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

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.card.CardCandidateUtrItemBuilder.CardClientEventInfoElement

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

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.metadata.BaseStr

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

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.card.CardDisplayType

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.card.CardItem

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

object CardCandidateUtrItemBuilder {

val CardClientEventInfoElement: String = "card"

}

case class CardCandidateUtrItemBuilder[-Query <: PipelineQuery](

clientEventInfoBuilder: BaseClientEventInfoBuilder[Query, CardCandidate],

cardUrlBuilder: BaseStr[Query, CardCandidate],

textBuilder: Option[BaseStr[Query, CardCandidate]],

subtextBuilder: Option[BaseStr[Query, CardCandidate]],

urlBuilder: Option[BaseUrlBuilder[Query, CardCandidate]],

cardDisplayType: Option[CardDisplayType],

feedbackActionInfoBuilder: Option[

BaseFeedbackActionInfoBuilder[Query, CardCandidate],

] = None)

extends CandidateUrtEntryBuilder[Query, CardCandidate, CardItem] {

override def apply(

query: Query,

cardCandidate: CardCandidate,

candidateFeatures: FeatureMap

): CardItem = CardItem(

id = cardCandidate.id,

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

clientEventInfo = clientEventInfoBuilder(

query,

cardCandidate,

candidateFeatures,

Some(CardClientEventInfoElement)),

feedbackActionInfo =

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

cardUrl = cardUrlBuilder(query, cardCandidate, candidateFeatures),

text = textBuilder.map(\_.apply(query, cardCandidate, candidateFeatures)),

subtext = textBuilder.map(\_.apply(query, cardCandidate, candidateFeatures)),

url = urlBuilder.map(\_.apply(query, cardCandidate, candidateFeatures)),

displayType = cardDisplayType

)

}