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

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.event\_summary.EventCandidateUrtItemBuilder.EventClientEventInfoElement

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

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

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

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

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

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

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.model.marshalling.response.urt.TimelineItem

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.event.EventSummaryItem

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

object EventCandidateUrtItemBuilder {

val EventClientEventInfoElement = "event"

}

case class EventCandidateUrtItemBuilder[Query <: PipelineQuery](

clientEventInfoBuilder: BaseClientEventInfoBuilder[Query, UnifiedEventCandidate],

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

None)

extends CandidateUrtEntryBuilder[Query, UnifiedEventCandidate, TimelineItem] {

override def apply(

query: Query,

candidate: UnifiedEventCandidate,

candidateFeatures: FeatureMap

): TimelineItem = {

EventSummaryItem(

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(EventClientEventInfoElement)

),

feedbackActionInfo =

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

title = candidateFeatures.get(EventTitleFeature),

displayType = candidateFeatures.get(EventDisplayType),

url = candidateFeatures.get(EventUrl),

image = candidateFeatures.getOrElse(EventImage, None),

timeString = candidateFeatures.getOrElse(EventTimeString, None)

)

}

}