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

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

import com.twitter.search.common.decider.SearchDecider;

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

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

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

public class RelevanceZeroResultsQueryCachePredicate

extends QueryCachePredicate<EarlybirdRequestContext> {

private final SearchDecider decider;

private final String relevanceCacheEnabledDeciderKey;

private final String relevanceZeroResultsCacheEnabledDeciderKey;

public RelevanceZeroResultsQueryCachePredicate(

SearchDecider decider, String normalizedSearchRootName) {

this.decider = decider;

this.relevanceCacheEnabledDeciderKey =

"relevance\_cache\_enabled\_" + normalizedSearchRootName;

this.relevanceZeroResultsCacheEnabledDeciderKey =

"relevance\_zero\_results\_cache\_enabled\_" + normalizedSearchRootName;

}

@Override

public Boolean shouldQueryCache(EarlybirdRequestContext requestContext) {

return EarlybirdRequestType.RELEVANCE == requestContext.getEarlybirdRequestType()

&& EarlybirdRequestUtil.isCachingAllowed(requestContext.getRequest())

&& decider.isAvailable(relevanceCacheEnabledDeciderKey)

&& decider.isAvailable(relevanceZeroResultsCacheEnabledDeciderKey);

}

}