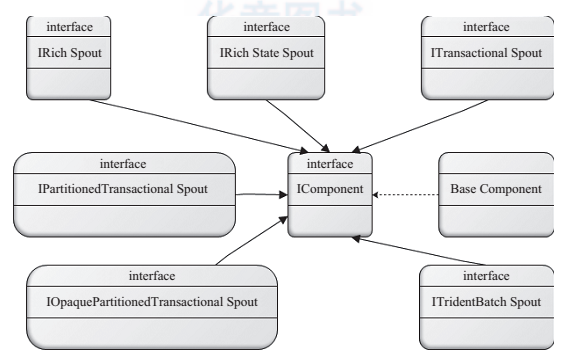
Spout数据源

消息源Spout是Storm的Topology中的消息生产者（即Tuple的创造者）。

Spout介绍

1. Spout的结构

Spout是Storm的核心组件之一，最源头的接口是IComponent，如图所示，几个Spout接口都继承自IComponent。



1. Spout发出的消息

Spout从外部获取数据后，向Topology中发出的Tuple可以是可靠的，也可以是不可靠的。

注意：一个可靠的消息源可以重新发射一个Tuple（如果该Tuple没有被Storm成功处理），但是一个不可靠的消息源Spout一旦发出，一个Tuple就把它彻底“遗忘”，也就不能再发了。

1. Spout发射的流

Spout可以发射多个流。要达到这样的效果，使用OutputFieldsDeclare.declareStream来定义多个流（即定义多个Stream），然后使用SpoutOutputCollector的emit来发射指定的流。

1. Spout的重要方法

Spout的重要方法是nextTuple()，nextTuple方法发射一个新的元组到Topology，如果没有新元组发射，则直接返回。注意任务Spout的nextTuple方法都不要实现成阻塞的，因为Storm要在相同的线程中调用Spout的方法。Spout的另外两个重要方法是ack()和fail()方法。当Spout发射的元组被拓扑成功处理时，调用ack方法；当处理失败时，调用fail方法。ack和fail方法仅被可靠的Spout调用。

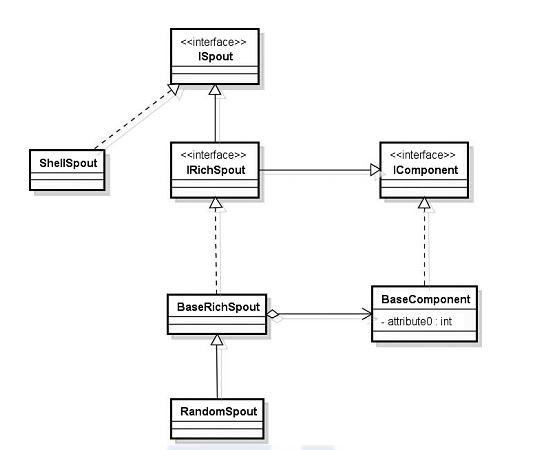
1. Spout的组件

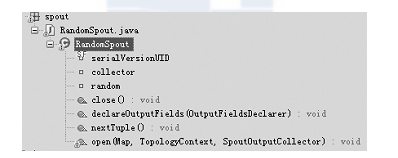
Spout的最顶层抽象是ISpout接口。在通常情况下（Shell和事务型的除外），实现一个Spout，可以直接实现接口IRichSpout，如果不想写多余的代码，可以直接继承BaseRichSpout。

【注】对于Spout，有ISpout、IRichSpout、BaseRichSpout；对于Bolt，有IBoult、IRichBolt、IBasicBolt、BaseBasicBolt，IBasicBolt、BaseBasicBolt不用每次execute完成都写ack/fail，因为已经帮你实现好了。

Spout实例

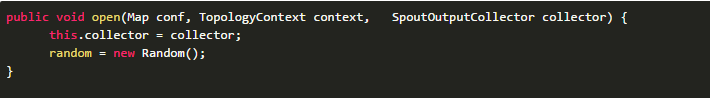
下面通过创建一个实例RandomSpout来介绍Spout，图1为RandomSpout继承自BasicRichSpout及其实现的原理图。图2列出了实例RandomSpout继承自BaseRichSpout中的几个重要方法。





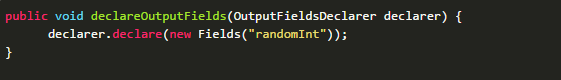
1. open方法

当一个Task被初始化时会调用此open方法。一般都会在此方法中初始化发送Tuple的对象SpoutOutputCollector和配置对象TopologyContext。



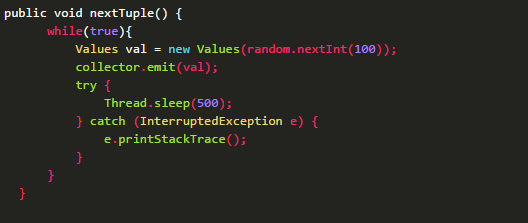
1. declareOutputFields方法

此方法用于声明当前Spout的Tuple发送流。流的定义是通过OutputFieldsDeclare.declareStream方法完成的，其中的参数包括了发送的域Fields。



1. nextTuple方法

这是Spout类中最重要的一个方法。发射一个Tuple到Topology都是通过该方法来实现的。



以上代码从100以内的整数中随机生成一个数作为Tuple的值，然后通过collector发送到Topology。

另外，除了上述几个方法之外，还有getComponentConfiguration、ack、fail和close方法等。getComponentConfiguration方法用于配置当前组件的参数，Storm监测到一个Tuple被成功处理时调用ack方法，处理失败时调用fail方法，这两个方法在BaseRickSpout类中已经被隐式实现了。