Twitter Boost App (TBA)

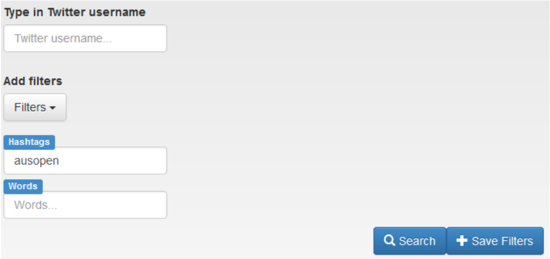
Livia Moroianu, Adrian Zamfirescu
Advanced Software Services
University Politehnica of Bucharest
Email: livia.moroianu@gmail.com, zamfirescuan@yahoo.com

***Abstract*—In a world where social media becomes a second reality, the need of staying in touch with online news and information of personal interest is becoming essential. Twitter is an online platform that fulfills this need, but the extremely large flow on information is a little bit too noisy and becomes somehow overwhelming for the user, who may just want to keep track and record of particular pieces of information regarding particular users.**

I. Introduction

Twitter Boost App (TBA) is intended to be a web application that integrates the Twitter API for access to Twitter information without requiring them to register for a Twitter account.

The Twitter information retrieval is made in a personalized manner, offering users a customizable interrogation.



ausopen

II. Achitecture

A. Tweet Filtering

The application provides a way for users to add links to any Twitter channels of interest and also advanced filters for the tweets displayed in the interface. The following filters are available in the interface:

1) HASHTAGS
2) REFERENCES
3) WORDS
4) TWITTER USER

B. Sentiment Analysis

Also knows as Opinion mining refers to the process of extracting subjective information from text or oral communication. Its goal is to determine the attitude of a speaker of a writer regarding some topic or in general. The result of this process consists of assigning a polarity to a piece of writing. A possible classification for text is the following: positive, negative, neutral;

More complex sentiment analysis systems aim to classify the polarity of a text at a higher granularity, considering also emotional states like: “angry”, “sad”, “happy”.

This project also uses sentiment analysis by integrating a specialized API, named “Datumbox”, that is able to return a possible classification for the polarity of a tweet text. It uses a machine learning algorithm trained with data sets consisting only of tweet texts.

III. Technologies

A. Spring

The Spring Framework lies at the root of the backend architecture and configuration of the application. Its APIs offer a rich high-level Java development environment that allows a large area of server issues to be handled.

The current project uses Spring for configuring the Model-View-Controller(MVC) architecture, establishing and using the connection to the database(a module that integrates with the Object Relational Mapping framewok Hibernate), integrating the security components (authentication, authorization) and solving tipical security issues in a web application.

B. Angular

Our web application consists mainly of two big modules: the backend and the frontend. Earlier, we mentioned about Spring taking care of the necessary flows that manage the transportation and manipulation of the application’s data. The other essential framework that links components, but on the frontend side, is called Angular JS - a MVC oriented JavaScript framework.

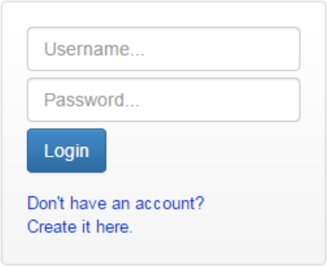
Angular does a couple of important things. Firstly, it creates a bridge between the backend and frontend modules, by assuring the HTTP communication through its AJAX based services. The user’s registration data, the login sensitive form credentials, the tweets sent by the server towards the client for display, everything is transported by calls of HTTP specific methods, managed by Angular.As a consequence, the web application is REST oriented and is primarily based on JSON objects as a way of data encapsulation.

What really creates the interactivity of the application’s pages is the Angular’s ability of dynamically generating the HTML DOM, given the logic written in code. We are actually talking about a single-page application, once the user gets authenticated.

IV. Implementation

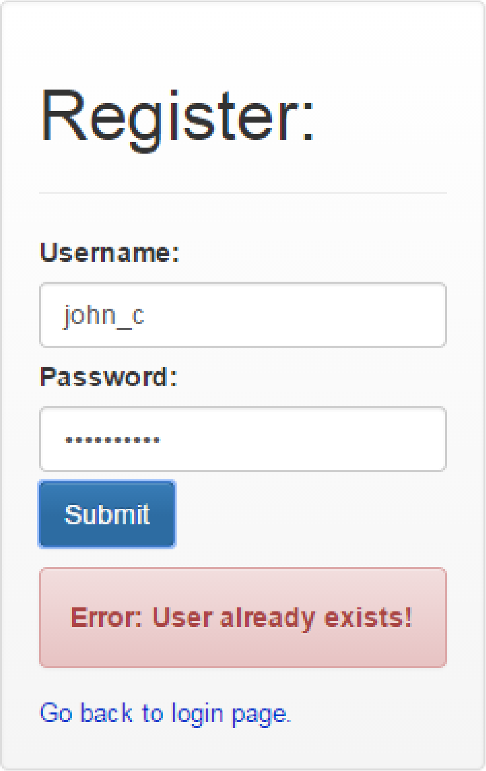
A. Authentication and authorization

In order to use the application's features, a user must first have an account which his tweeting experiences will be linked to. By default, the application's core page is protected from unauthenticated access attempts. Any URL request will be redirected to the default login page.



In the login section, the user can fill in their account's credentials that exist in the application's database, in order to further access the application's features. Any further activity will be related to their account.

If new, the user has the possibility of registering, by following an underlying link that leads to the registration form. An application username and password must be provided, but considering the restriction that every account in the application must have an unique username.


Register:

Username: john_c

Password: ••••••••••

Error: User already exists!

