package com.twitter.search.earlybird.factory;

import com.twitter.decider.Decider;

import com.twitter.search.common.schema.earlybird.EarlybirdCluster;

import com.twitter.search.earlybird.EarlybirdIndexConfig;

import com.twitter.search.earlybird.RealtimeEarlybirdIndexConfig;

import com.twitter.search.earlybird.archive.ArchiveOnDiskEarlybirdIndexConfig;

import com.twitter.search.earlybird.archive.ArchiveSearchPartitionManager;

import com.twitter.search.earlybird.common.config.EarlybirdConfig;

import com.twitter.search.earlybird.exception.CriticalExceptionHandler;

import com.twitter.search.earlybird.partition.SearchIndexingMetricSet;

public final class EarlybirdIndexConfigUtil {

private EarlybirdIndexConfigUtil() {

}

/\*\*

\* Creates the index config for this earlybird.

\*/

public static EarlybirdIndexConfig createEarlybirdIndexConfig(

Decider decider, SearchIndexingMetricSet searchIndexingMetricSet,

CriticalExceptionHandler criticalExceptionHandler) {

if (isArchiveSearch()) {

return new ArchiveOnDiskEarlybirdIndexConfig(decider, searchIndexingMetricSet,

criticalExceptionHandler);

} else if (isProtectedSearch()) {

return new RealtimeEarlybirdIndexConfig(

EarlybirdCluster.PROTECTED, decider, searchIndexingMetricSet, criticalExceptionHandler);

} else if (isRealtimeCG()) {

return new RealtimeEarlybirdIndexConfig(

EarlybirdCluster.REALTIME\_CG, decider, searchIndexingMetricSet, criticalExceptionHandler);

} else {

return new RealtimeEarlybirdIndexConfig(

EarlybirdCluster.REALTIME, decider, searchIndexingMetricSet, criticalExceptionHandler);

}

}

public static boolean isArchiveSearch() {

// Re-reading config on each call so that tests can reliably overwrite this

return EarlybirdConfig.getString("partition\_manager", "realtime")

.equals(ArchiveSearchPartitionManager.CONFIG\_NAME);

}

private static boolean isProtectedSearch() {

// Re-reading config on each call so that tests can reliably overwrite this

return EarlybirdConfig.getBool("protected\_index", false);

}

private static boolean isRealtimeCG() {

// Re-reading config on each call so that tests can reliably overwrite this

return EarlybirdConfig.getBool("realtime\_cg\_index", false);

}

}