package com.twitter.home\_mixer.functional\_component.decorator.builder

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

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

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

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

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

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

import com.twitter.timelineservice.suggests.{thriftscala => st}

case class ListClientEventDetailsBuilder(suggestType: st.SuggestType)

extends BaseClientEventDetailsBuilder[PipelineQuery, UniversalNoun[Any]] {

override def apply(

query: PipelineQuery,

candidate: UniversalNoun[Any],

candidateFeatures: FeatureMap

): Option[ClientEventDetails] = {

val clientEventDetails = ClientEventDetails(

conversationDetails = None,

timelinesDetails = Some(

TimelinesDetails(

injectionType = Some(suggestType.name),

controllerData = None,

sourceData = None)),

articleDetails = None,

liveEventDetails = None,

commerceDetails = None

)

Some(clientEventDetails)

}

}