package com.twitter.cr\_mixer.param

import com.twitter.conversions.DurationOps.\_

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.DurationConversion

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.HasDurationConversion

import com.twitter.timelines.configapi.Param

import com.twitter.util.Duration

object TopicTweetParams {

object MaxTweetAge

extends FSBoundedParam[Duration](

name = "topic\_tweet\_candidate\_generation\_max\_tweet\_age\_hours",

default = 24.hours,

min = 12.hours,

max = 48.hours

)

with HasDurationConversion {

override val durationConversion: DurationConversion = DurationConversion.FromHours

}

object MaxTopicTweetCandidatesParam

extends FSBoundedParam[Int](

name = "topic\_tweet\_max\_candidates\_num",

default = 200,

min = 0,

max = 1000

)

object MaxSkitTfgCandidatesParam

extends FSBoundedParam[Int](

name = "topic\_tweet\_skit\_tfg\_max\_candidates\_num",

default = 100,

min = 0,

max = 1000

)

object MaxSkitHighPrecisionCandidatesParam

extends FSBoundedParam[Int](

name = "topic\_tweet\_skit\_high\_precision\_max\_candidates\_num",

default = 100,

min = 0,

max = 1000

)

object MaxCertoCandidatesParam

extends FSBoundedParam[Int](

name = "topic\_tweet\_certo\_max\_candidates\_num",

default = 100,

min = 0,

max = 1000

)

// The min prod score for Certo L2-normalized cosine candidates

object CertoScoreThresholdParam

extends FSBoundedParam[Double](

name = "topic\_tweet\_certo\_score\_threshold",

default = 0.015,

min = 0,

max = 1

)

object SemanticCoreVersionIdParam

extends FSParam[Long](

name = "semantic\_core\_version\_id",

default = 1380520918896713735L

)

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

CertoScoreThresholdParam,

MaxTopicTweetCandidatesParam,

MaxTweetAge,

MaxCertoCandidatesParam,

MaxSkitTfgCandidatesParam,

MaxSkitHighPrecisionCandidatesParam,

SemanticCoreVersionIdParam

)

lazy val config: BaseConfig = {

val booleanOverrides = FeatureSwitchOverrideUtil.getBooleanFSOverrides()

val doubleOverrides =

FeatureSwitchOverrideUtil.getBoundedDoubleFSOverrides(CertoScoreThresholdParam)

val intOverrides = FeatureSwitchOverrideUtil.getBoundedIntFSOverrides(

MaxCertoCandidatesParam,

MaxSkitTfgCandidatesParam,

MaxSkitHighPrecisionCandidatesParam,

MaxTopicTweetCandidatesParam

)

val longOverrides = FeatureSwitchOverrideUtil.getLongFSOverrides(SemanticCoreVersionIdParam)

val durationFSOverrides = FeatureSwitchOverrideUtil.getDurationFSOverrides(MaxTweetAge)

val enumOverrides =

FeatureSwitchOverrideUtil.getEnumFSOverrides(NullStatsReceiver, Logger(getClass))

BaseConfigBuilder()

.set(booleanOverrides: \_\*)

.set(doubleOverrides: \_\*)

.set(intOverrides: \_\*)

.set(longOverrides: \_\*)

.set(enumOverrides: \_\*)

.set(durationFSOverrides: \_\*)

.build()

}

}