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

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMap

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

import com.twitter.product\_mixer.core.model.common.UniversalNoun

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.message.MessageAction

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.message.MessageTextAction

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

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

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

object MessageTextActionBuilder {

val MessageTextActionClientEventInfoElement: String = "message-text-action"

}

case class MessageTextActionBuilder[-Query <: PipelineQuery, -Candidate <: UniversalNoun[Any]](

textBuilder: BaseStr[Query, Candidate],

dismissOnClick: Boolean,

url: Option[String] = None,

clientEventInfo: Option[ClientEventInfo] = None,

onClickCallbacks: Option[List[Callback]] = None) {

def apply(

query: Query,

candidate: Candidate,

candidateFeatures: FeatureMap

): MessageTextAction = MessageTextAction(

text = textBuilder(query, candidate, candidateFeatures),

action = MessageAction(

dismissOnClick,

url,

clientEventInfo,

onClickCallbacks

)

)

}