package com.twitter.home\_mixer.product.for\_you

import com.twitter.clientapp.{thriftscala => ca}

import com.twitter.goldfinch.api.AdsInjectionSurfaceAreas

import com.twitter.home\_mixer.candidate\_pipeline.EditedTweetsCandidatePipelineConfig

import com.twitter.home\_mixer.candidate\_pipeline.NewTweetsPillCandidatePipelineConfig

import com.twitter.home\_mixer.functional\_component.decorator.urt.builder.AddEntriesWithReplaceAndShowAlertAndCoverInstructionBuilder

import com.twitter.home\_mixer.functional\_component.feature\_hydrator.FeedbackHistoryQueryFeatureHydrator

import com.twitter.home\_mixer.functional\_component.feature\_hydrator.\_

import com.twitter.home\_mixer.functional\_component.selector.DebunchCandidates

import com.twitter.home\_mixer.functional\_component.selector.UpdateConversationModuleId

import com.twitter.home\_mixer.functional\_component.selector.UpdateHomeClientEventDetails

import com.twitter.home\_mixer.functional\_component.selector.UpdateNewTweetsPillDecoration

import com.twitter.home\_mixer.functional\_component.side\_effect.\_

import com.twitter.home\_mixer.model.ClearCacheIncludeInstruction

import com.twitter.home\_mixer.model.HomeFeatures.InNetworkFeature

import com.twitter.home\_mixer.param.HomeGlobalParams.EnableImpressionBloomFilter

import com.twitter.home\_mixer.param.HomeGlobalParams.MaxNumberReplaceInstructionsParam

import com.twitter.home\_mixer.param.HomeMixerFlagName.ScribeClientEventsFlag

import com.twitter.home\_mixer.product.following.model.HomeMixerExternalStrings

import com.twitter.home\_mixer.product.for\_you.feature\_hydrator.TimelineServiceTweetsQueryFeatureHydrator

import com.twitter.home\_mixer.product.for\_you.model.ForYouQuery

import com.twitter.home\_mixer.product.for\_you.param.ForYouParam.ClearCacheOnPtr

import com.twitter.home\_mixer.product.for\_you.param.ForYouParam.EnableFlipInjectionModuleCandidatePipelineParam

import com.twitter.home\_mixer.product.for\_you.param.ForYouParam.FlipInlineInjectionModulePosition

import com.twitter.home\_mixer.product.for\_you.param.ForYouParam.ServerMaxResultsParam

import com.twitter.home\_mixer.product.for\_you.param.ForYouParam.TweetPreviewsPositionParam

import com.twitter.home\_mixer.product.for\_you.param.ForYouParam.WhoToFollowPositionParam

import com.twitter.home\_mixer.product.for\_you.param.ForYouParam.WhoToSubscribePositionParam

import com.twitter.home\_mixer.product.for\_you.side\_effect.ServedCandidateFeatureKeysKafkaSideEffectBuilder

import com.twitter.home\_mixer.product.for\_you.side\_effect.ServedCandidateKeysKafkaSideEffectBuilder

import com.twitter.home\_mixer.product.for\_you.side\_effect.ServedStatsSideEffect

import com.twitter.home\_mixer.util.CandidatesUtil

import com.twitter.inject.annotations.Flag

import com.twitter.logpipeline.client.common.EventPublisher

import com.twitter.product\_mixer.component\_library.feature\_hydrator.query.async.AsyncQueryFeatureHydrator

import com.twitter.product\_mixer.component\_library.feature\_hydrator.query.impressed\_tweets.ImpressedTweetsQueryFeatureHydrator

import com.twitter.product\_mixer.component\_library.feature\_hydrator.query.param\_gated.AsyncParamGatedQueryFeatureHydrator

import com.twitter.product\_mixer.component\_library.feature\_hydrator.query.social\_graph.PreviewCreatorsQueryFeatureHydrator

import com.twitter.product\_mixer.component\_library.feature\_hydrator.query.social\_graph.SGSFollowedUsersQueryFeatureHydrator

