package com.twitter.product\_mixer.core.model.marshalling.response.urt

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

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

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

sealed trait TimelineInstruction

case class AddEntriesTimelineInstruction(entries: Seq[TimelineEntry])

extends TimelineInstruction

with ContainsFeedbackActionInfos {

override def feedbackActionInfos: Seq[Option[FeedbackActionInfo]] =

entries.flatMap {

// Order is important, as entries that implement both ContainsFeedbackActionInfos and

// HasFeedbackActionInfo are expected to include both when implementing ContainsFeedbackActionInfos

case containsFeedbackActionInfos: ContainsFeedbackActionInfos =>

containsFeedbackActionInfos.feedbackActionInfos

case hasFeedbackActionInfo: HasFeedbackActionInfo =>

Seq(hasFeedbackActionInfo.feedbackActionInfo)

case \_ => Seq.empty

}

}

case class ReplaceEntryTimelineInstruction(entry: TimelineEntry)

extends TimelineInstruction

with ContainsFeedbackActionInfos {

override def feedbackActionInfos: Seq[Option[FeedbackActionInfo]] =

entry match {

// Order is important, as entries that implement both ContainsFeedbackActionInfos and

// HasFeedbackActionInfo are expected to include both when implementing ContainsFeedbackActionInfos

case containsFeedbackActionInfos: ContainsFeedbackActionInfos =>

containsFeedbackActionInfos.feedbackActionInfos

case hasFeedbackActionInfo: HasFeedbackActionInfo =>

Seq(hasFeedbackActionInfo.feedbackActionInfo)

case \_ => Seq.empty

}

}

case class AddToModuleTimelineInstruction(

moduleItems: Seq[ModuleItem],

moduleEntryId: String,

moduleItemEntryId: Option[String],

prepend: Option[Boolean])

extends TimelineInstruction

with ContainsFeedbackActionInfos {

override def feedbackActionInfos: Seq[Option[FeedbackActionInfo]] =

moduleItems.map(\_.item.feedbackActionInfo)

}

case class PinEntryTimelineInstruction(entry: TimelineEntry) extends TimelineInstruction

case class MarkEntriesUnreadInstruction(entryIds: Seq[String]) extends TimelineInstruction

case class ClearCacheTimelineInstruction() extends TimelineInstruction

sealed trait TimelineTerminationDirection

case object TopTermination extends TimelineTerminationDirection

case object BottomTermination extends TimelineTerminationDirection

case object TopAndBottomTermination extends TimelineTerminationDirection

case class TerminateTimelineInstruction(terminateTimelineDirection: TimelineTerminationDirection)

extends TimelineInstruction

case class ShowCoverInstruction(cover: Cover) extends TimelineInstruction

case class ShowAlertInstruction(showAlert: ShowAlert) extends TimelineInstruction