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

import com.twitter.inject.annotations.Flags

import com.twitter.inject.Injector

import com.twitter.inject.TwitterModule

import com.twitter.util.Duration

object ProductMixerFlagModule extends TwitterModule {

final val ServiceLocal = "service.local"

final val ConfigRepoLocalPath = "configrepo.local\_path"

final val FeatureSwitchesPath = "feature\_switches.path"

final val StratoLocalRequestTimeout = "strato.local.request\_timeout"

final val ScribeABImpressions = "scribe.ab\_impressions"

final val PipelineExecutionLoggerAllowList = "pipeline\_execution\_logger.allow\_list"

flag[Boolean](

name = ServiceLocal,

default = false,

help = "Is the server running locally or in a DC")

flag[String](

name = ConfigRepoLocalPath,

default = System.getProperty("user.home") + "/workspace/config",

help = "Path to your local config repo"

)

flag[Boolean](

name = ScribeABImpressions,

help = "Enable scribing of AB impressions"

)

flag[String](

name = FeatureSwitchesPath,

help = "Path to your local config repo"

)

flag[Duration](

name = StratoLocalRequestTimeout,

help = "Override the request timeout for Strato when running locally"

)

flag[Seq[String]](

name = PipelineExecutionLoggerAllowList,

default = Seq.empty,

help =

"Specify user role(s) for which detailed log messages (containing PII) can be made. Accepts a single role or a comma separated list 'a,b,c'"

)

/\*\*

\* Invoked at the end of server startup.

\*

\* If we're running locally, we display a nice message and a link to the admin page

\*/

override def singletonPostWarmupComplete(injector: Injector): Unit = {

val isLocalService = injector.instance[Boolean](Flags.named(ServiceLocal))

if (isLocalService) {

// Extract service name from clientId since there isn't a specific flag for that

val clientId = injector.instance[String](Flags.named("thrift.clientId"))

val name = clientId.split("\\.")(0)

val adminPort = injector.instance[String](Flags.named("admin.port"))

val url = s"http://localhost$adminPort/"

// Print instead of log, so it goes on stdout and doesn't get the logging decorations.

// Update our local development recipe (local-development.rst) if making changes to this

// message.

println("===============================================================================")

println(s"Welcome to a Product Mixer Service, $name")

println(s"You can view the admin endpoint and thrift web forms at $url")

println("Looking for support? Have questions? #product-mixer on Slack.")

println("===============================================================================")

}

}

}