package com.twitter.simclusters\_v2.common

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

EmbeddingType,

InternalId,

MultiEmbeddingType,

TopicId,

TopicSubId,

SimClustersEmbeddingId => ThriftEmbeddingId,

SimClustersMultiEmbeddingId => ThriftMultiEmbeddingId

}

/\*\*

\* Helper methods for SimClustersMultiEmbeddingId

\*/

object SimClustersMultiEmbeddingId {

private val MultiEmbeddingTypeToEmbeddingType: Map[MultiEmbeddingType, EmbeddingType] =

Map(

MultiEmbeddingType.LogFavApeBasedMuseTopic -> EmbeddingType.LogFavApeBasedMuseTopic,

MultiEmbeddingType.TwiceUserInterestedIn -> EmbeddingType.TwiceUserInterestedIn,

)

private val EmbeddingTypeToMultiEmbeddingType: Map[EmbeddingType, MultiEmbeddingType] =

MultiEmbeddingTypeToEmbeddingType.map(\_.swap)

def toEmbeddingType(multiEmbeddingType: MultiEmbeddingType): EmbeddingType = {

MultiEmbeddingTypeToEmbeddingType.getOrElse(

multiEmbeddingType,

throw new IllegalArgumentException(s"Invalid type: $multiEmbeddingType"))

}

def toMultiEmbeddingType(embeddingType: EmbeddingType): MultiEmbeddingType = {

EmbeddingTypeToMultiEmbeddingType.getOrElse(

embeddingType,

throw new IllegalArgumentException(s"Invalid type: $embeddingType")

)

}

/\*\*

\* Convert a SimClusters Multi-Embedding Id and SubId to SimClusters Embedding Id.

\*/

def toEmbeddingId(

simClustersMultiEmbeddingId: ThriftMultiEmbeddingId,

subId: Int

): ThriftEmbeddingId = {

val internalId = simClustersMultiEmbeddingId.internalId match {

case InternalId.TopicId(topicId) =>

InternalId.TopicSubId(

TopicSubId(topicId.entityId, topicId.language, topicId.country, subId))

case \_ =>

throw new IllegalArgumentException(

s"Invalid simClusters InternalId ${simClustersMultiEmbeddingId.internalId}")

}

ThriftEmbeddingId(

toEmbeddingType(simClustersMultiEmbeddingId.embeddingType),

simClustersMultiEmbeddingId.modelVersion,

internalId

)

}

/\*\*

\* Fetch a subId from a SimClusters EmbeddingId.

\*/

def toSubId(simClustersEmbeddingId: ThriftEmbeddingId): Int = {

simClustersEmbeddingId.internalId match {

case InternalId.TopicSubId(topicSubId) =>

topicSubId.subId

case \_ =>

throw new IllegalArgumentException(

s"Invalid SimClustersEmbeddingId InternalId type, $simClustersEmbeddingId")

}

}

/\*\*

\* Convert a SimClustersEmbeddingId to SimClustersMultiEmbeddingId.

\* Only support the Multi embedding based EmbeddingTypes.

\*/

def toMultiEmbeddingId(

simClustersEmbeddingId: ThriftEmbeddingId

): ThriftMultiEmbeddingId = {

simClustersEmbeddingId.internalId match {

case InternalId.TopicSubId(topicSubId) =>

ThriftMultiEmbeddingId(

toMultiEmbeddingType(simClustersEmbeddingId.embeddingType),

simClustersEmbeddingId.modelVersion,

InternalId.TopicId(TopicId(topicSubId.entityId, topicSubId.language, topicSubId.country))

)

case \_ =>

throw new IllegalArgumentException(

s"Invalid SimClustersEmbeddingId InternalId type, $simClustersEmbeddingId")

}

}

}