package com.twitter.home\_mixer.functional\_component.filter

import com.twitter.common\_internal.analytics.twitter\_client\_user\_agent\_parser.UserAgent

import com.twitter.home\_mixer.model.HomeFeatures.IsAncestorCandidateFeature

import com.twitter.home\_mixer.model.HomeFeatures.PersistenceEntriesFeature

import com.twitter.product\_mixer.component\_library.model.candidate.TweetCandidate

import com.twitter.product\_mixer.core.functional\_component.filter.Filter

import com.twitter.product\_mixer.core.functional\_component.filter.FilterResult

import com.twitter.product\_mixer.core.model.common.CandidateWithFeatures

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

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

import com.twitter.stitch.Stitch

import com.twitter.timelinemixer.injection.store.persistence.TimelinePersistenceUtils

import com.twitter.timelines.util.client\_info.ClientPlatform

object PreviouslyServedAncestorsFilter

extends Filter[PipelineQuery, TweetCandidate]

with TimelinePersistenceUtils {

override val identifier: FilterIdentifier = FilterIdentifier("PreviouslyServedAncestors")

override def apply(

query: PipelineQuery,

candidates: Seq[CandidateWithFeatures[TweetCandidate]]

): Stitch[FilterResult[TweetCandidate]] = {

val clientPlatform = ClientPlatform.fromQueryOptions(

clientAppId = query.clientContext.appId,

userAgent = query.clientContext.userAgent.flatMap(UserAgent.fromString))

val entries =

query.features.map(\_.getOrElse(PersistenceEntriesFeature, Seq.empty)).toSeq.flatten

val tweetIds = applicableResponses(clientPlatform, entries)

.flatMap(\_.entries.flatMap(\_.tweetIds(includeSourceTweets = true))).toSet

val ancestorIds =

candidates

.filter(\_.features.getOrElse(IsAncestorCandidateFeature, false)).map(\_.candidate.id).toSet

val (removed, kept) =

candidates

.map(\_.candidate).partition(candidate =>

tweetIds.contains(candidate.id) && ancestorIds.contains(candidate.id))

Stitch.value(FilterResult(kept = kept, removed = removed))

}

}