package com.twitter.recosinjector.event\_processors

import com.twitter.finagle.mtls.authentication.ServiceIdentifier

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.recosinjector.edges.{EventToMessageBuilder, UserUserEdge}

import com.twitter.recosinjector.publishers.KafkaEventPublisher

import com.twitter.scrooge.ThriftStructCodec

import com.twitter.socialgraph.thriftscala.WriteEvent

import com.twitter.util.Future

/\*\*

\* This processor listens to events from social graphs services. In particular, a major use case is

\* to listen to user-user follow events.

\*/

class SocialWriteEventProcessor(

override val eventBusStreamName: String,

override val thriftStruct: ThriftStructCodec[WriteEvent],

override val serviceIdentifier: ServiceIdentifier,

kafkaEventPublisher: KafkaEventPublisher,

userUserGraphTopic: String,

userUserGraphMessageBuilder: EventToMessageBuilder[WriteEvent, UserUserEdge]

)(

override implicit val statsReceiver: StatsReceiver)

extends EventBusProcessor[WriteEvent] {

override def processEvent(event: WriteEvent): Future[Unit] = {

userUserGraphMessageBuilder.processEvent(event).map { edges =>

edges.foreach { edge =>

kafkaEventPublisher.publish(edge.convertToRecosHoseMessage, userUserGraphTopic)

}

}

}

}