package com.twitter.product\_mixer.core.functional\_component.configapi.registry

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.logging.Logger

import com.twitter.servo.decider.DeciderGateBuilder

import com.twitter.timelines.configapi.FeatureSwitchOverrideUtil

import com.twitter.timelines.configapi.OptionalOverride

import com.twitter.timelines.configapi.decider.DeciderUtils

trait ParamConfigBuilder { paramConfig: ParamConfig =>

/\*\* Builds a Seq of [[OptionalOverride]]s based on the [[paramConfig]] \*/

def build(

deciderGateBuilder: DeciderGateBuilder,

statsReceiver: StatsReceiver

): Seq[OptionalOverride[\_]] = {

val logger = Logger(classOf[ParamConfig])

DeciderUtils.getBooleanDeciderOverrides(deciderGateBuilder, booleanDeciderOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getBooleanFSOverrides(booleanFSOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getOptionalBooleanOverrides(optionalBooleanOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getEnumFSOverrides(

statsReceiver.scope("enumConversion"),

logger,

enumFSOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getEnumSeqFSOverrides(

statsReceiver.scope("enumSeqConversion"),

logger,

enumSeqFSOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getBoundedDurationFSOverrides(boundedDurationFSOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getBoundedIntFSOverrides(boundedIntFSOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getBoundedOptionalIntOverrides(boundedOptionalIntOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getIntSeqFSOverrides(intSeqFSOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getBoundedLongFSOverrides(boundedLongFSOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getBoundedOptionalLongOverrides(boundedOptionalLongOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getLongSeqFSOverrides(longSeqFSOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getLongSetFSOverrides(longSetFSOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getBoundedDoubleFSOverrides(boundedDoubleFSOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getBoundedOptionalDoubleOverrides(

boundedOptionalDoubleOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getDoubleSeqFSOverrides(doubleSeqFSOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getStringFSOverrides(stringFSOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getStringSeqFSOverrides(stringSeqFSOverrides: \_\*) ++

FeatureSwitchOverrideUtil.getOptionalStringOverrides(optionalStringOverrides: \_\*) ++

gatedOverrides.flatMap {

case (fsName, overrides) => FeatureSwitchOverrideUtil.gatedOverrides(fsName, overrides: \_\*)

}.toSeq

}

}