B. Save Filter Groups

The application allows users to save groups of filters for later Twitter requests for Tweets.

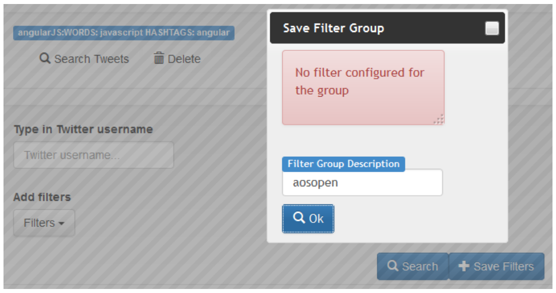
The user can then use their filter preferences for fast tweets search, without having to fill the filters form. Also, the user can delete these filter preferences.

Groups of filters are persisted in the database along with a short description.

We constructed 2 tables for persisting filter preferences in the database:

- FILTER_GROUP_PREFERENCES with columns: USER_ID and FILTER_GROUP_DESCRIPTION
- SAVED_FILTERS with columns FILTER_GROUP_ID, FILTER_CATEGORY, FILTER_CONTENT

When click event is raised on ”Save Filters” button, the user is prompted to provide a short description for the filter group. Error handling is also available, by alerting the user when the description is empty or no filter was selected. The image below illustrates the displayed window when the user wants to save a group of filters:

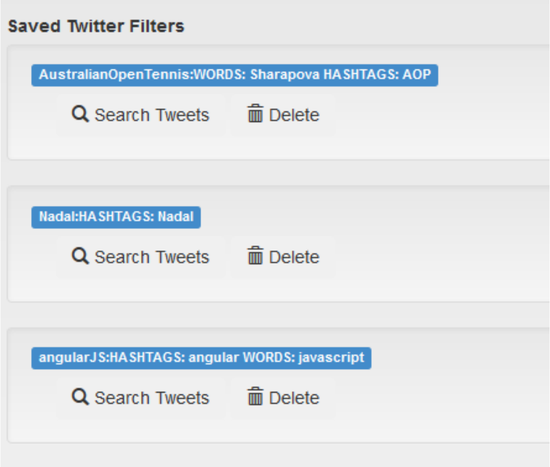


Save Filter Group

No filter configured for the group

Filter Group Description: aosopen

The picture below illustrates how saved filters are displayed in the interface.



Saved Twitter Filters

AustralianOpenTennis:WORDS: Sharapova HASHTAGS: AOP

Nadal:HASHTAGS: Nadal

angularJS:HASHTAGS: angular WORDS: javascript

- The "Search Tweets" button performs a search for tweets based on the corresponding saved filter group. The resulted tweets are displayed in the same section used for form search, at the bottom of the page.
- The "Delete" button erases the corresponding filter group from user's preferences.

C. Sentiment Analysis

For sentiment analysis we used a public service that performs text analysis using a machine learning platform.

Datumbox uses special algorithms, adapted to Twitter flow of data, to perform semantic analysis on Tweet text. The services classify a tweet text as one of positive, negative or neutral.

It uses complex classification techniques accessible through REST API. In order to user the service we signed up for an API key.

The listing below illustrates the URL we perform a HTTP request to. The response of the request is a JSON that we parse for extracting the classification of the tweet text.

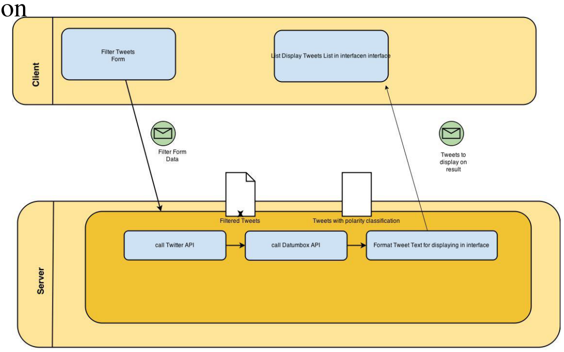
Listing 1. SentimentAnalysisService.java

```
private final String DATUMBOX_API_KEY = "XXXXX";
private final String DATUMBOX_REQUEST_URL =

"http://api.datumbox.com/1.0/
        TwitterSentimentAnalysis.json";
======
String urlParameters = "api_key=" + DATUMBOX_API_KEY +
"&text=" + tweetText;
String request = DATUMBOX_REQUEST_URL;
```

V. Application Interface and flow

The image below illustrates the general flow in the application



Client
Filter Tweets Form
List Display Tweets List in interfacen interface
Filter Form Data
Tweets to display on result
Filtered Tweets
Tweets with polarity classification
Server
call Twitter API
call Datumbox API
Format Tweet Text for displaying in interface

VI. Conclusions

Twitter has rapidly gained worldwide popularity since its debut and has also become an efficient way for people to get known and "followed".

"http://twittercounter.com" publishes the top 100 of the most followed Twitter users and the first place seems to "enjoy" an audience of over 64 millions of followers.

Given this spectacular popularity of Twitter many developers have focused lately on building applications that also integrate the Twitter API for allowing users to follow the activity of their favorite public persons.

Despite the growing interest of developers in integrating the Twitter API, many applications require the user to actually have a Twitter account and login with it. Our application bypasses this restriction and allows user to get access to public tweets in the absence of a Twitter account. The user creates a new account, managed by our application, and then enjoys a personalized interrogation of tweets based on saved favorite groups of filters.

References

[1] http://twittercounter.com/pages/10
[2] https://angularjs.org/
[3] http://docs.spring.io/spring-data/jpa/docs/current/reference/html/
[4] https://dev.twitter.com/
[5] twitter4j.org/
[6] www.liquibase.org/