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

import javax.inject.Inject;

import javax.inject.Named;

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

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

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

import com.twitter.search.common.root.SearchRootModule;

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

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

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

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

public class StrictRecencyCacheFilter extends

CacheFilter<EarlybirdRequestContext, EarlybirdRequest, EarlybirdResponse> {

/\*\*

\* Creates a cache filter for earlybird strict recency requests.

\*/

@Inject

public StrictRecencyCacheFilter(

@StrictRecencyCache Cache<EarlybirdRequest, EarlybirdResponse> cache,

SearchDecider decider,

@Named(SearchRootModule.NAMED\_NORMALIZED\_SEARCH\_ROOT\_NAME) String normalizedSearchRootName,

@Named(CacheCommonUtil.NAMED\_MAX\_CACHE\_RESULTS) int maxCacheResults) {

super(cache,

new StrictRecencyQueryCachePredicate(decider, normalizedSearchRootName),

new RecencyCacheRequestNormalizer(),

new RecencyAndRelevanceCachePostProcessor(),

new RecencyServicePostProcessor(cache, maxCacheResults),

new EarlybirdRequestPerClientCacheStats(

EarlybirdRequestType.STRICT\_RECENCY.getNormalizedName()));

}

}