package com.twitter.search.earlybird.factory;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import com.twitter.search.earlybird.config.TierConfig;

import com.twitter.search.earlybird.config.TierInfo;

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

public final class PartitionConfigUtil {

private static final Logger LOG = LoggerFactory.getLogger(PartitionConfigUtil.class);

private PartitionConfigUtil() {

}

/\*\*

\* Initiate PartitionConfig for earlybirds running on Aurora

\*/

public static PartitionConfig initPartitionConfigForAurora(int numOfInstances) {

String tier = EarlybirdProperty.EARLYBIRD\_TIER.get();

int partitionId = EarlybirdProperty.PARTITION\_ID.get();

int replicaId = EarlybirdProperty.REPLICA\_ID.get();

if (tier.equals(PartitionConfig.DEFAULT\_TIER\_NAME)) {

// realtime or protected earlybird

return new PartitionConfig(

partitionId,

EarlybirdProperty.SERVING\_TIMESLICES.get(),

replicaId,

numOfInstances,

EarlybirdProperty.NUM\_PARTITIONS.get());

} else {

// archive earlybird

TierInfo tierInfo = TierConfig.getTierInfo(tier);

return new PartitionConfig(tier, tierInfo.getDataStartDate(), tierInfo.getDataEndDate(),

partitionId, tierInfo.getMaxTimeslices(), replicaId, numOfInstances,

tierInfo.getNumPartitions());

}

}

/\*\*

\* Tries to create a new PartitionConfig instance based on the Aurora flags

\*/

public static PartitionConfig initPartitionConfig() {

return initPartitionConfigForAurora(EarlybirdProperty.NUM\_INSTANCES.get());

}

}