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

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.article.ArticleCandidateUrtItemBuilder.ArticleClientEventInfoElement

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

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

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

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

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

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.social\_context.BaseSocialContextBuilder

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.article.ArticleDisplayType

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.article.ArticleItem

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.article.ArticleSeedType

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.article.FollowingListSeed

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

object ArticleCandidateUrtItemBuilder {

val ArticleClientEventInfoElement: String = "article"

}

case class ArticleCandidateUrtItemBuilder[

-Query <: PipelineQuery,

Candidate <: BaseArticleCandidate

](

clientEventInfoBuilder: BaseClientEventInfoBuilder[Query, Candidate],

articleSeedType: ArticleSeedType = FollowingListSeed,

feedbackActionInfoBuilder: Option[

BaseFeedbackActionInfoBuilder[Query, Candidate]

] = None,

displayType: Option[ArticleDisplayType] = None,

socialContextBuilder: Option[BaseSocialContextBuilder[Query, Candidate]] = None,

) extends CandidateUrtEntryBuilder[Query, Candidate, ArticleItem] {

override def apply(

query: Query,

articleCandidate: Candidate,

candidateFeatures: FeatureMap

): ArticleItem = ArticleItem(

id = articleCandidate.id,

sortIndex = None, // Sort indexes are automatically set in the domain marshaller phase

clientEventInfo = clientEventInfoBuilder(

query,

articleCandidate,

candidateFeatures,

Some(ArticleClientEventInfoElement)),

feedbackActionInfo =

feedbackActionInfoBuilder.flatMap(\_.apply(query, articleCandidate, candidateFeatures)),

displayType = displayType,

socialContext =

socialContextBuilder.flatMap(\_.apply(query, articleCandidate, candidateFeatures)),

articleSeedType = articleSeedType

)

}