package com.twitter.product\_mixer.component\_library.premarshaller.urt.builder

import com.twitter.product\_mixer.component\_library.model.cursor.UrtPassThroughCursor

import com.twitter.product\_mixer.component\_library.premarshaller.cursor.UrtCursorSerializer

import com.twitter.product\_mixer.core.feature.Feature

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

import com.twitter.product\_mixer.core.model.marshalling.response.urt.operation.CursorType

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

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

case class PassThroughCursorBuilder[

-Query <: PipelineQuery with HasPipelineCursor[UrtPassThroughCursor]

](

cursorFeature: Feature[Query, String],

override val cursorType: CursorType)

extends UrtCursorBuilder[Query] {

override val includeOperation: IncludeInstruction[Query] = { (query, \_) =>

query.features.exists(\_.getOrElse(cursorFeature, "").nonEmpty)

}

override def cursorValue(

query: Query,

entries: Seq[TimelineEntry]

): String =

UrtCursorSerializer.serializeCursor(

UrtPassThroughCursor(

cursorSortIndex(query, entries),

query.features.map(\_.get(cursorFeature)).getOrElse(""),

cursorType = Some(cursorType)

)

)

}