如果要从0.10.1.x升级到0.10.2。请参与[0.10.2的升级](http://orchome.com/505)部分。主要强调了升级应用时需要考虑的不兼容性。下面是完整的0.10.2 API和变化列表，提升你的程序或简化代码，并包含了新功能的使用。

**0.10.2.0中Streams API的变化**

KafkaStreams中的新方法：

* 设置一个监听器来响应程序状态的变化（#setStateListener(StateListener listener)）。
* 通过#state检查当前应用程序的状态。
* 通过#metrics()检索全局度量注册
* 通过close（long timeout,TimeUnit timeUnit）关闭时的超时时间
* 通过#toString(String indent)检查Kafka Streams信息时指定自定义的缩进

StreamsConfig中的参数的变化：

* zookeeper.connect已废弃；Kafka Streams应用程序的topic管理不在与Zookeeper相互影响，而是使用新的broker管理协议（参见KIP-4,"主题管理模式"一节）
* 为度量，安全和客户端配置增加了许多新参数

StreamsMetrics接口中的变化

* 移除方法：#addLatencySensor()
* 增加方法: #addLatencyAndThroughputSensor(), #addThroughputSensor(), #recordThroughput(), #addSensor(), #removeSensor()

TopologyBuilder中的新方法：

* 增加了#addSource（）的重载，允许为每个源节点定义一个auto.offset.reset策略
* 增加了#addGlobalStore（）方法，添加到全局StateStores

KStreamBuilder中的新方法：

* 增加了#stream()和#table()的重载。允许为每个输入stream/table定义一个auto.offset.reset策略。  
  added method #globalKTable() to create a GlobalKTable
* 添加方法#globalKTable（）来创建GlobalKTable

KStream的新连接

* 增加了用于和KTable连接的#join()重载。
* 增加了用于和GlobalKTable连接的#join和leftJoin()重载。
* 注意，0.10.2中的连接有所改进，因此你需要与0.10.0.x和0.10.1.x进行对比，可能会看到不同的结果（参见Apache Kafka wiki中的Kafka Streams Join语义）

KTable连接的空键对其处理

* 像其他的KTable操作一样，KTable-KTable连接不再对空键抛出异常，而是静静的删除这些记录。

新增窗口类型会话窗口

* 添加了SessionWindows类来指定会话窗口
* 增加了KGroupedStream方法的重载：#count（），#reduce（）和#aggregate（），以允许会话窗口聚合。

TimestampExtractor的变化：

* #extract()方法增加了第二个参数
* 新的默认时间戳提取器类FailOnInvalidTimestamp（提供了与旧的（和已经移除的）默认提取器ConsumerRecordTimestampExtractor相同的功能）
* 新替代时间戳提取器类LogAndSkipOnInvalidTimestamp和UsePreviousTimeOnInvalidTimestamps.

许多DSL接口、类和方法的松散类型约束（参见.KIO-100）。

**Streams API更改为0.10.1.0**

流分组和聚合分为2个方法：

* 老的: KStream #aggregateByKey(), #reduceByKey(), #countByKey()
* 新的: KStream#groupByKey() plus KGroupedStream #aggregate(), #reduce(), 和 #count()
* 例子: stream.countByKey() 更改为 stream.groupByKey().count()

自动重新分配:

* 在变更密钥操作之后和aggregation/join之前，不需要调用through()
* 例子：stream.selectKey(...).through(...).countByKey() 变更为 stream.selectKey().groupByKey().count()

TopologyBuilder:

* #sourceTopics(String applicationId) and #topicGroups(String applicationId)方法简化为 #sourceTopics()和 #topicGroups()

DSL: 指向状态仓库名的新参数：

* 新的Interactive Queries功能需要为所有源KTables和窗口聚合结果KTables指定一个存储名称（之前的参数“operator/window name”现在是storeName）
* KStreamBuilder#table(String topic) 变更为 #topic(String topic, String storeName)
* KTable#through(String topic) 变更为 #through(String topic, String storeName)
* KGroupedStream #aggregate(), #reduce(), 和 #count() 需要增加额外的参数 "String storeName"  
  例子: stream.countByKey(TimeWindows.of("windowName", 1000)) 变更为 stream.groupByKey().count(TimeWindows.of(1000), "countStoreName")

窗口:

* Windows不再命名: TimeWindows.of("name", 1000) 变更为 TimeWindows.of(1000) (参见DSL:新参数指定状态仓库的名称)
* JoinWindows没有默认的大小: JoinWindows.of("name").within(1000) 变更为 to JoinWindows.of(1000)

作者：半兽人  
链接：http://www.orchome.com/335  
来源：OrcHome  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。