package com.twitter.home\_mixer.product.following

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

import com.twitter.home\_mixer.functional\_component.decorator.builder.HomeAdsClientEventDetailsBuilder

import com.twitter.home\_mixer.functional\_component.gate.ExcludeSoftUserGate

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

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

import com.twitter.home\_mixer.param.HomeGlobalParams

import com.twitter.home\_mixer.param.HomeGlobalParams.EnableAdvertiserBrandSafetySettingsFeatureHydratorParam

import com.twitter.home\_mixer.product.following.model.FollowingQuery

import com.twitter.home\_mixer.product.following.param.FollowingParam.EnableAdsCandidatePipelineParam

import com.twitter.home\_mixer.product.following.param.FollowingParam.EnableFastAds

import com.twitter.home\_mixer.service.HomeMixerAlertConfig

import com.twitter.product\_mixer.component\_library.candidate\_source.ads.AdsProdThriftCandidateSource

import com.twitter.product\_mixer.component\_library.decorator.urt.UrtItemCandidateDecorator

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.contextual\_ref.ContextualTweetRefBuilder

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.ad.AdsCandidateUrtItemBuilder

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.metadata.ClientEventInfoBuilder

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

import com.twitter.product\_mixer.component\_library.feature\_hydrator.candidate.param\_gated.ParamGatedCandidateFeatureHydrator

import com.twitter.product\_mixer.component\_library.gate.NonEmptyCandidatesGate

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

import com.twitter.product\_mixer.component\_library.pipeline.candidate.ads.AdsDependentCandidatePipelineConfig

import com.twitter.product\_mixer.component\_library.pipeline.candidate.ads.AdsDependentCandidatePipelineConfigBuilder

import com.twitter.product\_mixer.component\_library.pipeline.candidate.ads.CountCandidatesFromPipelines

import com.twitter.product\_mixer.component\_library.pipeline.candidate.ads.PipelineScopedOrganicItems

import com.twitter.product\_mixer.component\_library.pipeline.candidate.ads.ValidAdImpressionIdFilter

import com.twitter.product\_mixer.core.functional\_component.common.CandidateScope

import com.twitter.product\_mixer.core.gate.ParamNotGate

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

import com.twitter.product\_mixer.core.model.marshalling.response.rtf.safety\_level.TimelineHomePromotedHydrationSafetyLevel

import com.twitter.product\_mixer.core.model.marshalling.response.urt.contextual\_ref.TweetHydrationContext

import com.twitter.timelines.injection.scribe.InjectionScribeUtil

import com.twitter.timelineservice.suggests.{thriftscala => st}

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class FollowingAdsCandidatePipelineBuilder @Inject() (

adsCandidatePipelineConfigBuilder: AdsDependentCandidatePipelineConfigBuilder,

adsCandidateSource: AdsProdThriftCandidateSource,

advertiserBrandSafetySettingsFeatureHydrator: AdvertiserBrandSafetySettingsFeatureHydrator[

FollowingQuery,

AdsCandidate

]) {

private val identifier: CandidatePipelineIdentifier = CandidatePipelineIdentifier("FollowingAds")

private val suggestType = st.SuggestType.Promoted

private val clientEventInfoBuilder = ClientEventInfoBuilder(

component = InjectionScribeUtil.scribeComponent(suggestType).get,

detailsBuilder = Some(HomeAdsClientEventDetailsBuilder(Some(suggestType.name)))

)

private val contextualTweetRefBuilder = ContextualTweetRefBuilder(

TweetHydrationContext(

safetyLevelOverride = Some(TimelineHomePromotedHydrationSafetyLevel),

outerTweetContext = None

))

private val decorator = UrtItemCandidateDecorator(

AdsCandidateUrtItemBuilder(

tweetClientEventInfoBuilder = Some(clientEventInfoBuilder),

contextualTweetRefBuilder = Some(contextualTweetRefBuilder)

))

private val alerts = Seq(

HomeMixerAlertConfig.BusinessHours.defaultSuccessRateAlert(),

HomeMixerAlertConfig.BusinessHours.defaultEmptyResponseRateAlert()

)

def build(

organicCandidatePipelines: CandidateScope

): AdsDependentCandidatePipelineConfig[FollowingQuery] =

adsCandidatePipelineConfigBuilder.build[FollowingQuery](

adsCandidateSource = adsCandidateSource,

identifier = identifier,

adsDisplayLocationBuilder = query =>

if (query.params.getBoolean(EnableFastAds)) ads.DisplayLocation.TimelineHomeReverseChron

else ads.DisplayLocation.TimelineHome,

getOrganicItems = PipelineScopedOrganicItems(

pipelines = organicCandidatePipelines,

textFeature = TweetTextFeature,

languageFeature = TweetLanguageFeature

),

countNumOrganicItems = CountCandidatesFromPipelines(organicCandidatePipelines),

supportedClientParam = Some(EnableAdsCandidatePipelineParam),

gates = Seq(

ParamNotGate(

name = "AdsDisableInjectionBasedOnUserRole",

param = HomeGlobalParams.AdsDisableInjectionBasedOnUserRoleParam

),

ExcludeSoftUserGate,

NonEmptyCandidatesGate(organicCandidatePipelines)

),

filters = Seq(ValidAdImpressionIdFilter),

postFilterFeatureHydration = Seq(

ParamGatedCandidateFeatureHydrator(

EnableAdvertiserBrandSafetySettingsFeatureHydratorParam,

advertiserBrandSafetySettingsFeatureHydrator

)

),

decorator = Some(decorator),

alerts = alerts,

urtRequest = Some(true),

)

}