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.PerspectiveFilteredSocialContextFeatureHydrator

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

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

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

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.filter.RejectTweetFromViewerFilter

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

import com.twitter.home\_mixer.functional\_component.scorer.FeedbackFatigueScorer

import com.twitter.home\_mixer.functional\_component.scorer.OONTweetScalingScorer

import com.twitter.home\_mixer.marshaller.timelines.DeviceContextMarshaller

import com.twitter.home\_mixer.marshaller.timelines.TimelineServiceCursorMarshaller

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.model.HomeFeatures.TimelineServiceTweetsFeature

import com.twitter.home\_mixer.model.request.DeviceContext

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.EnableTimelineScorerCandidatePipelineParam

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

import com.twitter.product\_mixer.component\_library.candidate\_source.timeline\_scorer.ScoredTweetCandidateWithFocalTweet

import com.twitter.product\_mixer.component\_library.candidate\_source.timeline\_scorer.TimelineScorerCandidateSource

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.BaseCandidateSource

import com.twitter.product\_mixer.core.functional\_component.common.alert.Alert

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.scorer.Scorer

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.FSParam

import com.twitter.timelines.model.candidate.CandidateTweetSourceId

import com.twitter.timelines.service.{thriftscala => tst}

import com.twitter.timelinescorer.{thriftscala => t}

import com.twitter.timelineservice.{thriftscala => tlst}

import javax.inject.Inject

import javax.inject.Singleton

/\*\*

\* Candidate Pipeline Config that fetches tweets from the Timeline Scorer Candidate Source

\*/

@Singleton

class ForYouTimelineScorerCandidatePipelineConfig @Inject() (

timelineScorerCandidateSource: TimelineScorerCandidateSource,

deviceContextMarshaller: DeviceContextMarshaller,

tweetypieFeatureHydrator: TweetypieFeatureHydrator,

sgsValidSocialContextFeatureHydrator: SGSValidSocialContextFeatureHydrator,

perspectiveFilteredSocialContextFeatureHydrator: PerspectiveFilteredSocialContextFeatureHydrator,

namesFeatureHydrator: NamesFeatureHydrator,

focalTweetFeatureHydrator: FocalTweetFeatureHydrator,

invalidSubscriptionTweetFilter: InvalidSubscriptionTweetFilter,

homeFeedbackActionInfoBuilder: HomeFeedbackActionInfoBuilder,

homeTweetSocialContextBuilder: HomeTweetSocialContextBuilder)

extends CandidatePipelineConfig[

ForYouQuery,

t.ScoredTweetsRequest,

ScoredTweetCandidateWithFocalTweet,

TweetCandidate

] {

override val identifier: CandidatePipelineIdentifier =

CandidatePipelineIdentifier("ForYouTimelineScorerTweets")

private val TweetypieHydratedFilterId = "TweetypieHydrated"

private val QuotedTweetDroppedFilterId = "QuotedTweetDropped"

private val OutOfNetworkNSFWFilterId = "OutOfNetworkNSFW"

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

override val supportedClientParam: Option[FSParam[Boolean]] =

Some(EnableTimelineScorerCandidatePipelineParam)

override val candidateSource: BaseCandidateSource[

t.ScoredTweetsRequest,

ScoredTweetCandidateWithFocalTweet

] = timelineScorerCandidateSource

override val queryTransformer: CandidatePipelineQueryTransformer[

ForYouQuery,

t.ScoredTweetsRequest

] = { query =>

val deviceContext = query.deviceContext.getOrElse(DeviceContext.Empty)

val scoredTweetsRequestContext = t.v1.ScoredTweetsRequestContext(

contextualUserId = query.clientContext.userId,

timelineId = query.clientContext.userId.map(tlst.TimelineId(tlst.TimelineType.Home, \_, None)),

deviceContext = Some(deviceContextMarshaller(deviceContext, query.clientContext)),

seenTweetIds = query.seenTweetIds,

contextualUserContext = Some(tst.ContextualUserContext(query.clientContext.userRoles)),

timelineRequestCursor = query.pipelineCursor.flatMap(TimelineServiceCursorMarshaller(\_))

)

val candidateTweetSourceIds = Seq(

CandidateTweetSourceId.RecycledTweet,

CandidateTweetSourceId.OrganicTweet,

CandidateTweetSourceId.AncestorsOnlyOrganicTweet,

CandidateTweetSourceId.BackfillOrganicTweet,

CandidateTweetSourceId.CroonTweet,

CandidateTweetSourceId.RecommendedTweet,

CandidateTweetSourceId.FrsTweet,

CandidateTweetSourceId.ListTweet,

CandidateTweetSourceId.RecommendedTrendTweet,

CandidateTweetSourceId.PopularTopicTweet

)

val timelineServiceTweets =

query.features.map(\_.getOrElse(TimelineServiceTweetsFeature, Seq.empty)).getOrElse(Seq.empty)

val timelineEntries = timelineServiceTweets.map { id =>

tlst.TimelineEntry.Tweet(tlst.Tweet(statusId = id, sortIndex = id))

}

t.ScoredTweetsRequest.V1(

t.v1.ScoredTweetsRequest(

scoredTweetsRequestContext = Some(scoredTweetsRequestContext),

candidateTweetSourceIds =

Some(candidateTweetSourceIds.flatMap(CandidateTweetSourceId.toThrift)),

maxResultsCount = query.requestedMaxResults,

organicTimeline = Some(

tlst.Timeline(

timelineId = tlst.TimelineId(

timelineType = tlst.TimelineType.Home,

id = query.getRequiredUserId,

canonicalTimelineId = None),

entries = timelineEntries,

modules = tlst.TimelineModules()

))

)

)

}

override val featuresFromCandidateSourceTransformers: Seq[

CandidateFeatureTransformer[ScoredTweetCandidateWithFocalTweet]

] = Seq(ForYouTimelineScorerResponseFeatureTransformer)

override val resultTransformer: CandidatePipelineResultsTransformer[

ScoredTweetCandidateWithFocalTweet,

TweetCandidate

] = { candidateWithFocalTweetId =>

TweetCandidate(id = candidateWithFocalTweetId.candidate.tweetId)

}

override val preFilterFeatureHydrationPhase1: Seq[

BaseCandidateFeatureHydrator[ForYouQuery, TweetCandidate, \_]

] = Seq(

namesFeatureHydrator,

tweetypieFeatureHydrator,

InNetworkFeatureHydrator,

sgsValidSocialContextFeatureHydrator,

perspectiveFilteredSocialContextFeatureHydrator,

)

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

RetweetDeduplicationFilter,

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)

}

),

FeedbackFatigueFilter,

RejectTweetFromViewerFilter,

SocialContextFilter,

invalidSubscriptionTweetFilter,

InvalidConversationModuleFilter

)

override val postFilterFeatureHydration: Seq[

BaseCandidateFeatureHydrator[ForYouQuery, TweetCandidate, \_]

] = Seq(focalTweetFeatureHydrator)

override val scorers: Seq[Scorer[ForYouQuery, TweetCandidate]] =

Seq(OONTweetScalingScorer, FeedbackFatigueScorer)

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[Alert] = Seq(

HomeMixerAlertConfig.BusinessHours.defaultSuccessRateAlert(),

HomeMixerAlertConfig.BusinessHours.defaultEmptyResponseRateAlert(10, 20)

)

}