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

import com.twitter.finagle.stats.InMemoryStatsReceiver

import com.twitter.inject.Test

import com.twitter.unified\_user\_actions.thriftscala.\_

class NullableScalaSerdesSpec extends Test {

val counter = (new InMemoryStatsReceiver).counter("nullCounts")

val nullableDeserializer = NullableScalaSerdes.Thrift[UnifiedUserActionSpec](counter).deserializer

val serializer = NullableScalaSerdes.Thrift[UnifiedUserActionSpec]().serializer

val uua = UnifiedUserActionSpec(

userId = 1L,

payload = Some("test"),

)

test("serde") {

nullableDeserializer.deserialize("", serializer.serialize("", uua)) should be(uua)

nullableDeserializer.deserialize("", "Whatever".getBytes) should be(

null.asInstanceOf[UnifiedUserActionSpec])

counter.apply() should equal(1)

}

test("rate limited logger when there's an exception") {

for (\_ <- 1 to 10) {

nullableDeserializer.deserialize("", "Whatever".getBytes) should be(

null.asInstanceOf[UnifiedUserActionSpec])

}

TestLogAppender.events.size should (be(1) or be(2))

counter.apply() should equal(11)

}

}