package com.twitter.cr\_mixer.param

import com.twitter.finagle.stats.NullStatsReceiver

import com.twitter.logging.Logger

import com.twitter.timelines.configapi.BaseConfig

import com.twitter.timelines.configapi.BaseConfigBuilder

import com.twitter.timelines.configapi.FSEnumParam

import com.twitter.timelines.configapi.FSName

import com.twitter.timelines.configapi.FSParam

import com.twitter.timelines.configapi.FeatureSwitchOverrideUtil

import com.twitter.timelines.configapi.Param

import com.twitter.usersignalservice.thriftscala.SignalType

object GoodProfileClickParams {

object ClickMinDwellTimeParam extends Enumeration {

protected case class SignalTypeValue(signalType: SignalType) extends super.Val

import scala.language.implicitConversions

implicit def valueToSignalTypeValue(x: Value): SignalTypeValue =

x.asInstanceOf[SignalTypeValue]

val TotalDwellTime10s = SignalTypeValue(SignalType.GoodProfileClick)

val TotalDwellTime20s = SignalTypeValue(SignalType.GoodProfileClick20s)

val TotalDwellTime30s = SignalTypeValue(SignalType.GoodProfileClick30s)

}

object EnableSourceParam

extends FSParam[Boolean](

name = "signal\_good\_profile\_clicks\_enable\_source",

default = false

)

object ClickMinDwellTimeType

extends FSEnumParam[ClickMinDwellTimeParam.type](

name = "signal\_good\_profile\_clicks\_min\_dwelltime\_type\_id",

default = ClickMinDwellTimeParam.TotalDwellTime10s,

enum = ClickMinDwellTimeParam

)

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

Seq(EnableSourceParam, ClickMinDwellTimeType)

lazy val config: BaseConfig = {

val booleanOverrides = FeatureSwitchOverrideUtil.getBooleanFSOverrides(

EnableSourceParam

)

val enumOverrides = FeatureSwitchOverrideUtil.getEnumFSOverrides(

NullStatsReceiver,

Logger(getClass),

ClickMinDwellTimeType

)

BaseConfigBuilder()

.set(booleanOverrides: \_\*)

.set(enumOverrides: \_\*)

.build()

}

}