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

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

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

import com.twitter.product\_mixer.core.functional\_component.marshaller.response.urt.richtext.RichTextMarshaller

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

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

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class HalfCoverContentMarshaller @Inject() (

halfCoverDisplayTypeMarshaller: HalfCoverDisplayTypeMarshaller,

coverCtaMarshaller: CoverCtaMarshaller,

richTextMarshaller: RichTextMarshaller,

coverImageMarshaller: CoverImageMarshaller,

dismissInfoMarshaller: DismissInfoMarshaller,

callbackMarshaller: CallbackMarshaller) {

def apply(halfCover: HalfCoverContent): urt.Cover =

urt.Cover.HalfCover(

urt.HalfCover(

displayType = halfCoverDisplayTypeMarshaller(halfCover.displayType),

primaryText = richTextMarshaller(halfCover.primaryText),

primaryCoverCta = coverCtaMarshaller(halfCover.primaryCoverCta),

secondaryCoverCta = halfCover.secondaryCoverCta.map(coverCtaMarshaller(\_)),

secondaryText = halfCover.secondaryText.map(richTextMarshaller(\_)),

impressionCallbacks = halfCover.impressionCallbacks.map(\_.map(callbackMarshaller(\_))),

dismissible = halfCover.dismissible,

coverImage = halfCover.coverImage.map(coverImageMarshaller(\_)),

dismissInfo = halfCover.dismissInfo.map(dismissInfoMarshaller(\_))

))

}