

# Java-Final-Project

# This is where we can document how to use our project.

#### Twitter API Project Usage:

This project is meant to be used as a Server Application.

The Application will collect and cluster tweet information in the tweeted posts received from the Twitter API.

To run the program:

With war file we provided: (javaFinalProject.war in the code folder)

1-Open a command prompt in the same directory as the war file

2-Run the following command: java -jar javaFinalProject.war

3-You should see some output in the command prompt as the application should be starting up.

When you see "Started Application in xx.xx seconds" the application is running and ready.

To download a copy of the project to run in an IDE:

1- To launch the server you need Java and Maven Installed on the machine and added to the System Variables.

2- You need to have git installed or have a zip file of the project.

Steps for git:

a- Create a Directory

b- Execute command 'git init' in that directory.

c- Execute command 'git clone https://github.com/kd7wfp/Java-Final-Project.git'

Run Maven to install all dependencies.

3- You can now compile and run the project from Application.java.

The Spring server will start and be ready for input.

#### Graphical User Interface Documentation:

1- In your browser (preferably Google Chrome :) )navigate to localhost:8080 to bring up the main page.

2- You can now select one of the 5 categories of tweets.

a- Microsoft, Apple, Nike, Orem, Amazon.

3- A word cloud appears of common words in the received tweets.

4- Select one of the words in the cloud.

a- A display of all the tweets pertaining to the word you clicked will be displayed below.

#### End Point Documentation:

1- localhost:8080/twitter/getTweets?tweet=<Enter your Tweet Here>

This will retrieve the tweets and place them in raw form into the browser display.

a- Typically not used by the consumer. We use this internally to gather the tweets, or in development.

#### Basic Architecture:

1- Application.java contains the starting point for the program. From the main method in Application.java the Spring

Boot web service is started and ready to respond.

2- The UI which is powered by the files located under /webapp. All these .js files handle the different functions of the application.

3- Upon UI click events the browser will communicate to the java project under /java/com/twitter.

4- Based on the request from the click event the Twitter\*.java classes will crunch the information and hand it back to the browser to be formatted by the .css and .html files.