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 BypassInterleaveAndRankParams {

object EnableTwhinCollabFilterBypassParam

extends FSParam[Boolean](

name = "bypass\_interleave\_and\_rank\_twhin\_collab\_filter",

default = false

)

object EnableTwoTowerBypassParam

extends FSParam[Boolean](

name = "bypass\_interleave\_and\_rank\_two\_tower",

default = false

)

object EnableConsumerBasedTwhinBypassParam

extends FSParam[Boolean](

name = "bypass\_interleave\_and\_rank\_consumer\_based\_twhin",

default = false

)

object EnableConsumerBasedWalsBypassParam

extends FSParam[Boolean](

name = "bypass\_interleave\_and\_rank\_consumer\_based\_wals",

default = false

)

object TwhinCollabFilterBypassPercentageParam

extends FSBoundedParam[Double](

name = "bypass\_interleave\_and\_rank\_twhin\_collab\_filter\_percentage",

default = 0.0,

min = 0.0,

max = 1.0

)

object TwoTowerBypassPercentageParam

extends FSBoundedParam[Double](

name = "bypass\_interleave\_and\_rank\_two\_tower\_percentage",

default = 0.0,

min = 0.0,

max = 1.0

)

object ConsumerBasedTwhinBypassPercentageParam

extends FSBoundedParam[Double](

name = "bypass\_interleave\_and\_rank\_consumer\_based\_twhin\_percentage",

default = 0.0,

min = 0.0,

max = 1.0

)

object ConsumerBasedWalsBypassPercentageParam

extends FSBoundedParam[Double](

name = "bypass\_interleave\_and\_rank\_consumer\_based\_wals\_percentage",

default = 0.0,

min = 0.0,

max = 1.0

)

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

EnableTwhinCollabFilterBypassParam,

EnableTwoTowerBypassParam,

EnableConsumerBasedTwhinBypassParam,

EnableConsumerBasedWalsBypassParam,

TwhinCollabFilterBypassPercentageParam,

TwoTowerBypassPercentageParam,

ConsumerBasedTwhinBypassPercentageParam,

ConsumerBasedWalsBypassPercentageParam,

)

lazy val config: BaseConfig = {

val booleanOverrides = FeatureSwitchOverrideUtil.getBooleanFSOverrides(

EnableTwhinCollabFilterBypassParam,

EnableTwoTowerBypassParam,

EnableConsumerBasedTwhinBypassParam,

EnableConsumerBasedWalsBypassParam,

)

val doubleOverrides = FeatureSwitchOverrideUtil.getBoundedDoubleFSOverrides(

TwhinCollabFilterBypassPercentageParam,

TwoTowerBypassPercentageParam,

ConsumerBasedTwhinBypassPercentageParam,

ConsumerBasedWalsBypassPercentageParam,

)

BaseConfigBuilder()

.set(booleanOverrides: \_\*)

.set(doubleOverrides: \_\*)

.build()

}

}