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

import com.twitter.product\_mixer.component\_library.pipeline.candidate.flexible\_injection\_pipeline.FlipPromptCandidatePipelineConfigBuilder

import com.twitter.product\_mixer.component\_library.pipeline.candidate.who\_to\_follow\_module.WhoToFollowArmCandidatePipelineConfig

import com.twitter.product\_mixer.component\_library.pipeline.candidate.who\_to\_subscribe\_module.WhoToSubscribeCandidatePipelineConfig

import com.twitter.product\_mixer.component\_library.premarshaller.urt.UrtDomainMarshaller

import com.twitter.product\_mixer.component\_library.premarshaller.urt.builder.ClearCacheInstructionBuilder

import com.twitter.product\_mixer.component\_library.premarshaller.urt.builder.OrderedBottomCursorBuilder

import com.twitter.product\_mixer.component\_library.premarshaller.urt.builder.OrderedTopCursorBuilder

import com.twitter.product\_mixer.component\_library.premarshaller.urt.builder.ReplaceAllEntries

import com.twitter.product\_mixer.component\_library.premarshaller.urt.builder.ReplaceEntryInstructionBuilder

import com.twitter.product\_mixer.component\_library.premarshaller.urt.builder.ShowAlertInstructionBuilder

import com.twitter.product\_mixer.component\_library.premarshaller.urt.builder.ShowCoverInstructionBuilder

import com.twitter.product\_mixer.component\_library.premarshaller.urt.builder.StaticTimelineScribeConfigBuilder

import com.twitter.product\_mixer.component\_library.premarshaller.urt.builder.UrtMetadataBuilder

import com.twitter.product\_mixer.component\_library.selector.DropMaxCandidates

import com.twitter.product\_mixer.component\_library.selector.DropMaxModuleItemCandidates

import com.twitter.product\_mixer.component\_library.selector.DropModuleTooFewModuleItemResults

import com.twitter.product\_mixer.component\_library.selector.InsertAppendResults

import com.twitter.product\_mixer.component\_library.selector.InsertFixedPositionResults

import com.twitter.product\_mixer.component\_library.selector.SelectConditionally

import com.twitter.product\_mixer.component\_library.selector.UpdateSortCandidates

import com.twitter.product\_mixer.component\_library.selector.UpdateSortModuleItemCandidates

import com.twitter.product\_mixer.component\_library.selector.ads.AdsInjector

import com.twitter.product\_mixer.component\_library.selector.ads.InsertAdResults

import com.twitter.product\_mixer.core.functional\_component.common.SpecificPipeline

import com.twitter.product\_mixer.core.functional\_component.common.SpecificPipelines

import com.twitter.product\_mixer.core.functional\_component.configapi.StaticParam

import com.twitter.product\_mixer.core.functional\_component.feature\_hydrator.QueryFeatureHydrator

import com.twitter.product\_mixer.core.functional\_component.marshaller.TransportMarshaller

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

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

import com.twitter.product\_mixer.core.functional\_component.selector.Selector

import com.twitter.product\_mixer.core.functional\_component.side\_effect.PipelineResultSideEffect

import com.twitter.product\_mixer.core.model.common.UniversalNoun

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

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

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.marshalling.response.urt.Timeline

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

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

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

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

import com.twitter.product\_mixer.core.pipeline.candidate.CandidatePipelineConfig

import com.twitter.product\_mixer.core.pipeline.candidate.DependentCandidatePipelineConfig

import com.twitter.product\_mixer.core.pipeline.mixer.MixerPipelineConfig

import com.twitter.product\_mixer.core.product.guice.scope.ProductScoped

import com.twitter.stringcenter.client.StringCenter

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

import javax.inject.Inject

import javax.inject.Provider

import javax.inject.Singleton

@Singleton

class ForYouTimelineScorerMixerPipelineConfig @Inject() (

forYouTimelineScorerCandidatePipelineConfig: ForYouTimelineScorerCandidatePipelineConfig,

forYouPushToHomeTweetCandidatePipelineConfig: ForYouPushToHomeTweetCandidatePipelineConfig,

forYouConversationServiceCandidatePipelineConfig: ForYouConversationServiceCandidatePipelineConfig,

forYouAdsDependentCandidatePipelineBuilder: ForYouAdsDependentCandidatePipelineBuilder,

forYouWhoToFollowCandidatePipelineConfigBuilder: ForYouWhoToFollowCandidatePipelineConfigBuilder,

