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

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

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

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

trait IncludeInstruction[-Query <: PipelineQuery] { self =>

def apply(query: Query, entries: Seq[TimelineEntry]): Boolean

def inverse(): IncludeInstruction[Query] = new IncludeInstruction[Query] {

def apply(query: Query, entries: Seq[TimelineEntry]): Boolean = !self.apply(query, entries)

}

}

object AlwaysInclude extends IncludeInstruction[PipelineQuery] {

override def apply(query: PipelineQuery, entries: Seq[TimelineEntry]): Boolean = true

}

object IncludeOnFirstPage extends IncludeInstruction[PipelineQuery with HasPipelineCursor[\_]] {

override def apply(

query: PipelineQuery with HasPipelineCursor[\_],

entries: Seq[TimelineEntry]

): Boolean = query.isFirstPage

}

object IncludeAfterFirstPage extends IncludeInstruction[PipelineQuery with HasPipelineCursor[\_]] {

override def apply(

query: PipelineQuery with HasPipelineCursor[\_],

entries: Seq[TimelineEntry]

): Boolean = !query.isFirstPage

}