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

import com.google.inject.Provides

import com.twitter.abdecider.LoggingABDecider

import com.twitter.featureswitches.v2.FeatureSwitches

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

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.inject.TwitterModule

import com.twitter.inject.annotations.Flag

import com.twitter.product\_mixer.core.module.product\_mixer\_flags.ProductMixerFlagModule.ConfigRepoLocalPath

import com.twitter.product\_mixer.core.module.product\_mixer\_flags.ProductMixerFlagModule.FeatureSwitchesPath

import com.twitter.product\_mixer.core.module.product\_mixer\_flags.ProductMixerFlagModule.ServiceLocal

import com.twitter.timelines.features.app.ForcibleFeatureValuesModule

import javax.inject.Singleton

object FeatureSwitchesModule extends TwitterModule with ForcibleFeatureValuesModule {

private val DefaultConfigRepoPath = "/usr/local/config"

@Provides

@Singleton

def providesFeatureSwitches(

abDecider: LoggingABDecider,

statsReceiver: StatsReceiver,

@Flag(ServiceLocal) isServiceLocal: Boolean,

@Flag(ConfigRepoLocalPath) localConfigRepoPath: String,

@Flag(FeatureSwitchesPath) featuresPath: String

): FeatureSwitches = {

val configRepoPath = if (isServiceLocal) {

localConfigRepoPath

} else {

DefaultConfigRepoPath

}

val baseBuilder = FeatureSwitchesBuilder

.createDefault(featuresPath, abDecider, Some(statsReceiver))

.configRepoAbsPath(configRepoPath)

.forcedValues(getFeatureSwitchOverrides)

// Track stats when an experiment impression is made. For example:

// "experiment\_impressions/test\_experiment\_1234/"

// "experiment\_impressions/test\_experiment\_1234/control"

// "experiment\_impressions/test\_experiment\_1234/treatment"

.experimentImpressionStatsEnabled(true)

.unitsOfDiversionEnable(true)

val finalBuilder = if (isServiceLocal) {

baseBuilder

} else {

baseBuilder.serviceDetailsFromAurora()

}

finalBuilder.build()

}

}