package com.twitter.frigate.pushservice.store

import com.twitter.escherbird.util.uttclient.CachedUttClientV2

import com.twitter.escherbird.util.uttclient.InvalidUttEntityException

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.logging.Logger

import com.twitter.stitch.Stitch

import com.twitter.topiclisting.TopicListingViewerContext

import com.twitter.topiclisting.utt.LocalizedEntity

import com.twitter.topiclisting.utt.LocalizedEntityFactory

import com.twitter.util.Future

/\*\*

\*

\* @param viewerContext: [[TopicListingViewerContext]] for filtering topic

\* @param semanticCoreEntityIds: list of semantic core entities to hydrate

\*/

case class UttEntityHydrationQuery(

viewerContext: TopicListingViewerContext,

semanticCoreEntityIds: Seq[Long])

/\*\*

\*

\* @param cachedUttClientV2

\* @param statsReceiver

\*/

class UttEntityHydrationStore(

cachedUttClientV2: CachedUttClientV2,

statsReceiver: StatsReceiver,

log: Logger) {

private val stats = statsReceiver.scope(this.getClass.getSimpleName)

private val uttEntityNotFound = stats.counter("invalid\_utt\_entity")

private val deviceLanguageMismatch = stats.counter("language\_mismatch")

/\*\*

\* SemanticCore recommends setting language and country code to None to fetch all localized topic

\* names and apply filtering for locales on our end

\*

\* We use [[LocalizedEntityFactory]] from [[Topiclisting]] library to filter out topic name based

\* on user locale

\*

\* Some(LocalizedEntity) - LocalizedUttEntity found

\* None - LocalizedUttEntity not found

\*/

def getLocalizedTopicEntities(

query: UttEntityHydrationQuery

): Future[Seq[Option[LocalizedEntity]]] = Stitch.run {

Stitch.collect {

query.semanticCoreEntityIds.map { semanticCoreEntityId =>

val uttEntity = cachedUttClientV2.cachedGetUttEntity(

language = None,

country = None,

version = None,

entityId = semanticCoreEntityId)

uttEntity

.map { uttEntityMetadata =>

val localizedEntity = LocalizedEntityFactory.getLocalizedEntity(

uttEntityMetadata,

query.viewerContext,

enableInternationalTopics = true,

enableTopicDescription = true)

// update counter

localizedEntity.foreach { entity =>

if (!entity.nameMatchesDeviceLanguage) deviceLanguageMismatch.incr()

}

localizedEntity

}.handle {

case e: InvalidUttEntityException =>

log.error(e.getMessage)

uttEntityNotFound.incr()

None

}

}

}

}

}