package com.twitter.product\_mixer.core.functional\_component.marshaller.response.urt

import com.twitter.product\_mixer.core.functional\_component.marshaller.TransportMarshaller

import com.twitter.product\_mixer.core.functional\_component.marshaller.response.urt.cover.CoverContentMarshaller

import com.twitter.product\_mixer.core.functional\_component.marshaller.response.urt.metadata.ClientEventInfoMarshaller

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

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

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

import com.twitter.timelines.render.{thriftscala => urt}

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class CoverMarshaller @Inject() (

coverContentMarshaller: CoverContentMarshaller,

clientEventInfoMarshaller: ClientEventInfoMarshaller) {

def apply(cover: Cover): urt.ShowCover = cover match {

case halfCover: HalfCover =>

urt.ShowCover(

cover = coverContentMarshaller(halfCover.content),

clientEventInfo = cover.clientEventInfo.map(clientEventInfoMarshaller(\_)))

case fullCover: FullCover =>

urt.ShowCover(

cover = coverContentMarshaller(fullCover.content),

clientEventInfo = cover.clientEventInfo.map(clientEventInfoMarshaller(\_)))

}

}

class UnsupportedTimelineCoverException(cover: Cover)

extends UnsupportedOperationException(

"Unsupported timeline cover " + TransportMarshaller.getSimpleName(cover.getClass))