package com.twitter.timelineranker.parameters.recap

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.servo.decider.DeciderGateBuilder

import com.twitter.servo.decider.DeciderKeyName

import com.twitter.timelineranker.decider.DeciderKey

import com.twitter.timelineranker.parameters.recap.RecapParams.\_

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

import com.twitter.timelines.configapi.\_

import com.twitter.servo.decider.DeciderKeyEnum

object RecapProduction {

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

EnableRealGraphUsersParam -> DeciderKey.EnableRealGraphUsers,

MaxRealGraphAndFollowedUsersParam -> DeciderKey.MaxRealGraphAndFollowedUsers,

EnableContentFeaturesHydrationParam -> DeciderKey.RecapEnableContentFeaturesHydration,

MaxCountMultiplierParam -> DeciderKey.RecapMaxCountMultiplier,

EnableNewRecapAuthorPipeline -> DeciderKey.RecapAuthorEnableNewPipeline,

RecapParams.EnableExtraSortingInSearchResultParam -> DeciderKey.RecapEnableExtraSortingInResults

)

val intParams: Seq[MaxRealGraphAndFollowedUsersParam.type] = Seq(

MaxRealGraphAndFollowedUsersParam

)

val doubleParams: Seq[MaxCountMultiplierParam.type] = Seq(

MaxCountMultiplierParam

)

val boundedDoubleFeatureSwitchParams: Seq[FSBoundedParam[Double]] = Seq(

RecapParams.ProbabilityRandomTweetParam

)

val booleanParams: Seq[Param[Boolean]] = Seq(

EnableRealGraphUsersParam,

EnableContentFeaturesHydrationParam,

EnableNewRecapAuthorPipeline,

RecapParams.EnableExtraSortingInSearchResultParam

)

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

RecapParams.EnableReturnAllResultsParam,

RecapParams.IncludeRandomTweetParam,

RecapParams.IncludeSingleRandomTweetParam,

RecapParams.EnableInNetworkInReplyToTweetFeaturesHydrationParam,

RecapParams.EnableReplyRootTweetHydrationParam,

RecapParams.EnableSettingTweetTypesWithTweetKindOption,

RecapParams.EnableRelevanceSearchParam,

EnableTokensInContentFeaturesHydrationParam,

EnableTweetTextInContentFeaturesHydrationParam,

EnableExpandedExtendedRepliesFilterParam,

EnableConversationControlInContentFeaturesHydrationParam,

EnableTweetMediaHydrationParam,

ImputeRealGraphAuthorWeightsParam,

EnableExcludeSourceTweetIdsQueryParam

)

val boundedIntFeatureSwitchParams: Seq[FSBoundedParam[Int]] = Seq(

RecapParams.MaxFollowedUsersParam,

ImputeRealGraphAuthorWeightsPercentileParam,

RecapParams.RelevanceOptionsMaxHitsToProcessParam

)

}

class RecapProduction(deciderGateBuilder: DeciderGateBuilder, statsReceiver: StatsReceiver) {

val configHelper: ConfigHelper =

new ConfigHelper(RecapProduction.deciderByParam, deciderGateBuilder)

val intOverrides: Seq[OptionalOverride[Int]] =

configHelper.createDeciderBasedOverrides(RecapProduction.intParams)

val optionalBoundedIntFeatureSwitchOverrides: Seq[OptionalOverride[Option[Int]]] =

FeatureSwitchOverrideUtil.getBoundedOptionalIntOverrides(

(

MaxRealGraphAndFollowedUsersFSOverrideParam,

"max\_real\_graph\_and\_followers\_users\_fs\_override\_defined",

"max\_real\_graph\_and\_followers\_users\_fs\_override\_value"

)

)

val doubleOverrides: Seq[OptionalOverride[Double]] =

configHelper.createDeciderBasedOverrides(RecapProduction.doubleParams)

val booleanOverrides: Seq[OptionalOverride[Boolean]] =

configHelper.createDeciderBasedBooleanOverrides(RecapProduction.booleanParams)

val booleanFeatureSwitchOverrides: Seq[OptionalOverride[Boolean]] =

FeatureSwitchOverrideUtil.getBooleanFSOverrides(RecapProduction.booleanFeatureSwitchParams: \_\*)

val boundedDoubleFeatureSwitchOverrides: Seq[OptionalOverride[Double]] =

FeatureSwitchOverrideUtil.getBoundedDoubleFSOverrides(

RecapProduction.boundedDoubleFeatureSwitchParams: \_\*)

val boundedIntFeatureSwitchOverrides: Seq[OptionalOverride[Int]] =

FeatureSwitchOverrideUtil.getBoundedIntFSOverrides(

RecapProduction.boundedIntFeatureSwitchParams: \_\*)

val config: BaseConfig = new BaseConfigBuilder()

.set(

intOverrides: \_\*

)

.set(

booleanOverrides: \_\*

)

.set(

doubleOverrides: \_\*

)

.set(

booleanFeatureSwitchOverrides: \_\*

)

.set(

boundedIntFeatureSwitchOverrides: \_\*

)

.set(

optionalBoundedIntFeatureSwitchOverrides: \_\*

)

.set(

boundedDoubleFeatureSwitchOverrides: \_\*

)

.build(RecapProduction.getClass.getSimpleName)

}