forYouWhoToSubscribeCandidatePipelineConfigBuilder: ForYouWhoToSubscribeCandidatePipelineConfigBuilder,

flipPromptCandidatePipelineConfigBuilder: FlipPromptCandidatePipelineConfigBuilder,

editedTweetsCandidatePipelineConfig: EditedTweetsCandidatePipelineConfig,

newTweetsPillCandidatePipelineConfig: NewTweetsPillCandidatePipelineConfig[ForYouQuery],

forYouTweetPreviewsCandidatePipelineConfig: ForYouTweetPreviewsCandidatePipelineConfig,

dismissInfoQueryFeatureHydrator: DismissInfoQueryFeatureHydrator,

gizmoduckUserQueryFeatureHydrator: GizmoduckUserQueryFeatureHydrator,

impressionBloomFilterQueryFeatureHydrator: ImpressionBloomFilterQueryFeatureHydrator[ForYouQuery],

manhattanTweetImpressionsQueryFeatureHydrator: TweetImpressionsQueryFeatureHydrator[ForYouQuery],

memcacheTweetImpressionsQueryFeatureHydrator: ImpressedTweetsQueryFeatureHydrator,

persistenceStoreQueryFeatureHydrator: PersistenceStoreQueryFeatureHydrator,

requestQueryFeatureHydrator: RequestQueryFeatureHydrator[ForYouQuery],

timelineServiceTweetsQueryFeatureHydrator: TimelineServiceTweetsQueryFeatureHydrator,

lastNonPollingTimeQueryFeatureHydrator: LastNonPollingTimeQueryFeatureHydrator,

feedbackHistoryQueryFeatureHydrator: FeedbackHistoryQueryFeatureHydrator,

previewCreatorsQueryFeatureHydrator: PreviewCreatorsQueryFeatureHydrator,

sgsFollowedUsersQueryFeatureHydrator: SGSFollowedUsersQueryFeatureHydrator,

adsInjector: AdsInjector,

servedCandidateKeysKafkaSideEffectBuilder: ServedCandidateKeysKafkaSideEffectBuilder,

servedCandidateFeatureKeysKafkaSideEffectBuilder: ServedCandidateFeatureKeysKafkaSideEffectBuilder,

updateLastNonPollingTimeSideEffect: UpdateLastNonPollingTimeSideEffect[ForYouQuery, Timeline],

publishClientSentImpressionsEventBusSideEffect: PublishClientSentImpressionsEventBusSideEffect,

publishClientSentImpressionsManhattanSideEffect: PublishClientSentImpressionsManhattanSideEffect,

publishImpressionBloomFilterSideEffect: PublishImpressionBloomFilterSideEffect,

updateTimelinesPersistenceStoreSideEffect: UpdateTimelinesPersistenceStoreSideEffect,

truncateTimelinesPersistenceStoreSideEffect: TruncateTimelinesPersistenceStoreSideEffect,

homeScribeServedCandidatesSideEffect: HomeScribeServedCandidatesSideEffect,

servedStatsSideEffect: ServedStatsSideEffect,

clientEventsScribeEventPublisher: EventPublisher[ca.LogEvent],

externalStrings: HomeMixerExternalStrings,

@ProductScoped stringCenterProvider: Provider[StringCenter],

urtTransportMarshaller: UrtTransportMarshaller,

@Flag(ScribeClientEventsFlag) enableScribeClientEvents: Boolean)

extends MixerPipelineConfig[ForYouQuery, Timeline, urt.TimelineResponse] {

override val identifier: MixerPipelineIdentifier = MixerPipelineIdentifier("ForYouTimelineScorer")

private val MaxConsecutiveOutOfNetworkCandidates = 2

private val PushToHomeTweetPosition = 0

private val dependentCandidatesStep = MixerPipelineConfig.dependentCandidatePipelinesStep

private val resultSelectorsStep = MixerPipelineConfig.resultSelectorsStep

override def fetchQueryFeatures: Seq[QueryFeatureHydrator[ForYouQuery]] = Seq(

requestQueryFeatureHydrator,

persistenceStoreQueryFeatureHydrator,

timelineServiceTweetsQueryFeatureHydrator,

feedbackHistoryQueryFeatureHydrator,

previewCreatorsQueryFeatureHydrator,

sgsFollowedUsersQueryFeatureHydrator,

AsyncQueryFeatureHydrator(dependentCandidatesStep, dismissInfoQueryFeatureHydrator),

