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.metadata.ImageDisplayTypeMarshaller

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

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

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

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

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class FullCoverContentMarshaller @Inject() (

fullCoverDisplayTypeMarshaller: FullCoverDisplayTypeMarshaller,

coverCtaMarshaller: CoverCtaMarshaller,

richTextMarshaller: RichTextMarshaller,

imageVariantMarshaller: ImageVariantMarshaller,

dismissInfoMarshaller: DismissInfoMarshaller,

imageDisplayTypeMarshaller: ImageDisplayTypeMarshaller,

callbackMarshaller: CallbackMarshaller) {

def apply(fullCover: FullCoverContent): urt.Cover =

urt.Cover.FullCover(

urt.FullCover(

displayType = fullCoverDisplayTypeMarshaller(fullCover.displayType),

primaryText = richTextMarshaller(fullCover.primaryText),

primaryCoverCta = coverCtaMarshaller(fullCover.primaryCoverCta),

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

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

image = fullCover.imageVariant.map(imageVariantMarshaller(\_)),

details = fullCover.details.map(richTextMarshaller(\_)),

dismissInfo = fullCover.dismissInfo.map(dismissInfoMarshaller(\_)),

imageDisplayType = fullCover.imageDisplayType.map(imageDisplayTypeMarshaller(\_)),

impressionCallbacks = fullCover.impressionCallbacks.map(\_.map(callbackMarshaller(\_)))

))

}