# Abstract

This is a sample stack for ANEML (Angular, Node.js, Express, MarkLogic). Pronounced “animal”.

The framework was constructed using this website as a guide:

http://expressjs.com/starter/installing.html

# 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

# Developers toolkit

# Dev Environment setup

TO RUN THE DEMO:

1. Run the server node ./myapp/furserver.js

2. We need an HTTP server to serve our client pages. I’m using: <https://www.npmjs.com/package/http-server>. To install:

- npm install http-server –g

- start with http-server ./angular <<the parameter is the http file path>>

- Port defaults to 8080

Open Index.html in a browser window

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

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;