package com.twitter.cr\_mixer.logging

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

import com.twitter.cr\_mixer.thriftscala.ImpressesedBucketInfo

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.util.StatsUtil

import com.twitter.logging.Logger

import com.twitter.scrooge.BinaryThriftStructSerializer

import com.twitter.scrooge.ThriftStruct

import com.twitter.scrooge.ThriftStructCodec

object ScribeLoggerUtils {

/\*\*

\* Handles base64-encoding, serialization, and publish.

\*/

private[logging] def publish[T <: ThriftStruct](

logger: Logger,

codec: ThriftStructCodec[T],

message: T

): Unit = {

logger.info(BinaryThriftStructSerializer(codec).toString(message))

}

private[logging] def getImpressedBuckets(

scopedStats: StatsReceiver

): Option[List[ImpressesedBucketInfo]] = {

StatsUtil.trackNonFutureBlockStats(scopedStats.scope("getImpressedBuckets")) {

CrMixerImpressedBuckets.getAllImpressedBuckets.map { listBuckets =>

val listBucketsSet = listBuckets.toSet

scopedStats.stat("impressed\_buckets").add(listBucketsSet.size)

listBucketsSet.map { bucket =>

ImpressesedBucketInfo(

experimentId = bucket.experiment.settings.experimentId.getOrElse(-1L),

bucketName = bucket.name,

version = bucket.experiment.settings.version,

)

}.toList

}

}

}

}