package com.twitter.home\_mixer.candidate\_pipeline

import com.twitter.home\_mixer.model.HomeFeatures.AuthorIdFeature

import com.twitter.home\_mixer.model.HomeFeatures.InReplyToTweetIdFeature

import com.twitter.home\_mixer.model.HomeFeatures.IsRetweetFeature

import com.twitter.home\_mixer.model.HomeFeatures.SourceTweetIdFeature

import com.twitter.home\_mixer.model.HomeFeatures.SourceUserIdFeature

import com.twitter.product\_mixer.core.feature.Feature

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMap

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMapBuilder

import com.twitter.product\_mixer.core.functional\_component.transformer.CandidateFeatureTransformer

import com.twitter.product\_mixer.core.model.common.identifier.TransformerIdentifier

import com.twitter.timelineservice.{thriftscala => t}

object TimelineServiceResponseFeatureTransformer extends CandidateFeatureTransformer[t.Tweet] {

override val identifier: TransformerIdentifier = TransformerIdentifier("TimelineServiceResponse")

override val features: Set[Feature[\_, \_]] = Set(

AuthorIdFeature,

InReplyToTweetIdFeature,

IsRetweetFeature,

SourceTweetIdFeature,

SourceUserIdFeature,

)

override def transform(candidate: t.Tweet): FeatureMap = FeatureMapBuilder()

.add(AuthorIdFeature, candidate.userId)

.add(InReplyToTweetIdFeature, candidate.inReplyToStatusId)

.add(IsRetweetFeature, candidate.sourceStatusId.isDefined)

.add(SourceTweetIdFeature, candidate.sourceStatusId)

.add(SourceUserIdFeature, candidate.sourceUserId)

.build()

}