AsyncQueryFeatureHydrator(dependentCandidatesStep, gizmoduckUserQueryFeatureHydrator),

AsyncQueryFeatureHydrator(dependentCandidatesStep, lastNonPollingTimeQueryFeatureHydrator),

AsyncParamGatedQueryFeatureHydrator(

EnableImpressionBloomFilter,

resultSelectorsStep,

impressionBloomFilterQueryFeatureHydrator),

AsyncQueryFeatureHydrator(resultSelectorsStep, manhattanTweetImpressionsQueryFeatureHydrator),

AsyncQueryFeatureHydrator(resultSelectorsStep, memcacheTweetImpressionsQueryFeatureHydrator)

)

private val timelineScorerCandidatePipelineScope =

SpecificPipeline(forYouTimelineScorerCandidatePipelineConfig.identifier)

private val forYouAdsCandidatePipelineConfig = forYouAdsDependentCandidatePipelineBuilder

.build(timelineScorerCandidatePipelineScope)

private val forYouWhoToFollowCandidatePipelineConfig =

forYouWhoToFollowCandidatePipelineConfigBuilder.build()

private val forYouWhoToSubscribeCandidatePipelineConfig =

forYouWhoToSubscribeCandidatePipelineConfigBuilder.build()

private val flipPromptCandidatePipelineConfig =

flipPromptCandidatePipelineConfigBuilder.build[ForYouQuery](

supportedClientParam = Some(EnableFlipInjectionModuleCandidatePipelineParam)

)

override val candidatePipelines: Seq[CandidatePipelineConfig[ForYouQuery, \_, \_, \_]] = Seq(

forYouTimelineScorerCandidatePipelineConfig,

forYouPushToHomeTweetCandidatePipelineConfig,

forYouWhoToFollowCandidatePipelineConfig,

forYouWhoToSubscribeCandidatePipelineConfig,

forYouTweetPreviewsCandidatePipelineConfig,

flipPromptCandidatePipelineConfig

)

override val dependentCandidatePipelines: Seq[

DependentCandidatePipelineConfig[ForYouQuery, \_, \_, \_]

] = Seq(

forYouAdsCandidatePipelineConfig,

forYouConversationServiceCandidatePipelineConfig,

editedTweetsCandidatePipelineConfig,

newTweetsPillCandidatePipelineConfig

)

override val failOpenPolicies: Map[CandidatePipelineIdentifier, FailOpenPolicy] = Map(

forYouTimelineScorerCandidatePipelineConfig.identifier -> FailOpenPolicy.Always,

forYouAdsCandidatePipelineConfig.identifier -> FailOpenPolicy.Always,

forYouWhoToFollowCandidatePipelineConfig.identifier -> FailOpenPolicy.Always,

forYouWhoToSubscribeCandidatePipelineConfig.identifier -> FailOpenPolicy.Always,

forYouTweetPreviewsCandidatePipelineConfig.identifier -> FailOpenPolicy.Always,

flipPromptCandidatePipelineConfig.identifier -> FailOpenPolicy.Always,

editedTweetsCandidatePipelineConfig.identifier -> FailOpenPolicy.Always,

newTweetsPillCandidatePipelineConfig.identifier -> FailOpenPolicy.Always,

)

override val resultSelectors: Seq[Selector[ForYouQuery]] = Seq(

UpdateSortCandidates(

ordering = CandidatesUtil.reverseChronTweetsOrdering,

candidatePipeline = forYouConversationServiceCandidatePipelineConfig.identifier

),

UpdateSortCandidates(

ordering = CandidatesUtil.scoreOrdering,

candidatePipeline = forYouTimelineScorerCandidatePipelineConfig.identifier

),

UpdateSortModuleItemCandidates(

candidatePipeline = forYouTimelineScorerCandidatePipelineConfig.identifier,

ordering = CandidatesUtil.conversationModuleTweetsOrdering

),

DebunchCandidates(

pipelineScope = SpecificPipeline(forYouTimelineScorerCandidatePipelineConfig.identifier),

mustDebunch = {

case item: ItemCandidateWithDetails =>

!item.features.getOrElse(InNetworkFeature, false)

case module: ModuleCandidateWithDetails =>

!module.candidates.last.features.getOrElse(InNetworkFeature, false)

