package com.twitter.unified\_user\_actions.service

import com.twitter.clientapp.thriftscala.LogEvent

import com.twitter.finatra.decider.modules.DeciderModule

import com.twitter.finatra.kafka.serde.UnKeyed

import com.twitter.inject.server.TwitterServer

import com.twitter.kafka.client.processor.AtLeastOnceProcessor

import com.twitter.unified\_user\_actions.service.module.KafkaProcessorClientEventModule

object ClientEventServiceMain extends ClientEventService

class ClientEventService extends TwitterServer {

override val modules = Seq(KafkaProcessorClientEventModule, DeciderModule)

override protected def setup(): Unit = {}

override protected def start(): Unit = {

val processor = injector.instance[AtLeastOnceProcessor[UnKeyed, LogEvent]]

closeOnExit(processor)

processor.start()

}

}