package com.twitter.timelineranker.model

import com.twitter.search.earlybird.thriftscala.\_

import com.twitter.timelineranker.{thriftscala => thrift}

import com.twitter.timelines.model.TweetId

import com.twitter.timelines.model.UserId

object Tweet {

def fromThrift(tweet: thrift.Tweet): Tweet = {

Tweet(id = tweet.id)

}

}

case class Tweet(

id: TweetId,

userId: Option[UserId] = None,

sourceTweetId: Option[TweetId] = None,

sourceUserId: Option[UserId] = None)

extends TimelineEntry {

throwIfInvalid()

def throwIfInvalid(): Unit = {}

def toThrift: thrift.Tweet = {

thrift.Tweet(

id = id,

userId = userId,

sourceTweetId = sourceTweetId,

sourceUserId = sourceUserId)

}

def toTimelineEntryThrift: thrift.TimelineEntry = {

thrift.TimelineEntry.Tweet(toThrift)

}

def toThriftSearchResult: ThriftSearchResult = {

val metadata = ThriftSearchResultMetadata(

resultType = ThriftSearchResultType.Recency,

fromUserId = userId match {

case Some(id) => id

case None => 0L

},

isRetweet =

if (sourceUserId.isDefined || sourceUserId.isDefined) Some(true)

else

None,

sharedStatusId = sourceTweetId match {

case Some(id) => id

case None => 0L

},

referencedTweetAuthorId = sourceUserId match {

case Some(id) => id

case None => 0L

}

)

ThriftSearchResult(

id = id,

metadata = Some(metadata)

)

}

}