package com.twitter.search.earlybird\_root;

import java.util.ArrayList;

import java.util.List;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import com.twitter.finagle.Service;

import com.twitter.finagle.SimpleFilter;

import com.twitter.search.earlybird.thrift.EarlybirdRequest;

import com.twitter.search.earlybird.thrift.EarlybirdResponse;

import com.twitter.search.earlybird\_root.common.EarlybirdRequestContext;

import com.twitter.search.earlybird\_root.common.EarlybirdRequestType;

import com.twitter.util.Future;

/\*\*

\* Filter that returns a PARTITION\_SKIPPED response instead of sending the request to a partition

\* if the partition PartitionAccessController says its disabled for a request.

\*/

public final class SkipPartitionFilter extends

SimpleFilter<EarlybirdRequestContext, EarlybirdResponse> {

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

private final String tierName;

private final int partitionNum;

private final PartitionAccessController controller;

private SkipPartitionFilter(String tierName, int partitionNum,

PartitionAccessController controller) {

this.tierName = tierName;

this.partitionNum = partitionNum;

this.controller = controller;

}

@Override

public Future<EarlybirdResponse> apply(

EarlybirdRequestContext requestContext,

Service<EarlybirdRequestContext, EarlybirdResponse> service) {

EarlybirdRequest request = requestContext.getRequest();

if (!controller.canAccessPartition(tierName, partitionNum, request.getClientId(),

EarlybirdRequestType.of(request))) {

return Future.value(EarlybirdServiceScatterGatherSupport.newEmptyResponse());

}

return service.apply(requestContext);

}

/\*\*

\* Wrap the services with a SkipPartitionFilter

\*/

public static List<Service<EarlybirdRequestContext, EarlybirdResponse>> wrapServices(

String tierName,

List<Service<EarlybirdRequestContext, EarlybirdResponse>> clients,

PartitionAccessController controller) {

LOG.info("Creating SkipPartitionFilters for cluster: {}, tier: {}, partitions 0-{}",

controller.getClusterName(), tierName, clients.size() - 1);

List<Service<EarlybirdRequestContext, EarlybirdResponse>> wrappedServices = new ArrayList<>();

for (int partitionNum = 0; partitionNum < clients.size(); partitionNum++) {

SkipPartitionFilter filter = new SkipPartitionFilter(tierName, partitionNum, controller);

wrappedServices.add(filter.andThen(clients.get(partitionNum)));

}

return wrappedServices;

}

}