},

maxBunchSize = MaxConsecutiveOutOfNetworkCandidates

),

UpdateConversationModuleId(

pipelineScope = SpecificPipeline(forYouTimelineScorerCandidatePipelineConfig.identifier)

),

DropMaxCandidates(

candidatePipeline = forYouConversationServiceCandidatePipelineConfig.identifier,

maxSelectionsParam = ServerMaxResultsParam

),

DropMaxCandidates(

candidatePipeline = forYouTimelineScorerCandidatePipelineConfig.identifier,

maxSelectionsParam = ServerMaxResultsParam

),

DropMaxCandidates(

candidatePipeline = editedTweetsCandidatePipelineConfig.identifier,

maxSelectionsParam = MaxNumberReplaceInstructionsParam

),

DropMaxModuleItemCandidates(

candidatePipeline = forYouWhoToFollowCandidatePipelineConfig.identifier,

maxModuleItemsParam = StaticParam(WhoToFollowArmCandidatePipelineConfig.MaxCandidatesSize)

),

DropModuleTooFewModuleItemResults(

candidatePipeline = forYouWhoToSubscribeCandidatePipelineConfig.identifier,

minModuleItemsParam = StaticParam(WhoToSubscribeCandidatePipelineConfig.MinCandidatesSize)

),

DropMaxModuleItemCandidates(

candidatePipeline = forYouWhoToSubscribeCandidatePipelineConfig.identifier,

maxModuleItemsParam = StaticParam(WhoToSubscribeCandidatePipelineConfig.MaxCandidatesSize)

),

// Add Conversation Service tweets to results only if the scored pipeline doesn't return any

SelectConditionally(

selector = InsertAppendResults(

candidatePipeline = forYouConversationServiceCandidatePipelineConfig.identifier),

includeSelector = (\_, candidates, \_) =>

!candidates.exists(candidate =>

forYouTimelineScorerCandidatePipelineConfig.identifier == candidate.source)

),

InsertAppendResults(candidatePipeline = forYouTimelineScorerCandidatePipelineConfig.identifier),

InsertFixedPositionResults(

candidatePipeline = forYouTweetPreviewsCandidatePipelineConfig.identifier,

positionParam = TweetPreviewsPositionParam

),

InsertFixedPositionResults(

candidatePipeline = forYouWhoToFollowCandidatePipelineConfig.identifier,

positionParam = WhoToFollowPositionParam

),

InsertFixedPositionResults(

candidatePipeline = forYouWhoToSubscribeCandidatePipelineConfig.identifier,

positionParam = WhoToSubscribePositionParam

),

InsertFixedPositionResults(

candidatePipeline = flipPromptCandidatePipelineConfig.identifier,

positionParam = FlipInlineInjectionModulePosition

),

// Insert Push To Home Tweet at top of Timeline

InsertFixedPositionResults(

candidatePipeline = forYouPushToHomeTweetCandidatePipelineConfig.identifier,

positionParam = StaticParam(PushToHomeTweetPosition)

),

InsertAdResults(

surfaceAreaName = AdsInjectionSurfaceAreas.HomeTimeline,

adsInjector = adsInjector.forSurfaceArea(AdsInjectionSurfaceAreas.HomeTimeline),

adsCandidatePipeline = forYouAdsCandidatePipelineConfig.identifier

),

// This selector must come after the tweets are inserted into the results

DropModuleTooFewModuleItemResults(

candidatePipeline = forYouWhoToFollowCandidatePipelineConfig.identifier,

minModuleItemsParam = StaticParam(WhoToFollowArmCandidatePipelineConfig.MinCandidatesSize)

),

UpdateNewTweetsPillDecoration(

pipelineScope = SpecificPipelines(

forYouConversationServiceCandidatePipelineConfig.identifier,

forYouTimelineScorerCandidatePipelineConfig.identifier,

newTweetsPillCandidatePipelineConfig.identifier

),

stringCenter = stringCenterProvider.get(),

seeNewTweetsString = externalStrings.seeNewTweetsString,

tweetedString = externalStrings.tweetedString

),

InsertAppendResults(candidatePipeline = editedTweetsCandidatePipelineConfig.identifier),

