package com.twitter.search.earlybird\_root;

import java.util.List;

import javax.inject.Inject;

import com.google.common.collect.Lists;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import com.twitter.finagle.Service;

import com.twitter.search.common.root.PartitionLoggingSupport;

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

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

import com.twitter.util.Future;

/\*\*

\* A chain of scatter gather services.

\* Regular roots use ScatterGatherService directly. This class is only used by multi-tier roots.

\*/

public class EarlybirdChainedScatterGatherService extends

Service<EarlybirdRequestContext, List<Future<EarlybirdResponse>>> {

private static final Logger LOG =

LoggerFactory.getLogger(EarlybirdChainedScatterGatherService.class);

private final List<Service<EarlybirdRequestContext, EarlybirdResponse>> serviceChain;

/\*\*

\* Construct a ScatterGatherServiceChain, by loading configurations from earlybird-tiers.yml.

\*/

@Inject

public EarlybirdChainedScatterGatherService(

EarlybirdServiceChainBuilder serviceChainBuilder,

EarlybirdServiceScatterGatherSupport scatterGatherSupport,

PartitionLoggingSupport<EarlybirdRequestContext> partitionLoggingSupport) {

serviceChain =

serviceChainBuilder.buildServiceChain(scatterGatherSupport, partitionLoggingSupport);

if (serviceChain.isEmpty()) {

LOG.error("At least one tier has to be enabled.");

throw new RuntimeException("Root does not work with all tiers disabled.");

}

}

@Override

public Future<List<Future<EarlybirdResponse>>> apply(EarlybirdRequestContext requestContext) {

// Hit all tiers in parallel.

List<Future<EarlybirdResponse>> resultList =

Lists.newArrayListWithCapacity(serviceChain.size());

for (final Service<EarlybirdRequestContext, EarlybirdResponse> service : serviceChain) {

resultList.add(service.apply(requestContext));

}

return Future.value(resultList);

}

}