# Lab 5 – node.js (or Ruby on Rails) and twitter API

In this lab we are going to create a node server to read from the twitter stream and create a file like we used in lab 1; tweets.json

Your program should read 1000 tweets and put them in a file named according to convention: “*convention*-tweets.json”

For node.js –

You are to use the express framework to setup your server and you will need the fs module (<http://nodejs.org/api/fs.html>) in order to write to your file.

Information about express can be found here: <http://expressjs.com/>

Information about twitter’s streaming API can be found here;

<https://dev.twitter.com/docs/streaming-apis>

But you might want to check this out instead:

Checkout the ntwitter API as an option here; <https://github.com/AvianFlu/ntwitter>

It is not required in this lab to create a web page – however we should be able to use your file as an input file for lab 1.

You will be graded on the following;

Formatting and coding quality : 10

Objective 1 – Setup server using express : 10

Objective 2 – Read from API and save data : 10

Creativity/Coding style : 10

Documentation/Read.me : 10

rpitweets.js

var twitter = require('ntwitter');

var twit = new twitter({

consumer\_key: 'www',

consumer\_secret: ‘xxx',

access\_token\_key: ‘yyy’,

access\_token\_secret: 'zzz'

});

var http=require('http');

http.createServer(function(request, response) {

response.writeHead(200, {

'Content-type': 'text/plain'

});

response.end('Hello HTTP!');

}).listen(8000);

console.log('Listening on http://127.0.0.1:8000');

var sw='-73.68,42.72', ne='-73.67,42.73'; // RPI

twit.stream('statuses/filter', {'locations':sw +','+ne},

function(stream) {

stream.on('data', function (data) {

console.log(data);

});

});