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

object ConsumersBasedUserAdGraphParams {

object EnableSourceParam

extends FSParam[Boolean](

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

default = false

)

// UTG-Lookalike

object MinCoOccurrenceParam

extends FSBoundedParam[Int](

name = "consumers\_based\_user\_ad\_graph\_min\_co\_occurrence",

default = 2,

min = 0,

max = 500

)

object MinScoreParam

extends FSBoundedParam[Double](

name = "consumers\_based\_user\_ad\_graph\_min\_score",

default = 0.0,

min = 0.0,

max = 10.0

)

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

EnableSourceParam,

MinCoOccurrenceParam,

MinScoreParam

)

lazy val config: BaseConfig = {

val intOverrides = FeatureSwitchOverrideUtil.getBoundedIntFSOverrides(MinCoOccurrenceParam)

val doubleOverrides = FeatureSwitchOverrideUtil.getBoundedDoubleFSOverrides(MinScoreParam)

val booleanOverrides = FeatureSwitchOverrideUtil.getBooleanFSOverrides(EnableSourceParam)

BaseConfigBuilder()

.set(intOverrides: \_\*)

.set(booleanOverrides: \_\*)

.set(doubleOverrides: \_\*)

.build()

}

}