package com.twitter.follow\_recommendations.configapi

import com.twitter.decider.Recipient

import com.twitter.decider.SimpleRecipient

import com.twitter.follow\_recommendations.configapi.deciders.DeciderKey

import com.twitter.follow\_recommendations.configapi.deciders.DeciderParams

import com.twitter.follow\_recommendations.products.home\_timeline\_tweet\_recs.configapi.HomeTimelineTweetRecsParams

import com.twitter.servo.decider.DeciderGateBuilder

import com.twitter.timelines.configapi.\_

import com.twitter.timelines.configapi.decider.DeciderSwitchOverrideValue

import com.twitter.timelines.configapi.decider.GuestRecipient

import com.twitter.timelines.configapi.decider.RecipientBuilder

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class DeciderConfigs @Inject() (deciderGateBuilder: DeciderGateBuilder) {

val overrides: Seq[OptionalOverride[\_]] = DeciderConfigs.ParamsToDeciderMap.map {

case (params, deciderKey) =>

params.optionalOverrideValue(

DeciderSwitchOverrideValue(

feature = deciderGateBuilder.keyToFeature(deciderKey),

enabledValue = true,

recipientBuilder = DeciderConfigs.UserOrGuestOrRequest

)

)

}.toSeq

val config: BaseConfig = BaseConfigBuilder(overrides).build("FollowRecommendationServiceDeciders")

}

object DeciderConfigs {

val ParamsToDeciderMap = Map(

DeciderParams.EnableRecommendations -> DeciderKey.EnableRecommendations,

DeciderParams.EnableScoreUserCandidates -> DeciderKey.EnableScoreUserCandidates,

HomeTimelineTweetRecsParams.EnableProduct -> DeciderKey.EnableHomeTimelineTweetRecsProduct,

)

object UserOrGuestOrRequest extends RecipientBuilder {

def apply(requestContext: BaseRequestContext): Option[Recipient] = requestContext match {

case c: WithUserId if c.userId.isDefined =>

c.userId.map(SimpleRecipient)

case c: WithGuestId if c.guestId.isDefined =>

c.guestId.map(GuestRecipient)

case c: WithGuestId =>

RecipientBuilder.Request(c)

case \_ =>

throw new UndefinedUserIdNorGuestIDException(requestContext)

}

}

}