package com.twitter.timelineranker.in\_network\_tweets

import com.twitter.timelineranker.model.CandidateTweetsResult

import com.twitter.timelineranker.model.RecapQuery

import com.twitter.timelineranker.model.RecapQuery.DependencyProvider

import com.twitter.timelineranker.parameters.in\_network\_tweets.InNetworkTweetParams

import com.twitter.util.Future

/\*\*

\* A repository of in-network tweet candidates.

\* For now, it does not cache any results therefore forwards all calls to the underlying source.

\*/

class InNetworkTweetRepository(

source: InNetworkTweetSource,

realtimeCGSource: InNetworkTweetSource) {

private[this] val enableRealtimeCGProvider =

DependencyProvider.from(InNetworkTweetParams.EnableEarlybirdRealtimeCgMigrationParam)

def get(query: RecapQuery): Future[CandidateTweetsResult] = {

if (enableRealtimeCGProvider(query)) {

realtimeCGSource.get(query)

} else {

source.get(query)

}

}

def get(queries: Seq[RecapQuery]): Future[Seq[CandidateTweetsResult]] = {

Future.collect(queries.map(query => get(query)))

}

}