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

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

import com.twitter.product\_mixer.core.feature.Feature

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.timeline\_module.BaseModuleMetadataBuilder

import com.twitter.product\_mixer.core.model.common.CandidateWithFeatures

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

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

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

case class ConversationModuleMetadataBuilder[

Query <: PipelineQuery,

Candidate <: BaseTweetCandidate

](

ancestorIdsFeature: Feature[\_, Seq[Long]],

allIdsOrdering: Ordering[Long])

extends BaseModuleMetadataBuilder[Query, Candidate] {

override def apply(

query: Query,

candidates: Seq[CandidateWithFeatures[Candidate]]

): ModuleMetadata = {

val ancestors = candidates.last.features.getOrElse(ancestorIdsFeature, Seq.empty)

val sortedAllTweetIds = (candidates.last.candidate.id +: ancestors).sorted(allIdsOrdering)

ModuleMetadata(

adsMetadata = None,

conversationMetadata = Some(

ModuleConversationMetadata(

allTweetIds = Some(sortedAllTweetIds),

socialContext = None,

enableDeduplication = Some(true)

)),

gridCarouselMetadata = None

)

}

}