# Abstract

This is a sample application for the ANEML stack (Angular, Node.js, Express, MarkLogic). Pronounced “animal”. This project is intended to demonstrate the minimal components needed to implement an application using the ANEML stack, while also demonstrating an a common use case.

The demonstration application is a simple application called TweetDeck. This application allows a user to search for tweets stored in a MarkLogic database and plot the geographic location of the Tweeter on Google Maps.

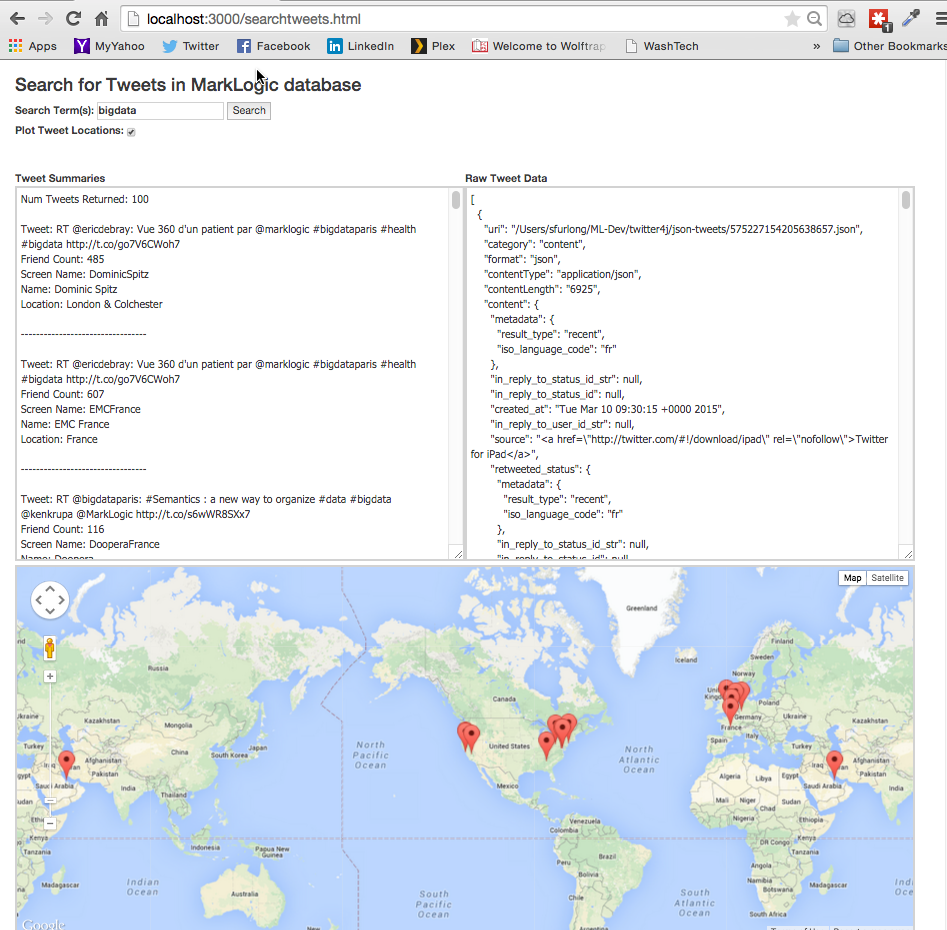


Figure Sample ANEML Application

# Architecture overview

There are two components:

1. Node.js Server. This code is in ./myapp/furserver.js

To start the server type: "node furserver.js"

It will now be listening at: 3000

Port is specified in furserver.js

Node API Docs: http://developer.marklogic.com/learn/node-client-api

2. Angular JS Client. This code is in ./index.html and ./script.js

3. Install MarkLogic Node.js development client

sudo npm install marklogic

\*See detailed instructions in the ML node-dev.pdf Developers Documentation

# Demo toolkit Installation

## Install Node.js

* Install Node from <https://nodejs.org>
* Once you've installed node, you will have an npm command.
* Create a package.json file in the directory of interest, if it does not exist already, with the npm init command.
  + $ npm init

## Install Express

* Install Express in the app directory
  + $ npm install express –save
  + $ npm install express

## Install MarkLogic Node.js Client

* $ npm install marklogic –save

## Install Bootstrap CSS Style Sheets (OPTIONAL)

* $ npm install bootstrap

## Install Demo Code from github

* get from GitHub

# To Run The Demo

## Start the Node.js Server

* $node server.js

# Testing

CTS:Search in Query Console

// find all documents with the word "car" and count them

var count = 0;

var results = new Array();

for (var result of cts.search(cts.andQuery([

cts.wordQuery("marklogic")])) ) {

count++;

//results.push(result);

};

results.push(fn.concat("Count = ", count));

results;

# TODOs

1. Query with Document Joins
2. AuthN via LDAP
3. AuthZ
4. Reporting
   1. Tableau
   2. Pentaho
   3. Qlick
5. Semantics and ontologies

# Appendix

## Working with Git

1. git init
2. vi .gitignore
   1. Add node\_modules
3. Git add -A //Recursibly adds all files to local git repository
4. Git commit –m “<<comment>>” //local commit
5. Git remote add origin <https://sfurlong@github.com/sfurlong/aneml.git>
6. git push -u origin master
7. Git status
8. Git log
9. Git diff head
10. Git push

## Installing & Uninstalling MarkLogic Database

* Install: $ rpm –i <<marklogic-rpm-name>>
* Start: $ /etc/init.d/MarkLogic start
* Stop: $ /etc/init.d/MarkLogic stop
* Uninstall: $ rpm –e MarkLogic
* Remove DB Files: $ rm –fr /var/opt/MarkLogic