package com.twitter.search.earlybird\_root;

import java.util.Map;

import javax.inject.Inject;

import javax.inject.Singleton;

import com.google.common.collect.ImmutableMap;

import com.google.common.collect.Maps;

import com.twitter.finagle.Service;

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

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

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

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

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

import com.twitter.search.earlybird\_root.routers.FacetsRequestRouter;

import com.twitter.search.earlybird\_root.routers.RecencyRequestRouter;

import com.twitter.search.earlybird\_root.routers.RelevanceRequestRouter;

import com.twitter.search.earlybird\_root.routers.RequestRouter;

import com.twitter.search.earlybird\_root.routers.TermStatsRequestRouter;

import com.twitter.search.earlybird\_root.routers.TopTweetsRequestRouter;

import com.twitter.util.Future;

@Singleton

public class SuperRootRequestTypeRouter

extends Service<EarlybirdRequestContext, EarlybirdResponse> {

private final Map<EarlybirdRequestType, RequestRouter> routingMap;

/\*\*

\* constructor

\*/

@Inject

public SuperRootRequestTypeRouter(

FacetsRequestRouter facetsRequestRouter,

TermStatsRequestRouter termStatsRequestRouter,

TopTweetsRequestRouter topTweetsRequestRouter,

RecencyRequestRouter recencyRequestRouter,

RelevanceRequestRouter relevanceRequestRouter

) {

routingMap = Maps.immutableEnumMap(

ImmutableMap.<EarlybirdRequestType, RequestRouter>builder()

.put(EarlybirdRequestType.FACETS, facetsRequestRouter)

.put(EarlybirdRequestType.TERM\_STATS, termStatsRequestRouter)

.put(EarlybirdRequestType.TOP\_TWEETS, topTweetsRequestRouter)

.put(EarlybirdRequestType.RECENCY, recencyRequestRouter)

.put(EarlybirdRequestType.STRICT\_RECENCY, recencyRequestRouter)

.put(EarlybirdRequestType.RELEVANCE, relevanceRequestRouter)

.build());

}

@Override

public Future<EarlybirdResponse> apply(EarlybirdRequestContext requestContext) {

EarlybirdRequest request = requestContext.getRequest();

if (request.getSearchQuery() == null) {

return Future.exception(new ClientErrorException(

"Client must fill in search Query object in request"));

}

EarlybirdRequestType requestType = requestContext.getEarlybirdRequestType();

if (routingMap.containsKey(requestType)) {

RequestRouter router = routingMap.get(requestType);

return router.route(requestContext);

} else {

return Future.exception(

new ClientErrorException(

"Request type " + requestType + " is unsupported. "

+ "Sorry this api is a bit hard to use.\n"

+ "for facets, call earlybirdRequest.setFacetsRequest\n"

+ "for termstats, call earluybirdRequest.setTermStatisticsRequest\n"

+ "for recency, strict recency, relevance or toptweets,\n"

+ " call req.setSearchQuery() and req.getSearchQuery().setRankingMode()\n"

+ " with the correct ranking mode and for strict recency call\n"

+ " earlybirdRequest.setQuerySource(ThriftQuerySource.GNIP)\n"));

}

}

}