package com.twitter.home\_mixer.product.for\_you

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

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

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

import com.twitter.home\_mixer.functional\_component.decorator.urt.builder.HomeFeedbackActionInfoBuilder

import com.twitter.home\_mixer.functional\_component.decorator.urt.builder.HomeTweetSocialContextBuilder

import com.twitter.home\_mixer.functional\_component.feature\_hydrator.InNetworkFeatureHydrator

import com.twitter.home\_mixer.functional\_component.feature\_hydrator.NamesFeatureHydrator

import com.twitter.home\_mixer.functional\_component.feature\_hydrator.TweetypieFeatureHydrator

import com.twitter.home\_mixer.functional\_component.filter.InvalidConversationModuleFilter

import com.twitter.home\_mixer.functional\_component.filter.InvalidSubscriptionTweetFilter

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

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

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

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

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

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

import com.twitter.home\_mixer.product.for\_you.candidate\_source.ScoredTweetWithConversationMetadata

import com.twitter.home\_mixer.product.for\_you.candidate\_source.ScoredTweetsProductCandidateSource

import com.twitter.home\_mixer.product.for\_you.feature\_hydrator.FocalTweetFeatureHydrator

import com.twitter.home\_mixer.product.for\_you.filter.SocialContextFilter

import com.twitter.home\_mixer.product.for\_you.model.ForYouQuery

import com.twitter.home\_mixer.product.for\_you.param.ForYouParam.EnableScoredTweetsCandidatePipelineParam

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

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

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

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.tweet.TweetCandidateUrtItemBuilder

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.timeline\_module.ManualModuleId

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.timeline\_module.StaticModuleDisplayTypeBuilder

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.timeline\_module.TimelineModuleBuilder

import com.twitter.product\_mixer.component\_library.filter.FeatureFilter

import com.twitter.product\_mixer.component\_library.filter.PredicateFeatureFilter

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

import com.twitter.product\_mixer.core.functional\_component.candidate\_source.CandidateSource

import com.twitter.product\_mixer.core.functional\_component.decorator.CandidateDecorator

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

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

import com.twitter.product\_mixer.core.functional\_component.gate.Gate

import com.twitter.product\_mixer.core.functional\_component.transformer.CandidateFeatureTransformer

import com.twitter.product\_mixer.core.functional\_component.transformer.CandidatePipelineQueryTransformer

import com.twitter.product\_mixer.core.functional\_component.transformer.CandidatePipelineResultsTransformer

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

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

import com.twitter.product\_mixer.core.model.marshalling.response.urt.EntryNamespace

import com.twitter.product\_mixer.core.model.marshalling.response.urt.timeline\_module.VerticalConversation

import com.twitter.product\_mixer.core.pipeline.candidate.CandidatePipelineConfig

import com.twitter.timelines.configapi.decider.DeciderParam

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class ForYouScoredTweetsCandidatePipelineConfig @Inject() (

scoredTweetsProductCandidateSource: ScoredTweetsProductCandidateSource,

focalTweetFeatureHydrator: FocalTweetFeatureHydrator,

namesFeatureHydrator: NamesFeatureHydrator,

tweetypieFeatureHydrator: TweetypieFeatureHydrator,

invalidSubscriptionTweetFilter: InvalidSubscriptionTweetFilter,

homeFeedbackActionInfoBuilder: HomeFeedbackActionInfoBuilder,

homeTweetSocialContextBuilder: HomeTweetSocialContextBuilder)

extends CandidatePipelineConfig[

ForYouQuery,

ForYouQuery,

ScoredTweetWithConversationMetadata,

TweetCandidate

] {

override val identifier: CandidatePipelineIdentifier =

CandidatePipelineIdentifier("ForYouScoredTweets")

private val TweetypieHydratedFilterId = "TweetypieHydrated"

private val QuotedTweetDroppedFilterId = "QuotedTweetDropped"

