package com.twitter.home\_mixer.functional\_component.decorator.urt.builder

import com.twitter.product\_mixer.component\_library.premarshaller.urt.builder.AlwaysInclude

import com.twitter.product\_mixer.component\_library.premarshaller.urt.builder.IncludeInstruction

import com.twitter.product\_mixer.component\_library.premarshaller.urt.builder.UrtInstructionBuilder

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

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

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

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

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

case class AddEntriesWithReplaceAndShowAlertAndCoverInstructionBuilder[Query <: PipelineQuery](

override val includeInstruction: IncludeInstruction[Query] = AlwaysInclude)

extends UrtInstructionBuilder[Query, AddEntriesTimelineInstruction] {

override def build(

query: Query,

entries: Seq[TimelineEntry]

): Seq[AddEntriesTimelineInstruction] = {

if (includeInstruction(query, entries)) {

val entriesToAdd = entries

.filterNot(\_.isInstanceOf[ShowAlert])

.filterNot(\_.isInstanceOf[Cover])

.filter(\_.entryIdToReplace.isEmpty)

if (entriesToAdd.nonEmpty) Seq(AddEntriesTimelineInstruction(entriesToAdd))

else Seq.empty

} else

Seq.empty

}

}