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

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

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

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

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

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

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

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

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

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

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

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.timeline\_module.ParamWhoToFollowModuleDisplayTypeBuilder

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

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

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

import com.twitter.timelineservice.model.rich.EntityIdType

import com.twitter.timelineservice.suggests.thriftscala.SuggestType

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class ForYouWhoToSubscribeCandidatePipelineConfigBuilder @Inject() (

whoToSubscribeCandidatePipelineConfigBuilder: WhoToSubscribeCandidatePipelineConfigBuilder,

homeWhoToSubscribeFeedbackActionInfoBuilder: HomeWhoToSubscribeFeedbackActionInfoBuilder) {

def build(): WhoToSubscribeCandidatePipelineConfig[ForYouQuery] = {

val gates: Seq[Gate[ForYouQuery]] = Seq(

TimelinesPersistenceStoreLastInjectionGate(

WhoToSubscribeMinInjectionIntervalParam,

PersistenceEntriesFeature,

EntityIdType.WhoToSubscribe

),

DismissFatigueGate(SuggestType.WhoToSubscribe, DismissInfoFeature)

)

whoToSubscribeCandidatePipelineConfigBuilder.build[ForYouQuery](

identifier = WhoToSubscribeCandidatePipelineConfig.identifier,

supportedClientParam = Some(EnableWhoToSubscribeCandidatePipelineParam),

alerts = alerts,

gates = gates,

moduleDisplayTypeBuilder =

ParamWhoToFollowModuleDisplayTypeBuilder(WhoToSubscribeDisplayTypeIdParam),

feedbackActionInfoBuilder = Some(homeWhoToSubscribeFeedbackActionInfoBuilder)

)

}

private val alerts = Seq(

HomeMixerAlertConfig.BusinessHours.defaultSuccessRateAlert(70),

HomeMixerAlertConfig.BusinessHours.defaultEmptyResponseRateAlert()

)

}