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

import javax.inject.Named;

import javax.inject.Singleton;

import com.google.inject.Provides;

import com.twitter.inject.TwitterModule;

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

import com.twitter.search.earlybird\_root.filters.EarlybirdTimeRangeFilter;

import com.twitter.search.earlybird\_root.filters.FullArchiveServingRangeProvider;

import com.twitter.search.earlybird\_root.filters.RealtimeServingRangeProvider;

import com.twitter.search.earlybird\_root.filters.ServingRangeProvider;

public class TermStatsRequestRouterModule extends TwitterModule {

public static final String FULL\_ARCHIVE\_TIME\_RANGE\_FILTER =

"term\_stats\_full\_archive\_time\_range\_filter";

public static final String REALTIME\_TIME\_RANGE\_FILTER =

"term\_stats\_realtime\_time\_range\_filter";

private static final String SUPERROOT\_TERM\_STATS\_SERVING\_RANGE\_BOUNDARY\_HOURS\_AGO\_DECIDER\_KEY =

"superroot\_term\_stats\_serving\_range\_boundary\_hours\_ago";

private ServingRangeProvider getFullArchiveTimeRangeProvider(final SearchDecider decider)

throws Exception {

return new FullArchiveServingRangeProvider(

decider, SUPERROOT\_TERM\_STATS\_SERVING\_RANGE\_BOUNDARY\_HOURS\_AGO\_DECIDER\_KEY);

}

private ServingRangeProvider getRealtimeTimeRangeProvider(final SearchDecider decider)

throws Exception {

return new RealtimeServingRangeProvider(

decider, SUPERROOT\_TERM\_STATS\_SERVING\_RANGE\_BOUNDARY\_HOURS\_AGO\_DECIDER\_KEY);

}

/\*\*

\* For term stats full archive cluster spans from 21 March to 2006 to 6 days ago from current time

\*/

@Provides

@Singleton

@Named(FULL\_ARCHIVE\_TIME\_RANGE\_FILTER)

private EarlybirdTimeRangeFilter providesFullArchiveTimeRangeFilter(final SearchDecider decider)

throws Exception {

return EarlybirdTimeRangeFilter.newTimeRangeFilterWithQueryRewriter(

getFullArchiveTimeRangeProvider(decider), decider);

}

/\*\*

\* For term stats realtime cluster spans from 6 days ago from current time to a far away date

\* into the future

\*/

@Provides

@Singleton

@Named(REALTIME\_TIME\_RANGE\_FILTER)

private EarlybirdTimeRangeFilter providesRealtimeTimeRangeFilter(final SearchDecider decider)

throws Exception {

return EarlybirdTimeRangeFilter.newTimeRangeFilterWithQueryRewriter(

getRealtimeTimeRangeProvider(decider), decider);

}

}