package com.twitter.tweetypie.backends

import com.twitter.configbus.client.ConfigbusClientException

import com.twitter.configbus.client.file.PollingConfigSourceBuilder

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.logging.Logger

import com.twitter.util.Activity

import com.twitter.util.Activity.\_

import com.twitter.conversions.DurationOps.\_

import com.twitter.io.Buf

trait ConfigBus {

def file(path: String): Activity[String]

}

object ConfigBus {

private[this] val basePath = "appservices/tweetypie"

private[this] val log = Logger(getClass)

def apply(stats: StatsReceiver, instanceId: Int, instanceCount: Int): ConfigBus = {

val client = PollingConfigSourceBuilder()

.statsReceiver(stats)

.pollPeriod(30.seconds)

.instanceId(instanceId)

.numberOfInstances(instanceCount)

.build()

val validBuffer = stats.counter("valid\_buffer")

def subscribe(path: String) =

client.subscribe(s"$basePath/$path").map(\_.configs).map {

case Buf.Utf8(string) =>

validBuffer.incr()

string

}

new ConfigBus {

def file(path: String): Activity[String] = {

val changes = subscribe(path).run.changes.dedupWith {

case (Failed(e1: ConfigbusClientException), Failed(e2: ConfigbusClientException)) =>

e1.getMessage == e2.getMessage

case other =>

false

}

Activity(changes)

}

}

}

}