/\*

\* Copyright (c) 2016 Fred Cecilia, Valentin Kasas, Olivier Girardot

\*

\* Permission is hereby granted, free of charge, to any person obtaining a copy of

\* this software and associated documentation files (the "Software"), to deal in

\* the Software without restriction, including without limitation the rights to

\* use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of

\* the Software, and to permit persons to whom the Software is furnished to do so,

\* subject to the following conditions:

\*

\* The above copyright notice and this permission notice shall be included in all

\* copies or substantial portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

\* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS

\* FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR

\* COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER

\* IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN

\* CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*/

//Derived from: https://github.com/aseigneurin/kafka-streams-scala

package com.twitter.unified\_user\_actions.kafka.serde

import com.twitter.finagle.stats.Counter

import com.twitter.finagle.stats.NullStatsReceiver

import com.twitter.finatra.kafka.serde.internal.\_

import com.twitter.unified\_user\_actions.kafka.serde.internal.\_

import com.twitter.scrooge.ThriftStruct

/\*\*

\* NullableScalaSerdes is pretty much the same as com.twitter.finatra.kafka.serde.ScalaSerdes

\* The only difference is that for the deserializer it returns null instead of throwing exceptions.

\* The caller can also provide a counter so that the number of corrupt/bad records can be counted.

\*/

object NullableScalaSerdes {

def Thrift[T <: ThriftStruct: Manifest](

nullCounter: Counter = NullStatsReceiver.NullCounter

): ThriftSerDe[T] = new ThriftSerDe[T](nullCounter = nullCounter)

def CompactThrift[T <: ThriftStruct: Manifest](

nullCounter: Counter = NullStatsReceiver.NullCounter

): CompactThriftSerDe[T] = new CompactThriftSerDe[T](nullCounter = nullCounter)

val Int = IntSerde

val Long = LongSerde

val Double = DoubleSerde

}