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

import com.twitter.product\_mixer.core.model.marshalling.HasMarshalling

/\*\*

\* These are the Ad Types exposed on AdUnits

\*

\* They are to be kept in sync with strato/config/src/thrift/com/twitter/strato/graphql/hubble.thrift

\*/

sealed trait AdType

object AdType {

case object Tweet extends AdType

case object Account extends AdType

case object InStreamVideo extends AdType

case object DisplayCreative extends AdType

case object Trend extends AdType

case object Spotlight extends AdType

case object Takeover extends AdType

}

trait SliceItem

case class TweetItem(id: Long) extends SliceItem

case class UserItem(id: Long) extends SliceItem

case class TwitterListItem(id: Long) extends SliceItem

case class DMConvoSearchItem(id: String, lastReadableEventId: Option[Long]) extends SliceItem

case class DMEventItem(id: Long) extends SliceItem

case class DMConvoItem(id: String, lastReadableEventId: Option[Long]) extends SliceItem

case class DMMessageSearchItem(id: Long) extends SliceItem

case class TopicItem(id: Long) extends SliceItem

case class TypeaheadEventItem(eventId: Long, metadata: Option[TypeaheadMetadata]) extends SliceItem

case class TypeaheadQuerySuggestionItem(query: String, metadata: Option[TypeaheadMetadata])

extends SliceItem

case class TypeaheadUserItem(

userId: Long,

metadata: Option[TypeaheadMetadata],

badges: Seq[UserBadge])

extends SliceItem

case class AdItem(adUnitId: Long, adAccountId: Long) extends SliceItem

case class AdCreativeItem(creativeId: Long, adType: AdType, adAccountId: Long) extends SliceItem

case class AdGroupItem(adGroupId: Long, adAccountId: Long) extends SliceItem

case class CampaignItem(campaignId: Long, adAccountId: Long) extends SliceItem

case class FundingSourceItem(fundingSourceId: Long, adAccountId: Long) extends SliceItem

sealed trait CursorType

case object PreviousCursor extends CursorType

case object NextCursor extends CursorType

@deprecated(

"GapCursors are not supported by Product Mixer Slice marshallers, if you need support for these reach out to #product-mixer")

case object GapCursor extends CursorType

// CursorItem extends SliceItem to enable support for GapCursors

case class CursorItem(value: String, cursorType: CursorType) extends SliceItem

case class SliceInfo(

previousCursor: Option[String],

nextCursor: Option[String])

case class Slice(

items: Seq[SliceItem],

sliceInfo: SliceInfo)

extends HasMarshalling

sealed trait TypeaheadResultContextType

case object You extends TypeaheadResultContextType

case object Location extends TypeaheadResultContextType

case object NumFollowers extends TypeaheadResultContextType

case object FollowRelationship extends TypeaheadResultContextType

case object Bio extends TypeaheadResultContextType

case object NumTweets extends TypeaheadResultContextType

case object Trending extends TypeaheadResultContextType

case object HighlightedLabel extends TypeaheadResultContextType

case class TypeaheadResultContext(

contextType: TypeaheadResultContextType,

displayString: String,

iconUrl: Option[String])

case class TypeaheadMetadata(

score: Double,

source: Option[String],

context: Option[TypeaheadResultContext])

// Used to render badges in Typeahead, such as Business-affiliated badges

case class UserBadge(badgeType: String, badgeUrl: String, description: String)