

Can You Say that Again?

A Real-World API and Data Exercise

Ria Galanos, Thomas Jefferson High School for Science and Technology

At this point, you are familiar with both the Java API and the Karel API. The Java API is the set of classes that are included with the Java Development Environment. The Karel API included all the classes needed to work with the Karel Robot environment. You may also be familiar with other APIs depending on where you learned to program in Java.

So, what's an API anyway?

API

Application Program Interface: Specifying how software components interact, APIs allow different applications to communicate with each other. This enables programmers to give an instruction to one app and receive information from another in return, creating interactive and dynamic content for their website.

Read more about APIs and why they are so useful at: <http://readwrite.com/2013/09/19/api-defined>. After you learn how to work with an industry-level API in this assignment, you'll be ready to tinker with any of the *50 Most Useful APIs for Developers* described here: <http://www.computersciencezone.org/50-most-useful-apis-for-developers/>

In this lab, we are going to use an API which will allow us to interact with Twitter. With just a few lines of code, we can access our personal tweets as well as the public tweets of others. And we can do much more!

Part I – Connect and Tweet

We will use the Twitter4j API which can be found at <http://twitter4j.org>. Click on JavaDoc.

There are a few housekeeping items to be tackled before you can use Twitter from your program. Steps 1-3 are required and steps 4-10 will make your life with this assignment much, much easier.

- 1) **Copy** the Twitter-Students folder from the R: drive.
- 2) Once the folder is on your personal drive, you'll notice a `lib` folder containing a `twitter4j-core-4.0.4.jar` file. You will need to set the **classpath** in *jGRASP* to `lib\twitter4j-core-4.0.4.jar`. Do this in *jGRASP* by:
 - a. On the menu toolbar, select Settings → PATH / CLASSPATH → Workspace.
 - b. Select the CLASSPATHS tab, then select New.
 - c. Click the Browse button next to the field labeled Directory Path or JAR File.
 - d. Browse to the location of where you saved the twitter4j-4.0.4 folder.

- e. Click on the Lib folder, then select the `twitter4j-core-4.0.4.jar` file. Select choose. Select OK. Select Apply. Select OK.
- 3) You should also notice a `twitter4j.properties` **file**. This file holds the credentials in order for you to access Twitter data. Twitter requires permission to use their API. Here is a NY Times article where Twitter revoked and then reinstated a company's access to the API: [In Reversal, Twitter to Let Politwoops Collect Politicians' Deleted Tweets](#).
- 4) The `twitter4j.properties` file currently holds some default values which grant you read-only access to Twitter. While this is enough for you to complete the assignment, it does not allow for multiple students to use the credentials at the same time. You may encounter a run-time error that states there is a rate limit issue if you using the credentials at the same time as other students. (This will probably work fine if you are coding at 5am.) To avoid this error, you should consider making your own Twitter account.
- 5) To make your own Twitter account, visit www.twitter.com and **sign up**. While you have to provide a valid email address as you will need to use it for verification, feel free to use a non-real username. You must add a mobile phone number to your Twitter profile in order to make a developer account later. You can also delete the account once the assignment is over, if desired.
- 6) Now that you have a Twitter account, you'll need to **visit** <http://apps.twitter.com> to set up the permissions to allow a non-Twitter app (your Java program) to access Twitter.
- 7) Select **Create New App** and fill in the first three fields of Application Details:
 - a. Name: <Name your project> perhaps something like APCS Twitter Project 2016
 - b. Description: <Anything> perhaps something like Learning to Work with APIs
 - c. Website: <Anything you want that starts with an http://> perhaps <http://www.tjhsst.edu>
- 8) Agree to the **Developer Agreement**. Click the Create Your Twitter Application button. Note: you will get a warning here if you did not add a mobile number to your Twitter account. Log back into Twitter to add a mobile number under Settings → Mobile.
- 9) If you want to be able to tweet out from your Java program, then visit the **Permissions tab** and select Read and Write. Leave Read Only if you do not wish to have write access. You will probably have to regenerate the four necessary keys each time you change permissions.
- 10) Retrieve the **four keys** you will need for the `twitter4j.properties` file and edit that file:
 - a. Go to My Applications.
 - b. Click on the application you made.
 - c. Go to the Keys and Access Tokens tab.
 - d. At the bottom, click on Create My Access Token.
 - e. Copy the four needed keys from the Twitter webpage: Consumer Key (API Key), Consumer Secret (API Secret), Access Token, and Access Token Secret. Paste the four new keys into your `twitter4j.properties` file in your folder.

```
oauth.consumerKey=  
oauth.consumerSecret=  
oauth.accessToken=  
oauth.accessTokenSecret=
```

