package com.twitter.timelineranker.parameters.uteg\_liked\_by\_tweets

import com.twitter.servo.decider.DeciderGateBuilder

import com.twitter.servo.decider.DeciderKeyName

import com.twitter.timelineranker.decider.DeciderKey

import com.twitter.timelineranker.parameters.uteg\_liked\_by\_tweets.UtegLikedByTweetsParams.\_

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

import com.twitter.timelines.configapi.BaseConfig

import com.twitter.timelines.configapi.BaseConfigBuilder

import com.twitter.timelines.configapi.FSBoundedParam

import com.twitter.timelines.configapi.FSParam

import com.twitter.timelines.configapi.FeatureSwitchOverrideUtil

import com.twitter.timelines.configapi.OptionalOverride

import com.twitter.timelines.configapi.Param

object UtegLikedByTweetsProduction {

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

EnableContentFeaturesHydrationParam -> DeciderKey.UtegLikedByTweetsEnableContentFeaturesHydration

)

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

EnableContentFeaturesHydrationParam

)

val intParams: Seq[Param[Int]] = Seq(

DefaultUTEGInNetworkCount,

DefaultMaxTweetCount,

DefaultUTEGOutOfNetworkCount,

MinNumFavoritedByUserIdsParam

)

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

UTEGRecommendationsFilter.EnableParam,

UTEGRecommendationsFilter.ExcludeQuoteTweetParam,

UTEGRecommendationsFilter.ExcludeReplyParam,

UTEGRecommendationsFilter.ExcludeRetweetParam,

UTEGRecommendationsFilter.ExcludeTweetParam,

EnableTokensInContentFeaturesHydrationParam,

EnableConversationControlInContentFeaturesHydrationParam,

UTEGRecommendationsFilter.ExcludeRecommendedRepliesToNonFollowedUsersParam,

EnableTweetTextInContentFeaturesHydrationParam,

EnableTweetMediaHydrationParam,

UtegLikedByTweetsParams.IncludeRandomTweetParam,

UtegLikedByTweetsParams.IncludeSingleRandomTweetParam,

UtegLikedByTweetsParams.EnableRelevanceSearchParam

)

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

EarlybirdScoreMultiplierParam,

UtegLikedByTweetsParams.ProbabilityRandomTweetParam

)

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

UtegLikedByTweetsParams.NumAdditionalRepliesParam

)

}

class UtegLikedByTweetsProduction(deciderGateBuilder: DeciderGateBuilder) {

val configHelper: ConfigHelper =

new ConfigHelper(UtegLikedByTweetsProduction.deciderByParam, deciderGateBuilder)

val booleanDeciderOverrides: Seq[OptionalOverride[Boolean]] =

configHelper.createDeciderBasedBooleanOverrides(

UtegLikedByTweetsProduction.booleanDeciderParams)

val boundedDoubleFeatureSwitchOverrides: Seq[OptionalOverride[Double]] =

FeatureSwitchOverrideUtil.getBoundedDoubleFSOverrides(

UtegLikedByTweetsProduction.boundedDoubleFeatureSwitchParams: \_\*)

val booleanFeatureSwitchOverrides: Seq[OptionalOverride[Boolean]] =

FeatureSwitchOverrideUtil.getBooleanFSOverrides(

UtegLikedByTweetsProduction.booleanFeatureSwitchParams: \_\*)

val boundedIntFeaturesSwitchOverrides: Seq[OptionalOverride[Int]] =

FeatureSwitchOverrideUtil.getBoundedIntFSOverrides(

UtegLikedByTweetsProduction.boundedIntFeatureSwitchParams: \_\*)

val config: BaseConfig = new BaseConfigBuilder()

.set(

booleanDeciderOverrides: \_\*

)

.set(

boundedDoubleFeatureSwitchOverrides: \_\*

)

.set(

booleanFeatureSwitchOverrides: \_\*

)

.set(

boundedIntFeaturesSwitchOverrides: \_\*

)

.build(UtegLikedByTweetsProduction.getClass.getSimpleName)

}