package com.twitter.cr\_mixer.param

import com.twitter.timelines.configapi.BaseConfig

import com.twitter.timelines.configapi.BaseConfigBuilder

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 ConsumerEmbeddingBasedCandidateGenerationParams {

object EnableTwHINParam

extends FSParam[Boolean](

name = "consumer\_embedding\_based\_candidate\_generation\_enable\_twhin",

default = false

)

object EnableTwoTowerParam

extends FSParam[Boolean](

name = "consumer\_embedding\_based\_candidate\_generation\_enable\_two\_tower",

default = false

)

object EnableLogFavBasedSimClustersTripParam

extends FSParam[Boolean](

name = "consumer\_embedding\_based\_candidate\_generation\_enable\_logfav\_based\_simclusters\_trip",

default = false

)

object EnableFollowBasedSimClustersTripParam

extends FSParam[Boolean](

name = "consumer\_embedding\_based\_candidate\_generation\_enable\_follow\_based\_simclusters\_trip",

default = false

)

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

EnableTwHINParam,

EnableTwoTowerParam,

EnableFollowBasedSimClustersTripParam,

EnableLogFavBasedSimClustersTripParam

)

lazy val config: BaseConfig = {

val booleanOverrides = FeatureSwitchOverrideUtil.getBooleanFSOverrides(

EnableTwHINParam,

EnableTwoTowerParam,

EnableFollowBasedSimClustersTripParam,

EnableLogFavBasedSimClustersTripParam

)

BaseConfigBuilder()

.set(booleanOverrides: \_\*)

.build()

}

}