package com.twitter.cr\_mixer.param

import com.twitter.finagle.stats.NullStatsReceiver

import com.twitter.logging.Logger

import com.twitter.timelines.configapi.BaseConfig

import com.twitter.timelines.configapi.BaseConfigBuilder

import com.twitter.timelines.configapi.FSBoundedParam

import com.twitter.timelines.configapi.FSEnumParam

import com.twitter.timelines.configapi.FSName

import com.twitter.timelines.configapi.FSParam

import com.twitter.timelines.configapi.FeatureSwitchOverrideUtil

import com.twitter.timelines.configapi.Param

object BlenderParams {

object BlendingAlgorithmEnum extends Enumeration {

val RoundRobin: Value = Value

val SourceTypeBackFill: Value = Value

val SourceSignalSorting: Value = Value

}

object ContentBasedSortingAlgorithmEnum extends Enumeration {

val FavoriteCount: Value = Value

val SourceSignalRecency: Value = Value

val RandomSorting: Value = Value

val SimilarityToSignalSorting: Value = Value

val CandidateRecency: Value = Value

}

object BlendingAlgorithmParam

extends FSEnumParam[BlendingAlgorithmEnum.type](

name = "blending\_algorithm\_id",

default = BlendingAlgorithmEnum.RoundRobin,

enum = BlendingAlgorithmEnum

)

object RankingInterleaveWeightShrinkageParam

extends FSBoundedParam[Double](

name = "blending\_enable\_ml\_ranking\_interleave\_weights\_shrinkage",

default = 1.0,

min = 0.0,

max = 1.0

)

object RankingInterleaveMaxWeightAdjustments

extends FSBoundedParam[Int](

name = "blending\_interleave\_max\_weighted\_adjustments",

default = 3000,

min = 0,

max = 9999

)

object SignalTypeSortingAlgorithmParam

extends FSEnumParam[ContentBasedSortingAlgorithmEnum.type](

name = "blending\_algorithm\_inner\_signal\_sorting\_id",

default = ContentBasedSortingAlgorithmEnum.SourceSignalRecency,

enum = ContentBasedSortingAlgorithmEnum

)

object ContentBlenderTypeSortingAlgorithmParam

extends FSEnumParam[ContentBasedSortingAlgorithmEnum.type](

name = "blending\_algorithm\_content\_blender\_sorting\_id",

default = ContentBasedSortingAlgorithmEnum.FavoriteCount,

enum = ContentBasedSortingAlgorithmEnum

)

//UserAffinities Algo Param: whether to distributed the source type weights

object EnableDistributedSourceTypeWeightsParam

extends FSParam[Boolean](

name = "blending\_algorithm\_enable\_distributed\_source\_type\_weights",

default = false

)

object BlendGroupingMethodEnum extends Enumeration {

val SourceKeyDefault: Value = Value("SourceKey")

val SourceTypeSimilarityEngine: Value = Value("SourceTypeSimilarityEngine")

val AuthorId: Value = Value("AuthorId")

}

object BlendGroupingMethodParam

extends FSEnumParam[BlendGroupingMethodEnum.type](

name = "blending\_grouping\_method\_id",

default = BlendGroupingMethodEnum.SourceKeyDefault,

enum = BlendGroupingMethodEnum

)

object RecencyBasedRandomSamplingHalfLifeInDays

extends FSBoundedParam[Int](

name = "blending\_interleave\_random\_sampling\_recency\_based\_half\_life\_in\_days",

default = 7,

min = 1,

max = 28

)

object RecencyBasedRandomSamplingDefaultWeight

extends FSBoundedParam[Double](

name = "blending\_interleave\_random\_sampling\_recency\_based\_default\_weight",

default = 1.0,

min = 0.1,

max = 2.0

)

object SourceTypeBackFillEnableVideoBackFill

extends FSParam[Boolean](

name = "blending\_enable\_video\_backfill",

default = false

)

val AllParams: Seq[Param[\_] with FSName] = Seq(

BlendingAlgorithmParam,

RankingInterleaveWeightShrinkageParam,

RankingInterleaveMaxWeightAdjustments,

EnableDistributedSourceTypeWeightsParam,

BlendGroupingMethodParam,

RecencyBasedRandomSamplingHalfLifeInDays,

RecencyBasedRandomSamplingDefaultWeight,

SourceTypeBackFillEnableVideoBackFill,

SignalTypeSortingAlgorithmParam,

ContentBlenderTypeSortingAlgorithmParam,

)

lazy val config: BaseConfig = {

val enumOverrides = FeatureSwitchOverrideUtil.getEnumFSOverrides(

NullStatsReceiver,

Logger(getClass),

BlendingAlgorithmParam,

BlendGroupingMethodParam,

SignalTypeSortingAlgorithmParam,

ContentBlenderTypeSortingAlgorithmParam

)

val booleanOverrides = FeatureSwitchOverrideUtil.getBooleanFSOverrides(

EnableDistributedSourceTypeWeightsParam,

SourceTypeBackFillEnableVideoBackFill

)

val intOverrides = FeatureSwitchOverrideUtil.getBoundedIntFSOverrides(

RankingInterleaveMaxWeightAdjustments,

RecencyBasedRandomSamplingHalfLifeInDays

)

val doubleOverrides = FeatureSwitchOverrideUtil.getBoundedDoubleFSOverrides(

RankingInterleaveWeightShrinkageParam,

RecencyBasedRandomSamplingDefaultWeight

)

BaseConfigBuilder()

.set(enumOverrides: \_\*)

.set(booleanOverrides: \_\*)

.set(intOverrides: \_\*)

.set(doubleOverrides: \_\*)

.build()

}

}