

CS172 Project: Twitter Tweet Tweet Crawler

Collaboration Report

Collaboration Details:

Team Members:

Michael Villanueva	mvill025	861074235
Kevin Hsieh	khsie003	861054367.

All the Work was done together meaning, we collaborated on all parts of the project. Because we only had one file to work on, we took turns coding on the Eclipse IDE. While one person was working on the code, the other person did research to figure out the other parts of structure of the project and the implementation of it.

Overview of System:

Architecture:

The structure the Twitter Tweet Tweet Crawler(TTTC) is very simple. Surprisingly, the source code of the TTTC exists as a single file named main.java. What enables the TTTC to have such simple implementation is its use of various API's. These API's allow a very straightforward, but effective structure to the Program. Twitter4j is used to get Tweets using twitterstream and a listener. TTTC sets the onStatus handler to write a JSON formatted string into a hashset - to prevent duplication of Tweets. After the hashset reaches the threshold of maximum Tweets, execution parameter 1, its will then stop the listener and write all JSON strings from the hashset as JSON objects into the output directory into an input file.

Data Collection Strategy

Twitter4j and the Twitter Streaming API was used to stream live Tweets. We set an array of queries from the execution parameter, args[2]. Since the input queries are separated by a ',' we parsed each query into a String array. The onStatus function will be called when there is a matching query in the Tweet. During the onStatus call, it creates a thread that

writes subfields of the Status object into a JSON object. After all subfields are gathered, we add the JSON object into the hashmap.

Data Structures Employed

As stated above, we used a hashset to store the twitter stream in JSON format. Each JSON string have the following fields: User, TimeStamp, Content, LinkUrl, LinkTitle, GeoLocation. If the Tweet we gathered does not have a link or GeoLocation information, these fields would store a null value.

Limitations:

Single Stream

Given that the Twitter API only lets you listen to one stream per IP address, we are limited to one stream in TTTC. Because of this limitation, we only used one listener to listen to the Tweet stream for a query included Tweet.

Possible Duplication

There may be possible duplication of Tweet entries from spam bot accounts. If two tweets with the same content and different Tweet id or user are collected, the hashset will still record both Tweets as different entries. However the content will be extremely similar.

No Ranking

Similar to the Possible Duplication problem. Since Ranking is not implemented, we will not have a variety of Tweets due to Tweet content similarities.

Instruction on how to deploy the system:

Parameters:

There should be three parameters followed by the executable:

- 1) Maximum Tweets to stream.
- 2) A line of queries, separated by a signal comma.
- 3) outputdirectory.

i.e. [user@server]./TweetCrawler <TweetNum> <q1,q2,q3,...>

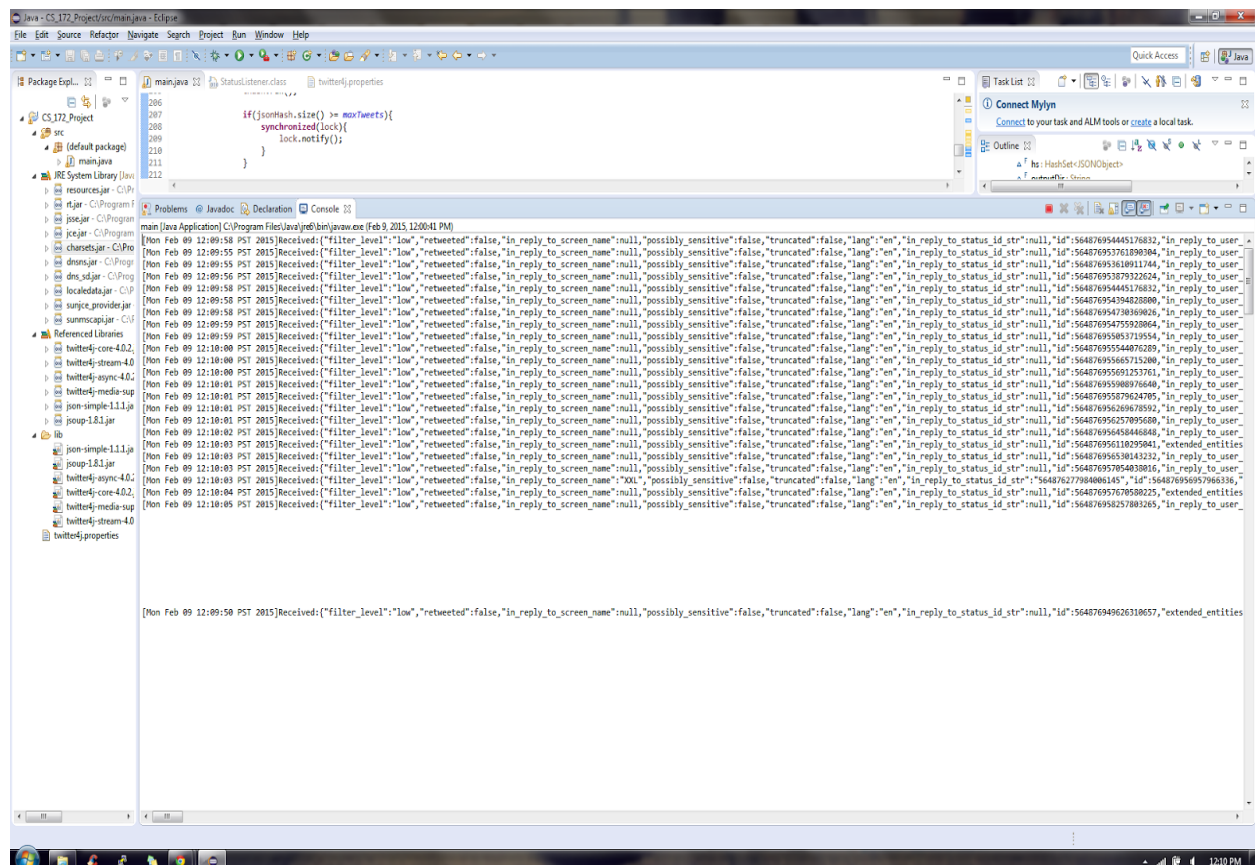
<out_dir>

With love and Internet. <3

Make sure that when you run the executable with the parameters that the program has access to the internet.

