package com.twitter.timelineranker.core

import com.twitter.recos.user\_tweet\_entity\_graph.thriftscala.TweetRecommendation

import com.twitter.search.earlybird.thriftscala.ThriftSearchResult

import com.twitter.timelineranker.model.RecapQuery

import com.twitter.timelines.model.TweetId

object CandidateEnvelope {

val EmptySearchResults: Seq[ThriftSearchResult] = Seq.empty[ThriftSearchResult]

val EmptyHydratedTweets: HydratedTweets = HydratedTweets(Seq.empty, Seq.empty)

val EmptyUtegResults: Map[TweetId, TweetRecommendation] = Map.empty[TweetId, TweetRecommendation]

}

case class CandidateEnvelope(

query: RecapQuery,

searchResults: Seq[ThriftSearchResult] = CandidateEnvelope.EmptySearchResults,

utegResults: Map[TweetId, TweetRecommendation] = CandidateEnvelope.EmptyUtegResults,

hydratedTweets: HydratedTweets = CandidateEnvelope.EmptyHydratedTweets,

followGraphData: FollowGraphDataFuture = FollowGraphDataFuture.EmptyFollowGraphDataFuture,

// The source tweets are

// - the retweeted tweet, for retweets

// - the inReplyTo tweet, for extended replies

sourceSearchResults: Seq[ThriftSearchResult] = CandidateEnvelope.EmptySearchResults,

sourceHydratedTweets: HydratedTweets = CandidateEnvelope.EmptyHydratedTweets)