package com.twitter.product\_mixer.component\_library.premarshaller.slice

import com.twitter.product\_mixer.component\_library.model.candidate.\_

import com.twitter.product\_mixer.component\_library.model.candidate.hubble.AdCreativeCandidate

import com.twitter.product\_mixer.component\_library.model.candidate.hubble.AdGroupCandidate

import com.twitter.product\_mixer.component\_library.model.candidate.hubble.AdUnitCandidate

import com.twitter.product\_mixer.component\_library.model.candidate.hubble.CampaignCandidate

import com.twitter.product\_mixer.component\_library.model.candidate.hubble.FundingSourceCandidate

import com.twitter.product\_mixer.component\_library.model.candidate.suggestion.QuerySuggestionCandidate

import com.twitter.product\_mixer.component\_library.model.candidate.suggestion.TypeaheadEventCandidate

import com.twitter.product\_mixer.component\_library.premarshaller.slice.builder.SliceBuilder

import com.twitter.product\_mixer.component\_library.premarshaller.slice.builder.SliceCursorBuilder

import com.twitter.product\_mixer.component\_library.premarshaller.slice.builder.SliceCursorUpdater

import com.twitter.product\_mixer.core.functional\_component.premarshaller.DomainMarshaller

import com.twitter.product\_mixer.core.functional\_component.premarshaller.UndecoratedCandidateDomainMarshallerException

import com.twitter.product\_mixer.core.functional\_component.premarshaller.UnsupportedCandidateDomainMarshallerException

import com.twitter.product\_mixer.core.functional\_component.premarshaller.UnsupportedModuleDomainMarshallerException

import com.twitter.product\_mixer.core.functional\_component.premarshaller.UnsupportedPresentationDomainMarshallerException

import com.twitter.product\_mixer.core.model.common.identifier.DomainMarshallerIdentifier

import com.twitter.product\_mixer.core.model.common.presentation.CandidateWithDetails

import com.twitter.product\_mixer.core.model.common.presentation.ItemCandidateWithDetails

import com.twitter.product\_mixer.core.model.common.presentation.ModuleCandidateWithDetails

import com.twitter.product\_mixer.core.model.common.presentation.slice.BaseSliceItemPresentation

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

import com.twitter.product\_mixer.core.pipeline.PipelineQuery

/\*\*

\* Domain marshaller that generates Slices automatically for most candidates but a different

\* presentation can be provided by decorators that implement [[BaseSliceItemPresentation]]. This will

\* only be necessary in the rare case that a candidate contains more than an id. For example,

\* cursors require a value/type rather than an id.

\*/

case class SliceDomainMarshaller[-Query <: PipelineQuery](

override val cursorBuilders: Seq[SliceCursorBuilder[Query]] = Seq.empty,

override val cursorUpdaters: Seq[SliceCursorUpdater[Query]] = Seq.empty,

override val identifier: DomainMarshallerIdentifier = DomainMarshallerIdentifier("Slice"))

extends DomainMarshaller[Query, Slice]

with SliceBuilder[Query] {

override def apply(

query: Query,

selections: Seq[CandidateWithDetails]

): Slice = {

val entries = selections.map {

case ItemCandidateWithDetails(\_, Some(presentation: BaseSliceItemPresentation), \_) =>

presentation.sliceItem

case candidateWithDetails @ ItemCandidateWithDetails(candidate, None, \_) =>

val source = candidateWithDetails.source

candidate match {

case candidate: BaseTopicCandidate => TopicItem(candidate.id)

case candidate: BaseTweetCandidate => TweetItem(candidate.id)

case candidate: BaseUserCandidate => UserItem(candidate.id)

case candidate: TwitterListCandidate => TwitterListItem(candidate.id)

case candidate: DMConvoSearchCandidate =>

DMConvoSearchItem(candidate.id, candidate.lastReadableEventId)

case candidate: DMEventCandidate =>

DMEventItem(candidate.id)

case candidate: DMConvoCandidate =>

DMConvoItem(candidate.id, candidate.lastReadableEventId)

case candidate: DMMessageSearchCandidate => DMMessageSearchItem(candidate.id)

case candidate: QuerySuggestionCandidate =>

TypeaheadQuerySuggestionItem(candidate.id, candidate.metadata)

case candidate: TypeaheadEventCandidate =>

TypeaheadEventItem(candidate.id, candidate.metadata)

case candidate: AdUnitCandidate =>

AdItem(candidate.id, candidate.adAccountId)

case candidate: AdCreativeCandidate =>

AdCreativeItem(candidate.id, candidate.adType, candidate.adAccountId)

case candidate: AdGroupCandidate =>

AdGroupItem(candidate.id, candidate.adAccountId)

case candidate: CampaignCandidate =>

CampaignItem(candidate.id, candidate.adAccountId)

case candidate: FundingSourceCandidate =>

FundingSourceItem(candidate.id, candidate.adAccountId)

case candidate: CursorCandidate =>

// Cursors must contain a cursor type which is defined by the presentation. As a result,

// cursors are expected to be handled by the Some(presentation) case above, and must not

// fall into this case.

throw new UndecoratedCandidateDomainMarshallerException(candidate, source)

case candidate =>

throw new UnsupportedCandidateDomainMarshallerException(candidate, source)

}

case itemCandidateWithDetails @ ItemCandidateWithDetails(candidate, Some(presentation), \_) =>

throw new UnsupportedPresentationDomainMarshallerException(

candidate,

presentation,

itemCandidateWithDetails.source)

case moduleCandidateWithDetails @ ModuleCandidateWithDetails(\_, presentation, \_) =>

throw new UnsupportedModuleDomainMarshallerException(

presentation,

moduleCandidateWithDetails.source)

}

buildSlice(query, entries)

}

}