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

/\*\*

\* ConsumersBasedUserVideoGraph Params: there are multiple ways (e.g. FRS, RealGraphIn) to generate consumersSeedSet for ConsumersBasedUserTweetGraph

\* for now we allow flexibility in tuning UVG params for different consumersSeedSet generation algo by giving the param name {consumerSeedSetAlgo}{ParamName}

\*/

object ConsumersBasedUserVideoGraphParams {

object EnableSourceParam

extends FSParam[Boolean](

name = "consumers\_based\_user\_video\_graph\_enable\_source",

default = false

)

// UTG-RealGraphIN

object RealGraphInMinCoOccurrenceParam

extends FSBoundedParam[Int](

name = "consumers\_based\_user\_video\_graph\_real\_graph\_in\_min\_co\_occurrence",

default = 3,

min = 0,

max = 500

)

object RealGraphInMinScoreParam

extends FSBoundedParam[Double](

name = "consumers\_based\_user\_video\_graph\_real\_graph\_in\_min\_score",

default = 2.0,

min = 0.0,

max = 10.0

)

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

EnableSourceParam,

RealGraphInMinCoOccurrenceParam,

RealGraphInMinScoreParam

)

lazy val config: BaseConfig = {

val intOverrides =

FeatureSwitchOverrideUtil.getBoundedIntFSOverrides(RealGraphInMinCoOccurrenceParam)

val doubleOverrides =

FeatureSwitchOverrideUtil.getBoundedDoubleFSOverrides(RealGraphInMinScoreParam)

val booleanOverrides = FeatureSwitchOverrideUtil.getBooleanFSOverrides(

EnableSourceParam

)

BaseConfigBuilder()

.set(intOverrides: \_\*)

.set(booleanOverrides: \_\*)

.set(doubleOverrides: \_\*)

.build()

}

}