package com.twitter.tsp.common

import com.twitter.abdecider.LoggingABDecider

import com.twitter.featureswitches.v2.FeatureSwitches

import com.twitter.featureswitches.v2.builder.{FeatureSwitchesBuilder => FsBuilder}

import com.twitter.featureswitches.v2.experimentation.NullBucketImpressor

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.util.Duration

case class FeatureSwitchesBuilder(

statsReceiver: StatsReceiver,

abDecider: LoggingABDecider,

featuresDirectory: String,

addServiceDetailsFromAurora: Boolean,

configRepoDirectory: String = "/usr/local/config",

fastRefresh: Boolean = false,

impressExperiments: Boolean = true) {

def build(): FeatureSwitches = {

val featureSwitches = FsBuilder()

.abDecider(abDecider)

.statsReceiver(statsReceiver)

.configRepoAbsPath(configRepoDirectory)

.featuresDirectory(featuresDirectory)

.limitToReferencedExperiments(shouldLimit = true)

.experimentImpressionStatsEnabled(true)

if (!impressExperiments) featureSwitches.experimentBucketImpressor(NullBucketImpressor)

if (addServiceDetailsFromAurora) featureSwitches.serviceDetailsFromAurora()

if (fastRefresh) featureSwitches.refreshPeriod(Duration.fromSeconds(10))

featureSwitches.build()

}

}