package com.twitter.cr\_mixer.util

import com.twitter.cr\_mixer.model.CandidateGenerationInfo

import com.twitter.cr\_mixer.model.SourceInfo

import com.twitter.cr\_mixer.thriftscala.CandidateGenerationKey

import com.twitter.cr\_mixer.thriftscala.SimilarityEngine

import com.twitter.cr\_mixer.thriftscala.SourceType

import com.twitter.simclusters\_v2.common.UserId

import com.twitter.simclusters\_v2.thriftscala.InternalId

import com.twitter.util.Time

object CandidateGenerationKeyUtil {

private val PlaceholderUserId = 0L // this default value will not be used

private val DefaultSourceInfo: SourceInfo = SourceInfo(

sourceType = SourceType.RequestUserId,

sourceEventTime = None,

internalId = InternalId.UserId(PlaceholderUserId)

)

def toThrift(

candidateGenerationInfo: CandidateGenerationInfo,

requestUserId: UserId

): CandidateGenerationKey = {

CandidateGenerationKey(

sourceType = candidateGenerationInfo.sourceInfoOpt.getOrElse(DefaultSourceInfo).sourceType,

sourceEventTime = candidateGenerationInfo.sourceInfoOpt

.getOrElse(DefaultSourceInfo).sourceEventTime.getOrElse(Time.fromMilliseconds(0L)).inMillis,

id = candidateGenerationInfo.sourceInfoOpt

.map(\_.internalId).getOrElse(InternalId.UserId(requestUserId)),

modelId = candidateGenerationInfo.similarityEngineInfo.modelId.getOrElse(""),

similarityEngineType =

Some(candidateGenerationInfo.similarityEngineInfo.similarityEngineType),

contributingSimilarityEngine =

Some(candidateGenerationInfo.contributingSimilarityEngines.map(se =>

SimilarityEngine(se.similarityEngineType, se.modelId, se.score)))

)

}

}