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

import java.util.Map;

import com.google.common.collect.Maps;

import com.twitter.finagle.Service;

import com.twitter.finagle.SimpleFilter;

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

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

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

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

import com.twitter.util.Future;

import com.twitter.util.FutureEventListener;

public class ResponseCodeStatFilter

extends SimpleFilter<EarlybirdRequest, EarlybirdResponse> {

private final Map<EarlybirdResponseCode, SearchCounter> responseCodeCounters;

/\*\*

\* Create ResponseCodeStatFilter

\*/

public ResponseCodeStatFilter() {

responseCodeCounters = Maps.newEnumMap(EarlybirdResponseCode.class);

for (EarlybirdResponseCode code : EarlybirdResponseCode.values()) {

SearchCounter stat = SearchCounter.export("response\_code\_" + code.name().toLowerCase());

responseCodeCounters.put(code, stat);

}

}

@Override

public Future<EarlybirdResponse> apply(

final EarlybirdRequest request,

final Service<EarlybirdRequest, EarlybirdResponse> service) {

return service.apply(request).addEventListener(

new FutureEventListener<EarlybirdResponse>() {

@Override

public void onSuccess(final EarlybirdResponse response) {

responseCodeCounters.get(response.getResponseCode()).increment();

}

@Override

public void onFailure(final Throwable cause) { }

});

}

}