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

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

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

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class PromotedMetadataMarshaller @Inject() (

disclosureTypeMarshaller: DisclosureTypeMarshaller,

adMetadataContainerMarshaller: AdMetadataContainerMarshaller,

clickTrackingInfoMarshaller: ClickTrackingInfoMarshaller) {

/\*\* See comments on [[com.twitter.product\_mixer.core.model.marshalling.response.urt.promoted.PromotedMetadata]]

\* regarding impressionId and impressionString

\*

\* TL;DR the domain model only has impressionString (the newer version) an this marshaller sets both

\* impressionId (the older) and impressionString based on it for compatibility.

\* \*/

def apply(promotedMetadata: PromotedMetadata): urt.PromotedMetadata =

urt.PromotedMetadata(

advertiserId = promotedMetadata.advertiserId,

impressionId = promotedMetadata.impressionString,

disclosureType = promotedMetadata.disclosureType.map(disclosureTypeMarshaller(\_)),

experimentValues = promotedMetadata.experimentValues,

promotedTrendId = promotedMetadata.promotedTrendId,

promotedTrendName = promotedMetadata.promotedTrendName,

promotedTrendQueryTerm = promotedMetadata.promotedTrendQueryTerm,

adMetadataContainer =

promotedMetadata.adMetadataContainer.map(adMetadataContainerMarshaller(\_)),

promotedTrendDescription = promotedMetadata.promotedTrendDescription,

impressionString = promotedMetadata.impressionString,

clickTrackingInfo = promotedMetadata.clickTrackingInfo.map(clickTrackingInfoMarshaller(\_))

)

}