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

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.generic\_summary.GenericSummaryCandidateUrtItemBuilder.GenericSummaryClientEventInfoElement

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

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.richtext.BaseRichTextBuilder

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.generic\_summary.GenericSummaryItem

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.generic\_summary.GenericSummaryItemDisplayType

import com.twitter.product\_mixer.core.model.marshalling.response.urt.media.Media

import com.twitter.product\_mixer.core.model.marshalling.response.urt.promoted.PromotedMetadata

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

import com.twitter.util.Time

object GenericSummaryCandidateUrtItemBuilder {

val GenericSummaryClientEventInfoElement: String = "genericsummary"

}

case class GenericSummaryCandidateUrtItemBuilder[-Query <: PipelineQuery](

clientEventInfoBuilder: BaseClientEventInfoBuilder[Query, GenericSummaryCandidate],

headlineRichTextBuilder: BaseRichTextBuilder[Query, GenericSummaryCandidate],

displayType: GenericSummaryItemDisplayType,

genericSummaryContextCandidateUrtItemBuilder: Option[

GenericSummaryContextBuilder[Query, GenericSummaryCandidate]

] = None,

genericSummaryActionCandidateUrtItemBuilder: Option[

GenericSummaryActionBuilder[Query, GenericSummaryCandidate]

] = None,

timestamp: Option[Time] = None,

userAttributionIds: Option[Seq[Long]] = None,

media: Option[Media] = None,

promotedMetadata: Option[PromotedMetadata] = None,

feedbackActionInfoBuilder: Option[BaseFeedbackActionInfoBuilder[Query, GenericSummaryCandidate]] =

None)

extends CandidateUrtEntryBuilder[Query, GenericSummaryCandidate, GenericSummaryItem] {

override def apply(

query: Query,

genericSummaryCandidate: GenericSummaryCandidate,

candidateFeatures: FeatureMap

): GenericSummaryItem = GenericSummaryItem(

id = genericSummaryCandidate.id,

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

clientEventInfo = clientEventInfoBuilder(

query,

genericSummaryCandidate,

candidateFeatures,

Some(GenericSummaryClientEventInfoElement)),

feedbackActionInfo =

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

headline = headlineRichTextBuilder.apply(query, genericSummaryCandidate, candidateFeatures),

displayType = displayType,

userAttributionIds = userAttributionIds.getOrElse(Seq.empty),

media = media,

context = genericSummaryContextCandidateUrtItemBuilder.map(

\_.apply(query, genericSummaryCandidate, candidateFeatures)),

timestamp = timestamp,

onClickAction = genericSummaryActionCandidateUrtItemBuilder.map(

\_.apply(query, genericSummaryCandidate, candidateFeatures)),

promotedMetadata = promotedMetadata

)

}