package com.twitter.search.earlybird\_root.filters;

import com.twitter.finagle.Service;

import com.twitter.finagle.SimpleFilter;

import com.twitter.search.common.metrics.SearchCounter;

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

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

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

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

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

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

import com.twitter.util.Function;

import com.twitter.util.Future;

public class MarkTweetSourceFilter

extends SimpleFilter<EarlybirdRequestContext, EarlybirdResponse> {

private final SearchCounter searchResultsNotSet;

private final ThriftTweetSource tweetSource;

public MarkTweetSourceFilter(ThriftTweetSource tweetSource) {

this.tweetSource = tweetSource;

searchResultsNotSet = SearchCounter.export(

tweetSource.name().toLowerCase() + "\_mark\_tweet\_source\_filter\_search\_results\_not\_set");

}

@Override

public Future<EarlybirdResponse> apply(

final EarlybirdRequestContext requestContext,

Service<EarlybirdRequestContext, EarlybirdResponse> service) {

return service.apply(requestContext).map(new Function<EarlybirdResponse, EarlybirdResponse>() {

@Override

public EarlybirdResponse apply(EarlybirdResponse response) {

if (response.getResponseCode() == EarlybirdResponseCode.SUCCESS

&& requestContext.getEarlybirdRequestType() != EarlybirdRequestType.TERM\_STATS) {

if (!response.isSetSearchResults()) {

searchResultsNotSet.increment();

} else {

for (ThriftSearchResult searchResult : response.getSearchResults().getResults()) {

searchResult.setTweetSource(tweetSource);

}

}

}

return response;

}

}

);

}

}