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

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

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.pipeline.PipelineCursorSerializer

import com.twitter.timelines.configapi.Param

/\*\*

\* Builds [[UrtUnorderedExcludeIdsCursor]] in the Bottom position when we want to also exclude ids

\* of items inside a module. The reason we cannot use [[UnorderedExcludeIdsBottomCursorBuilder]] in

\* such case is that the excludeIdsSelector of [[UnorderedExcludeIdsBottomCursorBuilder]] is doing a

\* one to one mapping between entries and excluded ids, but in case of having a module, a module

\* entry can result in excluding a sequence of entries.

\*

\* @param excludedIdsMaxLengthParam The maximum length of the cursor

\* @param excludeIdsSelector Specifies the entry Ids to populate on the `excludedIds` field

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

\*/

case class UnorderedExcludeIdsSeqBottomCursorBuilder(

override val excludedIdsMaxLengthParam: Param[Int],

excludeIdsSelector: PartialFunction[UniversalNoun[\_], Seq[Long]],

override val serializer: PipelineCursorSerializer[UrtUnorderedExcludeIdsCursor] =

UrtCursorSerializer)

extends BaseUnorderedExcludeIdsBottomCursorBuilder {

override def excludeEntriesCollector(entries: Seq[TimelineEntry]): Seq[Long] =

entries.collect(excludeIdsSelector).flatten

}