package com.twitter.visibility.configapi

import com.twitter.decider.Decider

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.logging.Logger

import com.twitter.servo.decider.DeciderGateBuilder

import com.twitter.timelines.configapi.CompositeConfig

import com.twitter.timelines.configapi.Config

import com.twitter.util.Memoize

import com.twitter.visibility.configapi.configs.VisibilityDeciders

import com.twitter.visibility.configapi.configs.VisibilityExperimentsConfig

import com.twitter.visibility.configapi.configs.VisibilityFeatureSwitches

import com.twitter.visibility.models.SafetyLevel

object ConfigBuilder {

def apply(statsReceiver: StatsReceiver, decider: Decider, logger: Logger): ConfigBuilder = {

val deciderGateBuilder: DeciderGateBuilder =

new DeciderGateBuilder(decider)

new ConfigBuilder(

deciderGateBuilder,

statsReceiver,

logger

)

}

}

class ConfigBuilder(

deciderGateBuilder: DeciderGateBuilder,

statsReceiver: StatsReceiver,

logger: Logger) {

def buildMemoized: SafetyLevel => Config = Memoize(build)

def build(safetyLevel: SafetyLevel): Config = {

new CompositeConfig(

VisibilityExperimentsConfig.config(safetyLevel) :+

VisibilityDeciders.config(deciderGateBuilder, logger, statsReceiver, safetyLevel) :+

VisibilityFeatureSwitches.config(statsReceiver, logger)

)

}

}