package com.twitter.unified\_user\_actions.adapter.email\_notification\_event

import com.twitter.finagle.stats.NullStatsReceiver

import com.twitter.finagle.stats.StatsReceiver

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

import com.twitter.ibis.thriftscala.NotificationScribe

import com.twitter.ibis.thriftscala.NotificationScribeType

import com.twitter.unified\_user\_actions.adapter.AbstractAdapter

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

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

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

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

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

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

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

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

class EmailNotificationEventAdapter

extends AbstractAdapter[NotificationScribe, UnKeyed, UnifiedUserAction] {

import EmailNotificationEventAdapter.\_

override def adaptOneToKeyedMany(

input: NotificationScribe,

statsReceiver: StatsReceiver = NullStatsReceiver

): Seq[(UnKeyed, UnifiedUserAction)] =

adaptEvent(input).map { e => (UnKeyed, e) }

}

object EmailNotificationEventAdapter {

def adaptEvent(scribe: NotificationScribe): Seq[UnifiedUserAction] = {

Option(scribe).flatMap { e =>

e.`type` match {

case NotificationScribeType.Click =>

val tweetIdOpt = e.logBase.flatMap(EmailNotificationEventUtils.extractTweetId)

(tweetIdOpt, e.impressionId) match {

case (Some(tweetId), Some(impressionId)) =>

Some(

UnifiedUserAction(

userIdentifier = UserIdentifier(userId = e.userId),

item = Item.TweetInfo(TweetInfo(actionTweetId = tweetId)),

actionType = ActionType.ClientTweetEmailClick,

eventMetadata = EmailNotificationEventUtils.extractEventMetaData(e),

productSurface = Some(ProductSurface.EmailNotification),

productSurfaceInfo = Some(

ProductSurfaceInfo.EmailNotificationInfo(

EmailNotificationInfo(notificationId = impressionId)))

)

)

case \_ => None

}

case \_ => None

}

}.toSeq

}

}