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

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

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

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

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

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

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.PipelineCursorSerializer

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

import com.twitter.search.common.util.bloomfilter.AdaptiveLongIntBloomFilterBuilder

/\*\*

\* Builds [[UrtUnorderedBloomFilterCursor]] in the Bottom position

\*

\* @param idSelector Specifies the entry from which to derive the `id` field

\* @param serializer Converts the cursor to an encoded string

\*/

case class UnorderedBloomFilterBottomCursorBuilder(

idSelector: PartialFunction[UniversalNoun[\_], Long],

serializer: PipelineCursorSerializer[UrtUnorderedBloomFilterCursor] = UrtCursorSerializer)

extends UrtCursorBuilder[

PipelineQuery with HasPipelineCursor[UrtUnorderedBloomFilterCursor]

] {

override val cursorType: CursorType = BottomCursor

override def cursorValue(

query: PipelineQuery with HasPipelineCursor[UrtUnorderedBloomFilterCursor],

entries: Seq[TimelineEntry]

): String = {

val bloomFilter = query.pipelineCursor.map(\_.longIntBloomFilter)

val ids = entries.collect(idSelector)

val cursor = UrtUnorderedBloomFilterCursor(

initialSortIndex = nextBottomInitialSortIndex(query, entries),

longIntBloomFilter = AdaptiveLongIntBloomFilterBuilder.build(ids, bloomFilter)

)

serializer.serializeCursor(cursor)

}

}