package com.twitter.tsp.stores

import com.twitter.escherbird.topicannotation.strato.thriftscala.TopicAnnotationValue

import com.twitter.escherbird.topicannotation.strato.thriftscala.TopicAnnotationView

import com.twitter.frigate.common.store.strato.StratoFetchableStore

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

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

import com.twitter.storehaus.ReadableStore

import com.twitter.strato.client.Client

import com.twitter.strato.thrift.ScroogeConvImplicits.\_

import com.twitter.util.Future

/\*\*

\* This is copied from `src/scala/com/twitter/topic\_recos/stores/SemanticCoreAnnotationStore.scala`

\* Unfortunately their version assumes (incorrectly) that there is no View which causes warnings.

\* While these warnings may not cause any problems in practice, better safe than sorry.

\*/

object SemanticCoreAnnotationStore {

private val column = "semanticCore/topicannotation/topicAnnotation.Tweet"

def getStratoStore(stratoClient: Client): ReadableStore[TweetId, TopicAnnotationValue] = {

StratoFetchableStore

.withView[TweetId, TopicAnnotationView, TopicAnnotationValue](

stratoClient,

column,

TopicAnnotationView())

}

case class TopicAnnotation(

topicId: TopicId,

ignoreSimClustersFilter: Boolean,

modelVersionId: Long)

}

/\*\*

\* Given a tweet Id, return the list of annotations defined by the TSIG team.

\*/

case class SemanticCoreAnnotationStore(stratoStore: ReadableStore[TweetId, TopicAnnotationValue])

extends ReadableStore[TweetId, Seq[SemanticCoreAnnotationStore.TopicAnnotation]] {

import SemanticCoreAnnotationStore.\_

override def multiGet[K1 <: TweetId](

ks: Set[K1]

): Map[K1, Future[Option[Seq[TopicAnnotation]]]] = {

stratoStore

.multiGet(ks)

.mapValues(\_.map(\_.map { topicAnnotationValue =>

topicAnnotationValue.annotationsPerModel match {

case Some(annotationWithVersions) =>

annotationWithVersions.flatMap { annotations =>

annotations.annotations.map { annotation =>

TopicAnnotation(

annotation.entityId,

annotation.ignoreQualityFilter.getOrElse(false),

annotations.modelVersionId

)

}

}

case \_ =>

Nil

}

}))

}

}