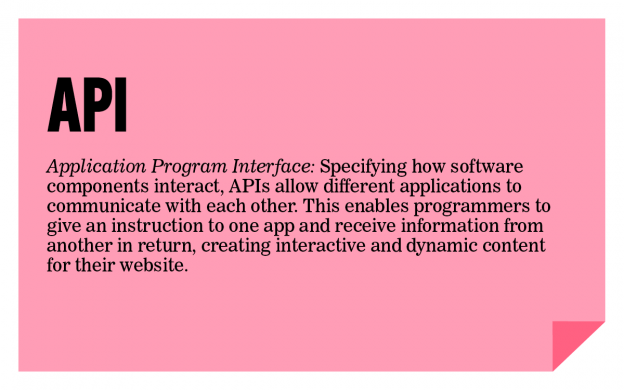
Can You Say that Again?

A Real-World API and Data Exercise

At this point, you are familiar with the Java API and the Lejos API. The Java API is the set of classes that are included with the Java Development Environment. The Lejos API included all the classes needed to work in the Lejos environment.

So, what’s an API anyway?



Read more about APIs and why they are so useful at: <http://readwrite.com/2013/09/19/api-defined>.

We are going to use an API which allows 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 decided to 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, and 10 are required and steps 4-9 will make your life with this assignment much, much easier.

1. **Copy** the Twitter-students folder from Canvas.
2. Once the folder is on your personal drive, you’ll notice a lib folder containing a twitter4j-core-4.0.2.jar file. You will need to add that .jar to your Eclipse project
3. You should also notice a **twitter4jproperties file**. This file holds the credentials in order for you to access Twitter data. The file currently holds some default values which grant you read-only access to Twitter. This file needs to be placed in the src folder. 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 are 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.
4. To make your own Twitter account, visit [www.twitter.com](http://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.
5. 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.
6. Select **Create New App** and fill in the first three fields of Application Details:
   1. Name: <Name your project>
   2. Description: Learning to Work with APIs
   3. Website: <Anything you want that starts with an http://>
7. Agree to the **Developer Agreement**.
8. 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.
9. Retrieve the **four keys** you will need for the twitter4j.properties file and edit that file:
10. Go to My Applications
11. Click on the application you made
12. Go to Keys and Access Tokens
13. Click on Create My Access Token
14. Get the four keys you need. Note: the Consumer Key is the API key and the ConsumerSecret is the APISecret. Update the properties file in your folder with your specific data.

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

1. Let's see if you can tweet from your Java program! Add this code into your sendTweet method:

chatbotTwitter.updateStatus("I just tweeted from my Java Chatbot program! #APCSRocks @CTECNow Thanks @cscheerleader & @codyhenrichsen! <insert your name>");

**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. To do this, you will need to grab a user’s most recent 2,000 tweets, store individual words in a data structure of your choice, and then write an algorithm to determine the most common term. Since it would not be interesting to count “the”, “a”, “and”, or similarly all-too common words, you need to remove those too-common words. Fortunately, we put those too-common words in the file commonWords.txt. This file should also be placed in the model package. Remove those too-common words from the tweeter’s list of words before determining the tweeter’s most commonly used word. Note: you will need multiple data structures for this assignment – at least one must be an ArrayList.