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

import com.twitter.product\_mixer.core.model.marshalling.response.slice.AdType

import com.twitter.product\_mixer.core.model.marshalling.response.slice

import com.twitter.strato.graphql.{thriftscala => t}

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class SliceItemMarshaller @Inject() () {

def apply(item: slice.SliceItem): t.SliceItem = {

item match {

case item: slice.TweetItem =>

t.SliceItem.TweetItem(t.TweetItem(id = item.id))

case item: slice.UserItem =>

t.SliceItem.UserItem(t.UserItem(id = item.id))

case item: slice.TwitterListItem =>

t.SliceItem.TwitterListItem(t.TwitterListItem(id = item.id))

case item: slice.DMConvoSearchItem =>

t.SliceItem.DmConvoSearchItem(t.DMConvoSearchItem(id = item.id))

case item: slice.DMConvoItem =>

t.SliceItem.DmConvoItem(t.DMConvoItem(id = item.id))

case item: slice.DMEventItem =>

t.SliceItem.DmEventItem(t.DMEventItem(id = item.id))

case item: slice.DMMessageSearchItem =>

t.SliceItem.DmMessageSearchItem(t.DMMessageSearchItem(id = item.id))

case item: slice.TopicItem =>

t.SliceItem.TopicItem(t.TopicItem(id = item.id.toString))

case item: slice.TypeaheadEventItem =>

t.SliceItem.TypeaheadEventItem(

t.TypeaheadEventItem(

eventId = item.eventId,

metadata = item.metadata.map(marshalTypeaheadMetadata)

)

)

case item: slice.TypeaheadQuerySuggestionItem =>

t.SliceItem.TypeaheadQuerySuggestionItem(

t.TypeaheadQuerySuggestionItem(

query = item.query,

metadata = item.metadata.map(marshalTypeaheadMetadata)

)

)

case item: slice.TypeaheadUserItem =>

t.SliceItem.TypeaheadUserItem(

t.TypeaheadUserItem(

userId = item.userId,

metadata = item.metadata.map(marshalTypeaheadMetadata),

badges = Some(item.badges.map { badge =>

t.UserBadge(

badgeUrl = badge.badgeUrl,

description = Some(badge.description),

badgeType = Some(badge.badgeType))

})

)

)

case item: slice.AdItem =>

t.SliceItem.AdItem(

t.AdItem(

adKey = t.AdKey(

adAccountId = item.adAccountId,

adUnitId = item.adUnitId,

)

)

)

case item: slice.AdCreativeItem =>

t.SliceItem.AdCreativeItem(

t.AdCreativeItem(

adCreativeKey = t.AdCreativeKey(

adAccountId = item.adAccountId,

adType = marshalAdType(item.adType),

creativeId = item.creativeId

)

)

)

case item: slice.AdGroupItem =>

t.SliceItem.AdGroupItem(

t.AdGroupItem(

adGroupKey = t.AdGroupKey(

adAccountId = item.adAccountId,

adGroupId = item.adGroupId

)

)

)

case item: slice.CampaignItem =>

t.SliceItem.CampaignItem(

t.CampaignItem(

campaignKey = t.CampaignKey(

adAccountId = item.adAccountId,

campaignId = item.campaignId

)

)

)

case item: slice.FundingSourceItem =>

t.SliceItem.FundingSourceItem(

t.FundingSourceItem(

fundingSourceKey = t.FundingSourceKey(

adAccountId = item.adAccountId,

fundingSourceId = item.fundingSourceId

)

)

)

}

}

private def marshalTypeaheadMetadata(metadata: slice.TypeaheadMetadata) = {

t.TypeaheadMetadata(

score = metadata.score,

source = metadata.source,

resultContext = metadata.context.map(context =>

t.TypeaheadResultContext(

displayString = context.displayString,

contextType = marshalRequestContextType(context.contextType),

iconUrl = context.iconUrl

))

)

}

private def marshalRequestContextType(

context: slice.TypeaheadResultContextType

): t.TypeaheadResultContextType = {

context match {

case slice.You => t.TypeaheadResultContextType.You

case slice.Location => t.TypeaheadResultContextType.Location

case slice.NumFollowers => t.TypeaheadResultContextType.NumFollowers

case slice.FollowRelationship => t.TypeaheadResultContextType.FollowRelationship

case slice.Bio => t.TypeaheadResultContextType.Bio

case slice.NumTweets => t.TypeaheadResultContextType.NumTweets

case slice.Trending => t.TypeaheadResultContextType.Trending

case slice.HighlightedLabel => t.TypeaheadResultContextType.HighlightedLabel

case \_ => t.TypeaheadResultContextType.Undefined

}

}

private def marshalAdType(

adType: AdType

): t.AdType = {

adType match {

case AdType.Tweet => t.AdType.Tweet

case AdType.Account => t.AdType.Account

case AdType.InStreamVideo => t.AdType.InStreamVideo

case AdType.DisplayCreative => t.AdType.DisplayCreative

case AdType.Trend => t.AdType.Trend

case AdType.Spotlight => t.AdType.Spotlight

case AdType.Takeover => t.AdType.Takeover

}

}

}