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

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

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

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

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

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

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

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

sealed trait TerminateInstructionBuilder[Query <: PipelineQuery]

extends UrtInstructionBuilder[Query, TerminateTimelineInstruction] {

def direction: TimelineTerminationDirection

override def build(

query: Query,

entries: Seq[TimelineEntry]

): Seq[TerminateTimelineInstruction] =

if (includeInstruction(query, entries))

Seq(TerminateTimelineInstruction(terminateTimelineDirection = direction))

else Seq.empty

}

case class TerminateTopInstructionBuilder[Query <: PipelineQuery](

override val includeInstruction: IncludeInstruction[Query] = AlwaysInclude)

extends TerminateInstructionBuilder[Query] {

override val direction = TopTermination

}

case class TerminateBottomInstructionBuilder[Query <: PipelineQuery](

override val includeInstruction: IncludeInstruction[Query] = AlwaysInclude)

extends TerminateInstructionBuilder[Query] {

override val direction = BottomTermination

}

case class TerminateTopAndBottomInstructionBuilder[Query <: PipelineQuery](

override val includeInstruction: IncludeInstruction[Query] = AlwaysInclude)

extends TerminateInstructionBuilder[Query] {

override val direction = TopAndBottomTermination

}