SelectConditionally(

selector =

InsertAppendResults(candidatePipeline = newTweetsPillCandidatePipelineConfig.identifier),

includeSelector = (\_, \_, results) => CandidatesUtil.containsType[TweetCandidate](results)

),

UpdateHomeClientEventDetails(

candidatePipelines = Set(

forYouConversationServiceCandidatePipelineConfig.identifier,

forYouTimelineScorerCandidatePipelineConfig.identifier

)

),

)

private val servedCandidateKeysKafkaSideEffect =

servedCandidateKeysKafkaSideEffectBuilder.build(

Set(forYouTimelineScorerCandidatePipelineConfig.identifier))

private val servedCandidateFeatureKeysKafkaSideEffect =

servedCandidateFeatureKeysKafkaSideEffectBuilder.build(

Set(forYouTimelineScorerCandidatePipelineConfig.identifier))

private val homeScribeClientEventSideEffect = HomeScribeClientEventSideEffect(

enableScribeClientEvents = enableScribeClientEvents,

logPipelinePublisher = clientEventsScribeEventPublisher,

injectedTweetsCandidatePipelineIdentifiers = Seq(

forYouTimelineScorerCandidatePipelineConfig.identifier,

forYouConversationServiceCandidatePipelineConfig.identifier

),

adsCandidatePipelineIdentifier = Some(forYouAdsCandidatePipelineConfig.identifier),

whoToFollowCandidatePipelineIdentifier =

Some(forYouWhoToFollowCandidatePipelineConfig.identifier),

whoToSubscribeCandidatePipelineIdentifier =

Some(forYouWhoToSubscribeCandidatePipelineConfig.identifier)

)

override val resultSideEffects: Seq[PipelineResultSideEffect[ForYouQuery, Timeline]] = Seq(

homeScribeClientEventSideEffect,

homeScribeServedCandidatesSideEffect,

publishClientSentImpressionsEventBusSideEffect,

publishClientSentImpressionsManhattanSideEffect,

publishImpressionBloomFilterSideEffect,

servedCandidateKeysKafkaSideEffect,

servedCandidateFeatureKeysKafkaSideEffect,

servedStatsSideEffect,

truncateTimelinesPersistenceStoreSideEffect,

updateLastNonPollingTimeSideEffect,

updateTimelinesPersistenceStoreSideEffect

)

override val domainMarshaller: DomainMarshaller[ForYouQuery, Timeline] = {

val instructionBuilders = Seq(

ClearCacheInstructionBuilder(

ClearCacheIncludeInstruction(

ClearCacheOnPtr.EnableParam,

ClearCacheOnPtr.MinEntriesParam

)

),

ReplaceEntryInstructionBuilder(ReplaceAllEntries),

// excludes alert, cover, and replace candidates

AddEntriesWithReplaceAndShowAlertAndCoverInstructionBuilder(),

ShowAlertInstructionBuilder(),

ShowCoverInstructionBuilder(),

)

val idSelector: PartialFunction[UniversalNoun[\_], Long] = {

// exclude ads while determining tweet cursor values

case item: TweetItem if item.promotedMetadata.isEmpty => item.id

case module: TimelineModule

if module.items.headOption.exists(\_.item.isInstanceOf[TweetItem]) =>

module.items.last.item match { case item: TweetItem => item.id }

}

val topCursorBuilder = OrderedTopCursorBuilder(idSelector)

val bottomCursorBuilder = OrderedBottomCursorBuilder(idSelector)

val metadataBuilder = UrtMetadataBuilder(

title = None,

scribeConfigBuilder = Some(

StaticTimelineScribeConfigBuilder(

TimelineScribeConfig(

page = Some("for\_you\_timeline\_scorer"),

section = None,

entityToken = None)))

)

UrtDomainMarshaller(

instructionBuilders = instructionBuilders,

metadataBuilder = Some(metadataBuilder),

cursorBuilders = Seq(topCursorBuilder, bottomCursorBuilder)

)

}

override val transportMarshaller: TransportMarshaller[Timeline, urt.TimelineResponse] =

urtTransportMarshaller

}