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

import com.twitter.product\_mixer.core.functional\_component.marshaller.response.graphql.contextual\_ref.ContextualTweetRefMarshaller

import com.twitter.product\_mixer.core.functional\_component.marshaller.response.urt.item.conversation\_annotation.ConversationAnnotationMarshaller

import com.twitter.product\_mixer.core.functional\_component.marshaller.response.urt.item.forward\_pivot.ForwardPivotMarshaller

import com.twitter.product\_mixer.core.functional\_component.marshaller.response.urt.item.tombstone.TombstoneInfoMarshaller

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

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

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

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.tweet.TweetItem

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

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

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

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class TweetItemMarshaller @Inject() (

tweetDisplayTypeMarshaller: TweetDisplayTypeMarshaller,

socialContextMarshaller: SocialContextMarshaller,

tweetHighlightsMarshaller: TweetHighlightsMarshaller,

tombstoneInfoMarshaller: TombstoneInfoMarshaller,

timelinesScoreInfoMarshaller: TimelinesScoreInfoMarshaller,

forwardPivotMarshaller: ForwardPivotMarshaller,

promotedMetadataMarshaller: PromotedMetadataMarshaller,

conversationAnnotationMarshaller: ConversationAnnotationMarshaller,

contextualTweetRefMarshaller: ContextualTweetRefMarshaller,

prerollMetadataMarshaller: PrerollMetadataMarshaller,

badgeMarshaller: BadgeMarshaller,

urlMarshaller: UrlMarshaller) {

def apply(tweetItem: TweetItem): urt.TimelineItemContent.Tweet = urt.TimelineItemContent.Tweet(

urt.Tweet(

id = tweetItem.id,

displayType = tweetDisplayTypeMarshaller(tweetItem.displayType),

socialContext = tweetItem.socialContext.map(socialContextMarshaller(\_)),

highlights = tweetItem.highlights.map(tweetHighlightsMarshaller(\_)),

innerTombstoneInfo = tweetItem.innerTombstoneInfo.map(tombstoneInfoMarshaller(\_)),

timelinesScoreInfo = tweetItem.timelinesScoreInfo.map(timelinesScoreInfoMarshaller(\_)),

hasModeratedReplies = tweetItem.hasModeratedReplies,

forwardPivot = tweetItem.forwardPivot.map(forwardPivotMarshaller(\_)),

innerForwardPivot = tweetItem.innerForwardPivot.map(forwardPivotMarshaller(\_)),

promotedMetadata = tweetItem.promotedMetadata.map(promotedMetadataMarshaller(\_)),

conversationAnnotation =

tweetItem.conversationAnnotation.map(conversationAnnotationMarshaller(\_)),

contextualTweetRef = tweetItem.contextualTweetRef.map(contextualTweetRefMarshaller(\_)),

prerollMetadata = tweetItem.prerollMetadata.map(prerollMetadataMarshaller(\_)),

replyBadge = tweetItem.replyBadge.map(badgeMarshaller(\_)),

destination = tweetItem.destination.map(urlMarshaller(\_))

)

)

}