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

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.tile.TileCandidateUrtItemBuilder.TopicTileClientEventInfoElement

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

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.item.tile.StandardTileContent

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.tile.TileItem

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

object TileCandidateUrtItemBuilder {

val TopicTileClientEventInfoElement: String = "tile"

}

case class TileCandidateUrtItemBuilder[-Query <: PipelineQuery](

clientEventInfoBuilder: BaseClientEventInfoBuilder[Query, PromptCarouselTileCandidate],

feedbackActionInfoBuilder: Option[

BaseFeedbackActionInfoBuilder[Query, PromptCarouselTileCandidate]

] = None)

extends CandidateUrtEntryBuilder[Query, PromptCarouselTileCandidate, TileItem] {

override def apply(

query: Query,

tileCandidate: PromptCarouselTileCandidate,

candidateFeatures: FeatureMap

): TileItem = TileItem(

id = tileCandidate.id,

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

clientEventInfo = clientEventInfoBuilder(

query,

tileCandidate,

candidateFeatures,

Some(TopicTileClientEventInfoElement)),

title = "", //This data is ignored do

supportingText = "",

feedbackActionInfo =

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

image = None,

url = None,

content = StandardTileContent(

title = "",

supportingText = "",

badge = None

)

)

}