package com.twitter.timelineranker.parameters.recap\_author

import com.twitter.servo.decider.DeciderGateBuilder

import com.twitter.servo.decider.DeciderKeyName

import com.twitter.timelineranker.decider.DeciderKey

import com.twitter.timelineranker.parameters.recap\_author.RecapAuthorParams.\_

import com.twitter.timelineranker.parameters.util.ConfigHelper

import com.twitter.timelines.configapi.\_

object RecapAuthorProduction {

val deciderByParam: Map[Param[\_], DeciderKeyName] = Map[Param[\_], DeciderKeyName](

EnableContentFeaturesHydrationParam -> DeciderKey.RecapAuthorEnableContentFeaturesHydration

)

val booleanParams: Seq[EnableContentFeaturesHydrationParam.type] = Seq(

EnableContentFeaturesHydrationParam

)

val booleanFeatureSwitchParams: Seq[FSParam[Boolean]] = Seq(

EnableTokensInContentFeaturesHydrationParam,

EnableTweetTextInContentFeaturesHydrationParam,

EnableConversationControlInContentFeaturesHydrationParam,

EnableTweetMediaHydrationParam,

EnableEarlybirdRealtimeCgMigrationParam

)

}

class RecapAuthorProduction(deciderGateBuilder: DeciderGateBuilder) {

val configHelper: ConfigHelper =

new ConfigHelper(RecapAuthorProduction.deciderByParam, deciderGateBuilder)

val booleanOverrides: Seq[OptionalOverride[Boolean]] =

configHelper.createDeciderBasedBooleanOverrides(RecapAuthorProduction.booleanParams)

val booleanFeatureSwitchOverrides: Seq[OptionalOverride[Boolean]] =

FeatureSwitchOverrideUtil.getBooleanFSOverrides(

RecapAuthorProduction.booleanFeatureSwitchParams: \_\*

)

val config: BaseConfig = new BaseConfigBuilder()

.set(

booleanOverrides: \_\*

).set(

booleanFeatureSwitchOverrides: \_\*

)

.build(RecapAuthorProduction.getClass.getSimpleName)

}