- 11) Now, the fun begins! Let's see if you can tweet from your Java program.
 - a. Type this code into your `Twitter_Driver` in the `tweetOut()` method:
`twitter.updateStatus(message);`
 - b. In the main method of `Twitter_Driver`:
 - i. Uncomment the line `bigBird.tweetOut(message);`
 - ii. Change message to be the string you wish to tweet. For your first one, consider giving out a shout out to the APCS A course, our school, and Ms. Galanos for teaching you how to tweet from your java program. Example: ("I just tweeted from my Java program! #APCSARocks @TJColonials Thanks @cscheerleader!");
 - c. Compile and run your program.
 - d. Visit your Twitter web page – you should be able to see your tweet.
 - e. Re-comment the line `bigBird.tweetOut(message);` so that you don't keep sending out the same tweet over and over while testing your program.

Part II – Most Common Word

You will access the last 2,000 tweets of a person's twitter handle (this can be your own, someone you know, or a public figure) and determine the most common word they use. One procedure to do this is:

- Get the most recent 2,000 tweets and store in a data structure (think about the ones you've learned so far).
- Split the tweets into individual words (blank delimited) and store the words in a data structure.
- Remove extraneous punctuation characters from words.
- Remove common and uninteresting words such as "the", "a", & "and".
- Find the word that occurs most often in the data structure.

In `TJ_Twitter`, you will need to implement the following methods:

- a. `void splitIntoWords()`

This method takes all the tweets in `statuses` and splits them up into individual words, storing them in `terms`.

- b. `String removePunctuation(String s)`

It is up to you to decide the specific punctuation to remove. As you run the program, you may decide to remove different punctuation symbols. You may find that people use extra spacing in their tweets and may want to remove those also.

- c. `void removeCommonEnglishWords()`

The list of common words can be found in `commonWords.txt`. Use the list to compare against terms and remove any terms that appear in `commonWords.txt`. If you wish to add to or change the list of common words, then you should edit `commonWords.txt`.

- d. `void sortAndRemoveEmpties()`

Use sorting code you implemented earlier this year. Do not use any built in sorting algorithms, you should only use code you wrote yourself. Since some people actually include extra spaces in their tweets, you may wish to remove terms that are just empty strings.

e. `String mostPopularWord()`

An efficient implementation of this method would be to find the word that is in the largest consecutive group of equal words. There are alternate and less efficient algorithms. Once you find the most common word, populate the `frequencyMax` variable with the number of times the most common word occurred.

f. `int getWordFrequency()`

An accessor method to report the number of occurrences of the most common word.

After testing and implementing the `TJ_Twitter` methods, uncomment the Part 2 block of code in the main method of `Twitter_Driver`. Compile, run, and test this main method.

Sample run

```
Please enter a Twitter handle, do not include the @ symbol --> taylorswift13
The most common word from @taylorswift13 is: love.
The word appears 149 times.
```

```
Please enter a Twitter handle, do not include the @ symbol --> billnye
The most common word from @billnye is: science.
The word appears 220 times.
```

```
Please enter a Twitter handle, do not include the @ symbol --> nasa
The most common word from @nasa is: space.
The word appears 208 times.
```

```
Please enter a Twitter handle, do not include the @ symbol --> tjcolonials
The most common word from @tjcolonials is: tj.
The word appears 621 times.
```

```
Enter Twitter handle (without the @ symbol, type done to quit) --> done
```

Part III

Team up with a partner and brainstorm something interesting you can do with Twitter. You have the JavaDocs as a resource (<http://twitter4j.org/oldjavadocs/4.0.4/index.html>) and you can find some sample code at <http://twitter4j.org/en/code-examples.html>.

Here are some ideas:

- 1) Filter your home timeline (these are all the tweets from people you follow) to avoid a particular term. For instance, if you are an Android fan, you could filter out all tweets with Apple in them.
- 2) Count the number of tweets originating from your zip code that contain the word "flu".
- 3) Send tweets every 10 minutes to your friend on his/her birthday. Use a timer, of course.
- 4) Search for tweets: (a) created in the last week, (b) that contained the term "dolphins" (aka Miami Dolphins), and (c) were sent from a location within a 10 mile radius of Ms. Galanos' house. Her sample code reports the time, username, and the actual tweet. The sample code for this can be found in the `sampleInvestigate` method.