package com.twitter.product\_mixer.component\_library.feature\_hydrator.candidate.ads

import com.twitter.adserver.{thriftscala => ad}

import com.twitter.product\_mixer.component\_library.model.candidate.ads.AdsCandidate

import com.twitter.product\_mixer.component\_library.model.query.ads.AdsQuery

import com.twitter.product\_mixer.core.feature.Feature

import com.twitter.product\_mixer.core.feature.FeatureWithDefaultOnFailure

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMap

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMapBuilder

import com.twitter.product\_mixer.core.functional\_component.feature\_hydrator.CandidateFeatureHydrator

import com.twitter.product\_mixer.core.model.common.identifier.FeatureHydratorIdentifier

import com.twitter.product\_mixer.core.pipeline.PipelineQuery

import com.twitter.stitch.Stitch

import com.twitter.storehaus.ReadableStore

import com.twitter.util.Future

import javax.inject.Inject

import javax.inject.Singleton

object AdvertiserBrandSafetySettingsFeature

extends FeatureWithDefaultOnFailure[AdsCandidate, Option[ad.AdvertiserBrandSafetySettings]] {

override val defaultValue = None

}

@Singleton

case class AdvertiserBrandSafetySettingsFeatureHydrator[

Query <: PipelineQuery with AdsQuery,

Candidate <: AdsCandidate] @Inject() (

advertiserBrandSafetySettingsStore: ReadableStore[Long, ad.AdvertiserBrandSafetySettings])

extends CandidateFeatureHydrator[Query, Candidate] {

override val identifier: FeatureHydratorIdentifier = FeatureHydratorIdentifier(

"AdvertiserBrandSafetySettings")

override val features: Set[Feature[\_, \_]] = Set(AdvertiserBrandSafetySettingsFeature)

override def apply(

query: Query,

candidate: Candidate,

existingFeatures: FeatureMap

): Stitch[FeatureMap] = {

val featureMapFuture: Future[FeatureMap] = advertiserBrandSafetySettingsStore

.get(candidate.adImpression.advertiserId)

.map { advertiserBrandSafetySettingsOpt =>

FeatureMapBuilder()

.add(AdvertiserBrandSafetySettingsFeature, advertiserBrandSafetySettingsOpt).build()

}

Stitch.callFuture(featureMapFuture)

}

}