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

import com.twitter.servo.decider.DeciderGateBuilder

import com.twitter.servo.decider.DeciderKeyName

import com.twitter.timelineranker.decider.DeciderKey

import com.twitter.timelineranker.parameters.recap\_hydration.RecapHydrationParams.\_

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

import com.twitter.timelines.configapi.\_

object RecapHydrationProduction {

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

EnableContentFeaturesHydrationParam -> DeciderKey.RecapHydrationEnableContentFeaturesHydration

)

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

EnableContentFeaturesHydrationParam

)

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

EnableTokensInContentFeaturesHydrationParam,

EnableTweetTextInContentFeaturesHydrationParam,

EnableConversationControlInContentFeaturesHydrationParam,

EnableTweetMediaHydrationParam

)

}

class RecapHydrationProduction(deciderGateBuilder: DeciderGateBuilder) {

val configHelper: ConfigHelper =

new ConfigHelper(RecapHydrationProduction.deciderByParam, deciderGateBuilder)

val booleanOverrides: Seq[OptionalOverride[Boolean]] =

configHelper.createDeciderBasedBooleanOverrides(RecapHydrationProduction.booleanParams)

val booleanFeatureSwitchOverrides: Seq[OptionalOverride[Boolean]] =

FeatureSwitchOverrideUtil.getBooleanFSOverrides(

RecapHydrationProduction.booleanFeatureSwitchParams: \_\*

)

val config: BaseConfig = new BaseConfigBuilder()

.set(

booleanOverrides: \_\*

).set(

booleanFeatureSwitchOverrides: \_\*

)

.build(RecapHydrationProduction.getClass.getSimpleName)

}