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.FeatureSwitchOverrideUtil

import com.twitter.timelines.configapi.Param

object TweetBasedUserAdGraphParams {

object MinCoOccurrenceParam

extends FSBoundedParam[Int](

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

default = 1,

min = 0,

max = 500

)

object ConsumersBasedMinScoreParam

extends FSBoundedParam[Double](

name = "tweet\_based\_user\_ad\_graph\_consumers\_based\_min\_score",

default = 0.0,

min = 0.0,

max = 10.0

)

object MaxConsumerSeedsNumParam

extends FSBoundedParam[Int](

name = "tweet\_based\_user\_ad\_graph\_max\_user\_seeds\_num",

default = 100,

min = 0,

max = 300

)

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

MinCoOccurrenceParam,

MaxConsumerSeedsNumParam,

ConsumersBasedMinScoreParam

)

lazy val config: BaseConfig = {

val intOverrides = FeatureSwitchOverrideUtil.getBoundedIntFSOverrides(

MinCoOccurrenceParam,

MaxConsumerSeedsNumParam

)

val doubleOverrides =

FeatureSwitchOverrideUtil.getBoundedDoubleFSOverrides(ConsumersBasedMinScoreParam)

BaseConfigBuilder()

.set(intOverrides: \_\*)

.set(doubleOverrides: \_\*)

.build()

}

}