package com.twitter.timelineranker.model

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

import com.twitter.timelines.model.UserId

import com.twitter.timelineservice.model.TimelineId

import com.twitter.timelineservice.model.core.TimelineKind

object Timeline {

def empty(id: TimelineId): Timeline = {

Timeline(id, Nil)

}

def fromThrift(timeline: thrift.Timeline): Timeline = {

Timeline(

id = TimelineId.fromThrift(timeline.id),

entries = timeline.entries.map(TimelineEntryEnvelope.fromThrift)

)

}

def throwIfIdInvalid(id: TimelineId): Unit = {

// Note: if we support timelines other than TimelineKind.home, we need to update

// the implementation of userId method here and in TimelineQuery class.

require(id.kind == TimelineKind.home, s"Expected TimelineKind.home, found: ${id.kind}")

}

}

case class Timeline(id: TimelineId, entries: Seq[TimelineEntryEnvelope]) {

throwIfInvalid()

def userId: UserId = {

id.id

}

def throwIfInvalid(): Unit = {

Timeline.throwIfIdInvalid(id)

entries.foreach(\_.throwIfInvalid())

}

def toThrift: thrift.Timeline = {

thrift.Timeline(

id = id.toThrift,

entries = entries.map(\_.toThrift)

)

}

}