(itemProcessor())
        .writer(itemWriter())
        .transactionAttribute(attribute)
        .build();
}
```