package com.twitter.home\_mixer.functional\_component.decorator.urt.builder

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

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

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

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

import com.twitter.home\_mixer.util.CandidatesUtil

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

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

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.metadata.BaseFeedbackActionInfoBuilder

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

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

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

import com.twitter.timelines.util.FeedbackMetadataSerializer

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class HomeFeedbackActionInfoBuilder @Inject() (

notInterestedTopicFeedbackActionBuilder: NotInterestedTopicFeedbackActionBuilder,

dontLikeFeedbackActionBuilder: DontLikeFeedbackActionBuilder)

extends BaseFeedbackActionInfoBuilder[PipelineQuery, TweetCandidate] {

override def apply(

query: PipelineQuery,

candidate: TweetCandidate,

candidateFeatures: FeatureMap

): Option[FeedbackActionInfo] = {

val supportedProduct = query.product match {

case FollowingProduct => query.params(EnableNahFeedbackInfoParam)

case ForYouProduct => true

case \_ => false

}

val isAuthoredByViewer = CandidatesUtil.isAuthoredByViewer(query, candidateFeatures)

if (supportedProduct && !isAuthoredByViewer) {

val feedbackActions = Seq(

notInterestedTopicFeedbackActionBuilder(candidateFeatures),

dontLikeFeedbackActionBuilder(query, candidate, candidateFeatures)

).flatten

val feedbackMetadata = FeedbackMetadataSerializer.serialize(

t.FeedbackMetadata(injectionType = candidateFeatures.getOrElse(SuggestTypeFeature, None)))

Some(

FeedbackActionInfo(

feedbackActions = feedbackActions,

feedbackMetadata = Some(feedbackMetadata),

displayContext = None,

clientEventInfo = None

))

} else None

}

}