private val OutOfNetworkNSFWFilterId = "OutOfNetworkNSFW"

private val ConversationModuleNamespace = EntryNamespace("home-conversation")

override val gates: Seq[Gate[ForYouQuery]] = Seq(SupportedLanguagesGate)

override val candidateSource: CandidateSource[ForYouQuery, ScoredTweetWithConversationMetadata] =

scoredTweetsProductCandidateSource

override val enabledDeciderParam: Option[DeciderParam[Boolean]] =

Some(EnableScoredTweetsCandidatePipelineParam)

override val queryTransformer: CandidatePipelineQueryTransformer[ForYouQuery, ForYouQuery] =

identity

override val featuresFromCandidateSourceTransformers: Seq[

CandidateFeatureTransformer[ScoredTweetWithConversationMetadata]

] = Seq(ForYouScoredTweetsResponseFeatureTransformer)

override val resultTransformer: CandidatePipelineResultsTransformer[

ScoredTweetWithConversationMetadata,

TweetCandidate

] = { sourceResults => TweetCandidate(sourceResults.tweetId) }

override val preFilterFeatureHydrationPhase1: Seq[

BaseCandidateFeatureHydrator[ForYouQuery, TweetCandidate, \_]

] = Seq(InNetworkFeatureHydrator, namesFeatureHydrator, tweetypieFeatureHydrator)

override val filters: Seq[Filter[ForYouQuery, TweetCandidate]] = Seq(

FeatureFilter.fromFeature(FilterIdentifier(TweetypieHydratedFilterId), IsHydratedFeature),

PredicateFeatureFilter.fromPredicate(

FilterIdentifier(QuotedTweetDroppedFilterId),

shouldKeepCandidate = { features => !features.getOrElse(QuotedTweetDroppedFeature, false) }

),

PredicateFeatureFilter.fromPredicate(

FilterIdentifier(OutOfNetworkNSFWFilterId),

shouldKeepCandidate = { features =>

features.getOrElse(InNetworkFeature, false) ||

!features.getOrElse(IsNsfwFeature, false)

}

),

SocialContextFilter,

invalidSubscriptionTweetFilter,

InvalidConversationModuleFilter

)

override val postFilterFeatureHydration: Seq[

BaseCandidateFeatureHydrator[ForYouQuery, TweetCandidate, \_]

] = Seq(focalTweetFeatureHydrator)

override val decorator: Option[CandidateDecorator[ForYouQuery, TweetCandidate]] = {

val clientEventInfoBuilder = HomeClientEventInfoBuilder()

val tweetItemBuilder = TweetCandidateUrtItemBuilder(

clientEventInfoBuilder = clientEventInfoBuilder,

socialContextBuilder = Some(homeTweetSocialContextBuilder),

timelinesScoreInfoBuilder = Some(HomeTimelinesScoreInfoBuilder),

feedbackActionInfoBuilder = Some(homeFeedbackActionInfoBuilder)

)

val tweetDecorator = UrtItemCandidateDecorator(tweetItemBuilder)

val moduleBuilder = TimelineModuleBuilder(

entryNamespace = ConversationModuleNamespace,

clientEventInfoBuilder = clientEventInfoBuilder,

moduleIdGeneration = ManualModuleId(0L),

displayTypeBuilder = StaticModuleDisplayTypeBuilder(VerticalConversation),

metadataBuilder = Some(HomeConversationModuleMetadataBuilder())

)

Some(

UrtMultipleModulesDecorator(

urtItemCandidateDecorator = tweetDecorator,

moduleBuilder = moduleBuilder,

groupByKey = (\_, \_, candidateFeatures) =>

candidateFeatures.getOrElse(ConversationModuleFocalTweetIdFeature, None)

))

}

override val alerts = Seq(

HomeMixerAlertConfig.BusinessHours.defaultSuccessRateAlert(),

HomeMixerAlertConfig.BusinessHours.defaultEmptyResponseRateAlert(10, 20)

)

}