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

import com.google.common.base.Optional;

import com.twitter.search.common.caching.CacheUtil;

import com.twitter.search.common.caching.SearchQueryNormalizer;

import com.twitter.search.common.caching.filter.CacheRequestNormalizer;

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

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

public class RelevanceZeroResultsCacheRequestNormalizer

extends CacheRequestNormalizer<EarlybirdRequestContext, EarlybirdRequest> {

@Override

public Optional<EarlybirdRequest> normalizeRequest(EarlybirdRequestContext requestContext) {

// If the query is not personalized, it means that:

// - RelevanceCacheRequestNormalizer has already normalized it into a cacheable query.

// - RelevanceCacheFilter could not find a response for this query in the cache.

//

// So if we try to normalize it here again, we will succeed, but then

// RelevanceZeroResultsCacheFilter will do the same look up in the cache, which will again

// result in a cache miss. There is no need to do this look up twice, so if the query is not

// personalized, return Optional.absent().

//

// If the query is personalized, strip all personalization fields and normalize the request.

if (!SearchQueryNormalizer.queryIsPersonalized(requestContext.getRequest().getSearchQuery())) {

return Optional.absent();

}

return Optional.fromNullable(

CacheUtil.normalizeRequestForCache(requestContext.getRequest(), true));

}

}