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

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

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

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

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

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

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

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

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

case class TopicNotInterestedFeedbackActionInfoBuilder[-Query <: PipelineQuery]()

extends BaseFeedbackActionInfoBuilder[Query, BaseTopicCandidate] {

override def apply(

query: Query,

topicCandidate: BaseTopicCandidate,

candidateFeatures: FeatureMap

): Option[FeedbackActionInfo] = {

Some(

FeedbackActionInfo(

feedbackActions = Seq(

FeedbackAction(

feedbackType = RichBehavior,

richBehavior = Some(

RichFeedbackBehaviorMarkNotInterestedTopic(topicCandidate.id.toString)

),

hasUndoAction = Some(true),

prompt = None,

confirmation = None,

feedbackUrl = None,

clientEventInfo = None,

childFeedbackActions = None,

confirmationDisplayType = None,

icon = None,

subprompt = None,

encodedFeedbackRequest = None

)

),

feedbackMetadata = None,

displayContext = None,

clientEventInfo = None

))

}

}