package com.twitter.home\_mixer.candidate\_pipeline

import com.twitter.conversions.DurationOps.\_

import com.twitter.home\_mixer.functional\_component.gate.RequestContextNotGate

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

import com.twitter.home\_mixer.model.request.DeviceContext

import com.twitter.home\_mixer.model.request.HasDeviceContext

import com.twitter.home\_mixer.service.HomeMixerAlertConfig

import com.twitter.product\_mixer.component\_library.decorator.urt.UrtItemCandidateDecorator

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.alert.DurationParamBuilder

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.alert.ShowAlertCandidateUrtItemBuilder

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.alert.StaticShowAlertColorConfigurationBuilder

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.alert.StaticShowAlertDisplayLocationBuilder

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.alert.StaticShowAlertIconDisplayInfoBuilder

import com.twitter.product\_mixer.component\_library.gate.FeatureGate

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

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMap

import com.twitter.product\_mixer.core.functional\_component.candidate\_source.CandidateSource

import com.twitter.product\_mixer.core.functional\_component.candidate\_source.StaticCandidateSource

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

import com.twitter.product\_mixer.core.functional\_component.decorator.CandidateDecorator

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.item.alert.BaseDurationBuilder

import com.twitter.product\_mixer.core.functional\_component.gate.Gate

import com.twitter.product\_mixer.core.functional\_component.transformer.CandidatePipelineQueryTransformer

import com.twitter.product\_mixer.core.functional\_component.transformer.CandidatePipelineResultsTransformer

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

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

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

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

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

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

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

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

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

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

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

import com.twitter.util.Duration

import javax.inject.Inject

import javax.inject.Singleton

/\*\*

\* Candidate Pipeline Config that creates the New Tweets Pill

\*/

@Singleton

class NewTweetsPillCandidatePipelineConfig[Query <: PipelineQuery with HasDeviceContext] @Inject() (

) extends DependentCandidatePipelineConfig[

Query,

Unit,

ShowAlertCandidate,

ShowAlertCandidate

] {

import NewTweetsPillCandidatePipelineConfig.\_

override val identifier: CandidatePipelineIdentifier =

CandidatePipelineIdentifier("NewTweetsPill")

override val gates: Seq[Gate[Query]] = Seq(

RequestContextNotGate(Seq(DeviceContext.RequestContext.PullToRefresh)),

FeatureGate.fromFeature(GetNewerFeature)

)

override val candidateSource: CandidateSource[Unit, ShowAlertCandidate] =

StaticCandidateSource(

CandidateSourceIdentifier(identifier.name),

Seq(ShowAlertCandidate(id = identifier.name, userIds = Seq.empty))

)

override val queryTransformer: CandidatePipelineQueryTransformer[Query, Unit] = { \_ => Unit }

override val resultTransformer: CandidatePipelineResultsTransformer[

ShowAlertCandidate,

ShowAlertCandidate

] = { candidate => candidate }

override val decorator: Option[CandidateDecorator[Query, ShowAlertCandidate]] = {

val triggerDelayBuilder = new BaseDurationBuilder[Query] {

override def apply(

query: Query,

candidate: ShowAlertCandidate,

features: FeatureMap

): Option[Duration] = {

val delay = query.deviceContext.flatMap(\_.requestContextValue) match {

case Some(DeviceContext.RequestContext.TweetSelfThread) => 0.millis

case Some(DeviceContext.RequestContext.ManualRefresh) => 0.millis

case \_ => TriggerDelay

}

Some(delay)

}

}

val homeShowAlertCandidateBuilder = ShowAlertCandidateUrtItemBuilder(

alertType = NewTweets,

colorConfigBuilder = StaticShowAlertColorConfigurationBuilder(DefaultColorConfig),

displayLocationBuilder = StaticShowAlertDisplayLocationBuilder(Top),

triggerDelayBuilder = Some(triggerDelayBuilder),

displayDurationBuilder = Some(DurationParamBuilder(StaticParam(DisplayDuration))),

iconDisplayInfoBuilder = Some(StaticShowAlertIconDisplayInfoBuilder(DefaultIconDisplayInfo))

)

Some(UrtItemCandidateDecorator(homeShowAlertCandidateBuilder))

}

override val alerts = Seq(

HomeMixerAlertConfig.BusinessHours.defaultSuccessRateAlert(),

HomeMixerAlertConfig.BusinessHours.defaultEmptyResponseRateAlert()

)

}

object NewTweetsPillCandidatePipelineConfig {

val DefaultColorConfig: ShowAlertColorConfiguration = ShowAlertColorConfiguration(

background = TwitterBlueRosettaColor,

text = WhiteRosettaColor,

border = Some(WhiteRosettaColor)

)

val DefaultIconDisplayInfo: ShowAlertIconDisplayInfo =

ShowAlertIconDisplayInfo(icon = UpArrow, tint = WhiteRosettaColor)

// Unlimited display time (until user takes action)

val DisplayDuration = -1.millisecond

val TriggerDelay = 4.minutes

}