package com.twitter.cr\_mixer.module.core

import com.google.inject.Provides

import com.twitter.cr\_mixer.featureswitch.CrMixerLoggingABDecider

import com.twitter.featureswitches.v2.FeatureSwitches

import com.twitter.featureswitches.v2.builder.FeatureSwitchesBuilder

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

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.inject.TwitterModule

import com.twitter.inject.annotations.Flag

import com.twitter.util.Duration

import javax.inject.Singleton

object FeatureSwitchesModule extends TwitterModule {

flag(

name = "featureswitches.path",

default = "/features/cr-mixer/main",

help = "path to the featureswitch configuration directory"

)

flag(

"use\_config\_repo\_mirror.bool",

false,

"If true, read config from a different directory, to facilitate testing.")

val DefaultFastRefresh: Boolean = false

val AddServiceDetailsFromAurora: Boolean = true

val ImpressExperiments: Boolean = true

@Provides

@Singleton

def providesFeatureSwitches(

@Flag("featureswitches.path") featureSwitchDirectory: String,

@Flag("use\_config\_repo\_mirror.bool") useConfigRepoMirrorFlag: Boolean,

abDecider: CrMixerLoggingABDecider,

statsReceiver: StatsReceiver

): FeatureSwitches = {

val configRepoAbsPath =

getConfigRepoAbsPath(useConfigRepoMirrorFlag)

val fastRefresh =

shouldFastRefresh(useConfigRepoMirrorFlag)

val featureSwitches = FeatureSwitchesBuilder()

.abDecider(abDecider)

.statsReceiver(statsReceiver.scope("featureswitches-v2"))

.configRepoAbsPath(configRepoAbsPath)

.featuresDirectory(featureSwitchDirectory)

.limitToReferencedExperiments(shouldLimit = true)

.experimentImpressionStatsEnabled(true)

if (!ImpressExperiments) featureSwitches.experimentBucketImpressor(NullBucketImpressor)

if (AddServiceDetailsFromAurora) featureSwitches.serviceDetailsFromAurora()

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

featureSwitches.build()

}

private def getConfigRepoAbsPath(

useConfigRepoMirrorFlag: Boolean

): String = {

if (useConfigRepoMirrorFlag)

"config\_repo\_mirror/"

else "/usr/local/config"

}

private def shouldFastRefresh(

useConfigRepoMirrorFlag: Boolean

): Boolean = {

if (useConfigRepoMirrorFlag)

true

else DefaultFastRefresh

}

}