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

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.trend.TrendCandidateUrtItemBuilder.TrendsClientEventInfoElement

import com.twitter.product\_mixer.component\_library.model.candidate.trends\_events.TrendDescription

import com.twitter.product\_mixer.component\_library.model.candidate.trends\_events.TrendDomainContext

import com.twitter.product\_mixer.component\_library.model.candidate.trends\_events.TrendGroupedTrends

import com.twitter.product\_mixer.component\_library.model.candidate.trends\_events.TrendNormalizedTrendName

import com.twitter.product\_mixer.component\_library.model.candidate.trends\_events.TrendTrendName

import com.twitter.product\_mixer.component\_library.model.candidate.trends\_events.TrendTweetCount

import com.twitter.product\_mixer.component\_library.model.candidate.trends\_events.TrendUrl

import com.twitter.product\_mixer.component\_library.model.candidate.trends\_events.UnifiedTrendCandidate

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.promoted.BasePromotedMetadataBuilder

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

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.trend.TrendItem

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

object TrendCandidateUrtItemBuilder {

final val TrendsClientEventInfoElement = "trend"

}

case class TrendCandidateUrtItemBuilder[Query <: PipelineQuery](

trendMetaDescriptionBuilder: TrendMetaDescriptionBuilder[Query, UnifiedTrendCandidate],

promotedMetadataBuilder: BasePromotedMetadataBuilder[Query, UnifiedTrendCandidate],

clientEventInfoBuilder: BaseClientEventInfoBuilder[Query, UnifiedTrendCandidate],

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

None)

extends CandidateUrtEntryBuilder[Query, UnifiedTrendCandidate, TimelineItem] {

override def apply(

query: Query,

candidate: UnifiedTrendCandidate,

candidateFeatures: FeatureMap

): TimelineItem = {

TrendItem(

id = candidate.id,

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

clientEventInfo = clientEventInfoBuilder(

query = query,

candidate = candidate,

candidateFeatures = candidateFeatures,

element = Some(TrendsClientEventInfoElement)

),

feedbackActionInfo = None,

normalizedTrendName = candidateFeatures.get(TrendNormalizedTrendName),

trendName = candidateFeatures.get(TrendTrendName),

url = candidateFeatures.get(TrendUrl),

description = candidateFeatures.getOrElse(TrendDescription, None),

metaDescription = trendMetaDescriptionBuilder(query, candidate, candidateFeatures),

tweetCount = candidateFeatures.getOrElse(TrendTweetCount, None),

domainContext = candidateFeatures.getOrElse(TrendDomainContext, None),

promotedMetadata = promotedMetadataBuilder(

query = query,

candidate = candidate,

candidateFeatures = candidateFeatures

),

groupedTrends = candidateFeatures.getOrElse(TrendGroupedTrends, None)

)

}

}