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

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

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

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

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

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.BaseClientEventDetailsBuilder

/\*\*

\* Sets the [[ClientEventInfo]] with the `component` field set to [[component]]

\* @see [[http://go/client-events]]

\*/

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

component: String,

detailsBuilder: Option[BaseClientEventDetailsBuilder[Query, Candidate]] = None)

extends BaseClientEventInfoBuilder[Query, Candidate] {

override def apply(

query: Query,

candidate: Candidate,

candidateFeatures: FeatureMap,

element: Option[String]

): Option[ClientEventInfo] =

Some(

ClientEventInfo(

component = Some(component),

element = element,

details = detailsBuilder.flatMap(\_.apply(query, candidate, candidateFeatures)),

action = None,

entityToken = None)

)

}

/\*\*

\* In rare cases you might not want to send client event info. For

\* example, this might be set already on the client for some legacy

\* timelines.

\*/

object EmptyClientEventInfoBuilder

extends BaseClientEventInfoBuilder[PipelineQuery, UniversalNoun[Any]] {

override def apply(

query: PipelineQuery,

candidate: UniversalNoun[Any],

candidateFeatures: FeatureMap,

element: Option[String]

): Option[ClientEventInfo] = None

}