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

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

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

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

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

import com.twitter.product\_mixer.core.model.marshalling.response.urt.cover.FullCover.FullCoverEntryNamespace

import com.twitter.product\_mixer.core.model.marshalling.response.urt.cover.HalfCover.HalfCoverEntryNamespace

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

object HalfCover {

val HalfCoverEntryNamespace = EntryNamespace("half-cover")

}

case class HalfCover(

override val id: String,

override val sortIndex: Option[Long],

override val clientEventInfo: Option[ClientEventInfo],

content: HalfCoverContent)

extends Cover {

override val entryNamespace: EntryNamespace = HalfCoverEntryNamespace

// Note that sort index is not used for Covers, as they are not TimelineEntry and do not have entryId

override def withSortIndex(newSortIndex: Long): TimelineEntry =

copy(sortIndex = Some(newSortIndex))

// Not used for covers

override def feedbackActionInfo: Option[FeedbackActionInfo] = None

}

object FullCover {

val FullCoverEntryNamespace = EntryNamespace("full-cover")

}

case class FullCover(

override val id: String,

override val sortIndex: Option[Long],

override val clientEventInfo: Option[ClientEventInfo],

content: FullCoverContent)

extends Cover {

override val entryNamespace: EntryNamespace = FullCoverEntryNamespace

// Note that sort index is not used for Covers, as they are not TimelineEntry and do not have entryId

override def withSortIndex(newSortIndex: Long): TimelineEntry =

copy(sortIndex = Some(newSortIndex))

// Not used for covers

override def feedbackActionInfo: Option[FeedbackActionInfo] = None

}