package com.twitter.cr\_mixer.param

import com.twitter.timelines.configapi.BaseConfig

import com.twitter.timelines.configapi.BaseConfigBuilder

import com.twitter.timelines.configapi.FSBoundedParam

import com.twitter.timelines.configapi.FSName

import com.twitter.timelines.configapi.FSParam

import com.twitter.timelines.configapi.FeatureSwitchOverrideUtil

import com.twitter.timelines.configapi.Param

import com.twitter.follow\_recommendations.thriftscala.DisplayLocation

import com.twitter.timelines.configapi.FSEnumParam

import com.twitter.logging.Logger

import com.twitter.finagle.stats.NullStatsReceiver

object FrsParams {

object EnableSourceParam

extends FSParam[Boolean](

name = "signal\_frs\_enable\_source",

default = false

)

object EnableSourceGraphParam

extends FSParam[Boolean](

name = "graph\_frs\_enable\_source",

default = false

)

object MinScoreParam

extends FSBoundedParam[Double](

name = "signal\_frs\_min\_score",

default = 0.4,

min = 0.0,

max = 1.0

)

object MaxConsumerSeedsNumParam

extends FSBoundedParam[Int](

name = "graph\_frs\_max\_user\_seeds\_num",

default = 200,

min = 0,

max = 1000

)

/\*\*

\* These params below are only used for FrsTweetCandidateGenerator and shouldn't be used in other endpoints

\* \* FrsBasedCandidateGenerationMaxSeedsNumParam

\* \* FrsCandidateGenerationDisplayLocationParam

\* \* FrsCandidateGenerationDisplayLocation

\* \* FrsBasedCandidateGenerationMaxCandidatesNumParam

\*/

object FrsBasedCandidateGenerationEnableVisibilityFilteringParam

extends FSParam[Boolean](

name = "frs\_based\_candidate\_generation\_enable\_vf",

default = true

)

object FrsBasedCandidateGenerationMaxSeedsNumParam

extends FSBoundedParam[Int](

name = "frs\_based\_candidate\_generation\_max\_seeds\_num",

default = 100,

min = 0,

max = 800

)

object FrsBasedCandidateGenerationDisplayLocation extends Enumeration {

protected case class FrsDisplayLocationValue(displayLocation: DisplayLocation) extends super.Val

import scala.language.implicitConversions

implicit def valueToDisplayLocationValue(x: Value): FrsDisplayLocationValue =

x.asInstanceOf[FrsDisplayLocationValue]

val DisplayLocation\_ContentRecommender: FrsDisplayLocationValue = FrsDisplayLocationValue(

DisplayLocation.ContentRecommender)

val DisplayLocation\_Home: FrsDisplayLocationValue = FrsDisplayLocationValue(

DisplayLocation.HomeTimelineTweetRecs)

val DisplayLocation\_Notifications: FrsDisplayLocationValue = FrsDisplayLocationValue(

DisplayLocation.TweetNotificationRecs)

}

object FrsBasedCandidateGenerationDisplayLocationParam

extends FSEnumParam[FrsBasedCandidateGenerationDisplayLocation.type](

name = "frs\_based\_candidate\_generation\_display\_location\_id",

default = FrsBasedCandidateGenerationDisplayLocation.DisplayLocation\_Home,

enum = FrsBasedCandidateGenerationDisplayLocation

)

object FrsBasedCandidateGenerationMaxCandidatesNumParam

extends FSBoundedParam[Int](

name = "frs\_based\_candidate\_generation\_max\_candidates\_num",

default = 100,

min = 0,

max = 2000

)

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

EnableSourceParam,

EnableSourceGraphParam,

MinScoreParam,

MaxConsumerSeedsNumParam,

FrsBasedCandidateGenerationMaxSeedsNumParam,

FrsBasedCandidateGenerationDisplayLocationParam,

FrsBasedCandidateGenerationMaxCandidatesNumParam,

FrsBasedCandidateGenerationEnableVisibilityFilteringParam

)

lazy val config: BaseConfig = {

val booleanOverrides = FeatureSwitchOverrideUtil.getBooleanFSOverrides(

EnableSourceParam,

EnableSourceGraphParam,

FrsBasedCandidateGenerationEnableVisibilityFilteringParam

)

val doubleOverrides = FeatureSwitchOverrideUtil.getBoundedDoubleFSOverrides(MinScoreParam)

val intOverrides = FeatureSwitchOverrideUtil.getBoundedIntFSOverrides(

MaxConsumerSeedsNumParam,

FrsBasedCandidateGenerationMaxSeedsNumParam,

FrsBasedCandidateGenerationMaxCandidatesNumParam)

val enumOverrides = FeatureSwitchOverrideUtil.getEnumFSOverrides(

NullStatsReceiver,

Logger(getClass),

FrsBasedCandidateGenerationDisplayLocationParam,

)

BaseConfigBuilder()

.set(booleanOverrides: \_\*)

.set(doubleOverrides: \_\*)

.set(intOverrides: \_\*)

.set(enumOverrides: \_\*)

.build()

}

}