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

import java.util.Optional;

import javax.inject.Inject;

import com.google.common.base.Preconditions;

import com.twitter.finagle.Service;

import com.twitter.finagle.SimpleFilter;

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

import com.twitter.search.earlybird.common.ClientIdUtil;

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

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

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

import com.twitter.search.earlybird\_root.quota.ClientIdQuotaManager;

import com.twitter.search.earlybird\_root.quota.QuotaInfo;

import com.twitter.util.Future;

public class ClientIdArchiveAccessFilter extends SimpleFilter<EarlybirdRequest, EarlybirdResponse> {

private static final String UNAUTHORIZED\_ARCHIVE\_ACCESS\_COUNTER\_PATTERN =

"unauthorized\_access\_to\_full\_archive\_by\_client\_%s";

private final ClientIdQuotaManager quotaManager;

/\*\*

\* Construct the filter by using ClientIdQuotaManager

\*/

@Inject

public ClientIdArchiveAccessFilter(ClientIdQuotaManager quotaManager) {

this.quotaManager = Preconditions.checkNotNull(quotaManager);

}

@Override

public Future<EarlybirdResponse> apply(EarlybirdRequest request,

Service<EarlybirdRequest, EarlybirdResponse> service) {

String clientId = ClientIdUtil.getClientIdFromRequest(request);

Optional<QuotaInfo> quotaInfoOptional = quotaManager.getQuotaForClient(clientId);

QuotaInfo quotaInfo = quotaInfoOptional.orElseGet(quotaManager::getCommonPoolQuota);

if (!quotaInfo.hasArchiveAccess() && request.isGetOlderResults()) {

SearchCounter unauthorizedArchiveAccessCounter = SearchCounter.export(

String.format(UNAUTHORIZED\_ARCHIVE\_ACCESS\_COUNTER\_PATTERN, clientId));

unauthorizedArchiveAccessCounter.increment();

String message = String.format(

"Client %s is not whitelisted for archive access. Request access at go/searchquota.",

clientId);

EarlybirdResponse response = new EarlybirdResponse(

EarlybirdResponseCode.QUOTA\_EXCEEDED\_ERROR, 0)

.setDebugString(message);

return Future.value(response);

}

return service.apply(request);

}

}