package com.twitter.follow\_recommendations.common.models

import com.twitter.finagle.tracing.Trace

import com.twitter.follow\_recommendations.logging.{thriftscala => offline}

import com.twitter.follow\_recommendations.{thriftscala => t}

import com.twitter.scrooge.BinaryThriftStructSerializer

import com.twitter.suggests.controller\_data.thriftscala.ControllerData

import com.twitter.util.Base64StringEncoder

/\*\*

\* used for attribution per target-candidate pair

\* @param sessionId trace-id of the finagle request

\* @param controllerData 64-bit encoded binary attributes of our recommendation

\* @param algorithmId id for identifying a candidate source. maintained for backwards compatibility

\*/

case class TrackingToken(

sessionId: Long,

displayLocation: Option[DisplayLocation],

controllerData: Option[ControllerData],

algorithmId: Option[Int]) {

def toThrift: t.TrackingToken = {

Trace.id.traceId.toLong

t.TrackingToken(

sessionId = sessionId,

displayLocation = displayLocation.map(\_.toThrift),

controllerData = controllerData,

algoId = algorithmId

)

}

def toOfflineThrift: offline.TrackingToken = {

offline.TrackingToken(

sessionId = sessionId,

displayLocation = displayLocation.map(\_.toOfflineThrift),

controllerData = controllerData,

algoId = algorithmId

)

}

}

object TrackingToken {

val binaryThriftSerializer = BinaryThriftStructSerializer[t.TrackingToken](t.TrackingToken)

def serialize(trackingToken: TrackingToken): String = {

Base64StringEncoder.encode(binaryThriftSerializer.toBytes(trackingToken.toThrift))

}

def deserialize(trackingTokenStr: String): TrackingToken = {

fromThrift(binaryThriftSerializer.fromBytes(Base64StringEncoder.decode(trackingTokenStr)))

}

def fromThrift(token: t.TrackingToken): TrackingToken = {

TrackingToken(

sessionId = token.sessionId,

displayLocation = token.displayLocation.map(DisplayLocation.fromThrift),

controllerData = token.controllerData,

algorithmId = token.algoId

)

}

}

trait HasTrackingToken {

def trackingToken: Option[TrackingToken]

}