Screenshots showing the system in action:

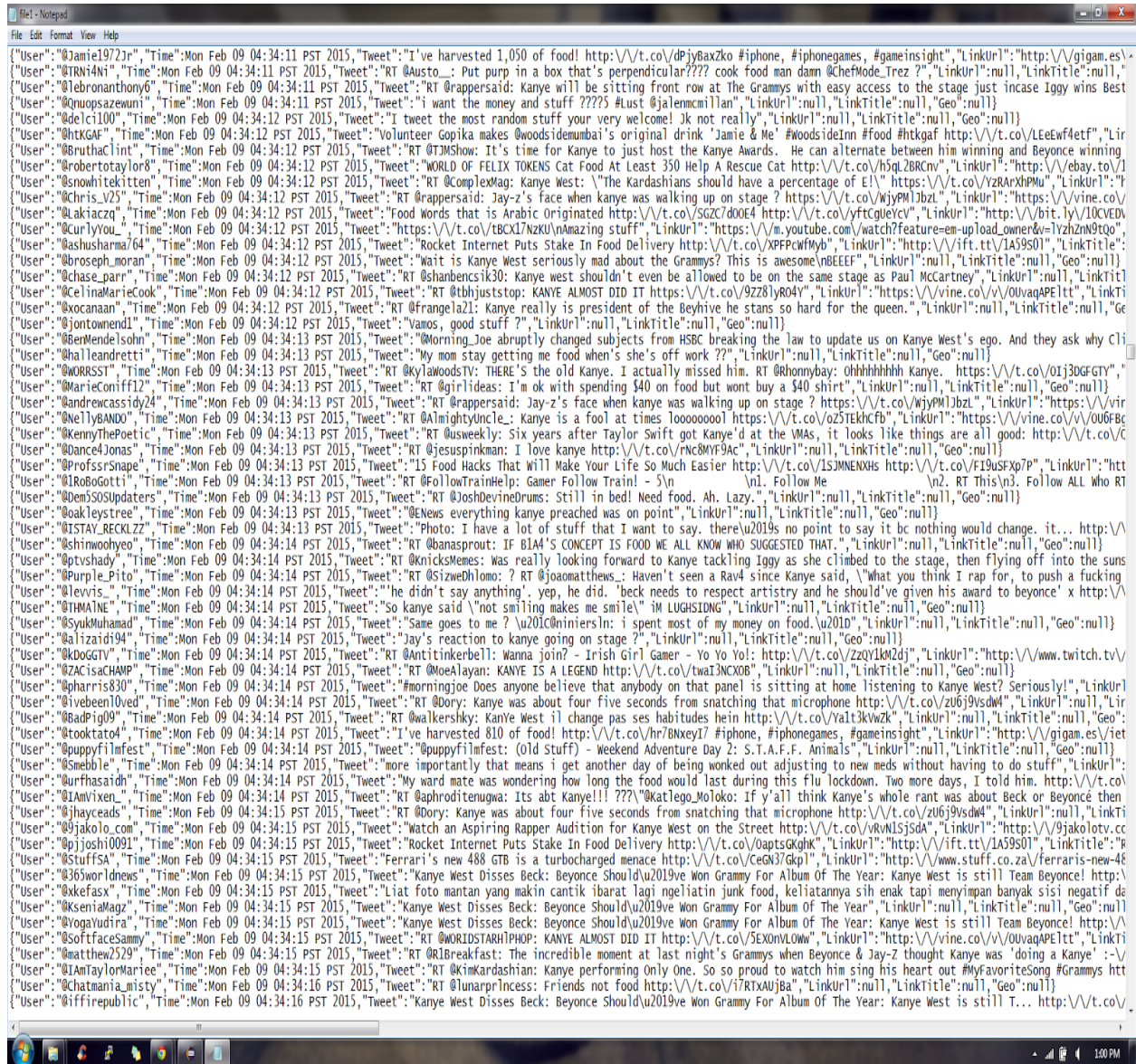
Streaming Tweets:



Example of one output:

```
{"User": "@RickyMak", "Time": "Mon Feb 09 11:59:51 PST 2015", "Tweet": "Good stuff. http://t.co/CiqvD2U5kM", "LinkUrl": "http://fb.me/6m4HftB4H", "LinkTitle": "» What u2019s Up Now", "Geo": null}
```

Example of file output:



The screenshot shows a Notepad window with a large list of JSON objects, each representing a tweet. The objects are formatted as follows:

```
{
  "User": "@User",
  "Time": "Mon Feb 09 04:34:11 PST 2015",
  "Tweet": "I've harvested 1,050 of food! http://t.co/dpY8axZko #iphone, #iphonegames, #gameinsight",
  "LinkUrl": "http://t.co/gigam.es",
  "LinkTitle": null,
  "Geo": null
}
```

The list continues with many more tweets, including mentions of Kanye West, Beyoncé, and various other users. The tweets are timestamped as of February 9, 2015, at 04:34:11 PST. The JSON objects are separated by newlines, and the entire list is contained within a single text file.