

*Update Log*

Key:

Italics mean it’s happening currently

June 5th 2015

*Pi had ssh, apache, and php installed and running...unfortunately:*

I battled with the pi for about 2 hours to get neo4j running and ran into several issues. ARMv7 compatibility problems. Java 8 overloading the SD card on my pi (Couldn’t even tab to complete a command because the card was so full it couldn’t create a .temp file) All in all, I was fed up. I was going to upgrade the card to a 16gb instead of 4gb (don’t even know why that was in there) but I figured I never use my desktop for anything really so I’ll leave the server on there.

Errors I hit ( just for documentation) :

* Pi java version op code unavailable
  + <https://github.com/OpenGrok/OpenGrok/issues/864>
* Java 8 update failed on not enough space
  + <http://www.rpiblog.com/2014/03/installing-oracle-jdk-8-on-raspberry-pi.html>

No that we’re using my desktop, things should actually be really fast regardless. Going to figure out how to dedicate cores to the neo4j processes.

*Config Completed (Active):*

* *Initial Environment*
  + SSH should be up for Eura (my desktop) on **bearnet.ddns.net** on port **21988**
    - your credentials are the same
  + neo4j is up, should be able to access by **bearnet.ddns.net:7474**
    - neo4j admin account is still **neo4j**
    - password is **hQc6ALFqMGvD294Nc0wP**
  + Apache2 installed & running
  + Php installed & running ( confirm visit to index.php)
* *Post Environment*
  + Installed Composer
    - [*https://getcomposer.org/*](https://getcomposer.org/)
* Needs an api to connect with neo4j, we’ll be using NeoClient
  + <https://github.com/neoxygen/neo4j-neoclient>
  + Example Project <https://github.com/neo4j-contrib/developer-resources/tree/gh-pages/language-guides/php/neoclient>
* Why?
  + More frequently updated and supported compared to neo4jPHP & other client apis
  + Multi db capabilities (been thinking we may have a db for each locality (city))
* Installed NeoClient with Composer

*Coding Completed:*

* began php implementation for connecting neo4j

June 6th 2015

*~~\_POST~~ \_GET, isset(), responding, and the persistent need for a schema*

Files hosted from /var/www on the server

Code worked on:

*-* create\_user.php

$\_GET[<variable>] is the call to pull variables from the url address sent in with

[www.mywebsite.com?name=taylor](http://www.mywebsite.com?name=taylor)&password=blabla20202

right now, create\_user should work simply creating a node with a username entered

more options are available but we should test to make sure we can get this working

right for our schema.

- db\_connect.php - might not be necessary

- db\_config.php

Schema:  
 We still need a schema. The data will have to be formated in the nodes so I drew up the schema graph that I shared with you. Need to make sure we have certain levels of updating in our functions. For example, if a user doesn’t enter a photo at account creation do we still add an empty byte array to the node?

Data:

Much of the data, as seen in the data graph I shared, will actually be represented in relationships between nodes, so the hard data we store within node is actually very little. It’s nice since we won’t have to enter as much data as before, the data will be created and fetched as a result of relationships. For example, we don’t need to physically add a user to another user's array of friends. instead, we could simply call the follow\_user.php with the calling user and the followed user passed in and the relationship can be established without formatting or managing an json array of data. Simply creating a relationship between the nodes.

June 6th 2015

*I had it backwards*

It’s not $\_POST that gets data from the address line, its $\_GET which I’ve updated the create\_user.php file to represent. Also, this hosts a wealth of information on neo4j calls

<http://neo4j.com/docs/stable/>

June 8th 2015

*Optional Data*

TODO: Get returned row entries from a “created\_row” return functions for success status

Optional data is going to be kinda weird cause we’ll have to update this as it’s presented. Fortunately neo4j makes this incredibly easy. We can just tack different elements onto items as they are presented. Don’t know if this is better than just adding empty values to invalid fields (such as users not providing an about, do we add an about that's an empty string “” or just not add it at all?). I would prefer not adding them at all,but the question is when getting the queries from the database into android,wouldn’t that cause some null exceptions?like lets say you are getting three users about me, but only one has it ,would a query like that return an error or nullpointer exception? So would adding empty strings be “safer” ?

My thought is it might actually improve speeds if we don’t. Additionally never use the actual string of the neo4j admin and pass. Use the constant presenting in the other file. I’m going to restrict this from user access so even if people got our php code they wouldn’t know the database authentication. Similarly we’re going to do the same thing with hashes for the android installations. Are you saying that we don’t use an authentication when querying?like now we are using the constants NEO\_ADMIN and NEO\_PASSWORD.

Tomorrow morning I’ll write more DAL (data access layer) functions as opposed to just data creation so we can begin to get some valid returns. Make sure to pull the repo and reset the origin. I changed several names of the repos and kicked out teachers so it’s simply our project again :)

here’s some helpful commands:

ereio@Eura:/var/www/html$ git remote remove origin

ereio@Eura:/var/www/html$ git remote add origin https://tereio@bitbucket.org/dingohub/hubbubdb.git

-- Debugged create\_user and tested it , only bug found was a misplaced ‘}’ . Code works perfectly ,good job.

--Debugged create\_event and tested it, only bug found is ‘event\_date’ wasn’t added to the parameters.

used the following to test and started changing values :

<http://localhost/HubbubDb/hubbubdb/database/create_event.php?title=bars&location=panama&start_time=12:00pm&start_date=Tuesday&end_time=3:00pm&end_date=Friday&tags=soccer&owner=ali>

--compser.json and composer.lock file got added to the the git too, those should be ignored right?

--Made get\_todays\_event.php which returns today's events it specifies only a start\_date of the event. The value of the start\_date would be the date the user is using the app.

<http://localhost/HubbubDb/hubbubdb/database/get_todays_event.php?start_date=12/2/2015>

--Created follow\_user.php which creates a relationship of one user with another,right now im using email to find the two nodes and make the relationship. Should we use installation ID ? or id instead? Should we add a property to the relationship? Ex: date the friendship started..

<http://localhost/HubbubDb/hubbubdb/database/follow_user.php?email=ali_leb28@hotmail.com&friend_email=taylortereio@gmail.com>

June 11th 2015

*Less Functions*

How I think we should layout functions:  
  
get\_users.php

get\_events.php

get\_hubs.php

Similarly:

update\_users.php

update\_events.php

update\_hubs.php

Finally:

delete\_users.php

delete\_events.php

delete\_hubs.php

How it searches and what it uses to search will be determined dynamically. For example, if it see’s you’ve sent in a *user ID* and the *relationship (follows)* then it will search for all users who are following the *user ID* passed in. It will allow us less functions for searches and I think the whole thing will be much more elegant than creating a functions for each type of operation.

*Working* Total Function List:

1. create\_users.php
2. create\_events.php
3. create\_hubs.php  
   get\_users.php
4. get\_events.php
5. get\_hubs.php
6. update\_users.php
7. update\_events.php
8. update\_hubs.php
9. delete\_users.php
10. delete\_events.php
11. delete\_hubs.php

June 15th 2015

*Gotta go to class*

Just some quick notes:

* the get\_followers.php created will be absorbed into the get\_users call. //DONE
* The get followers should be working and direction of the relationship is a param to the call itself.
* The the “shift” function just finds all the closing brackets “}” and replaces all but the last one with “} AND”
* Also are you using an IDE or are you just doing it in a text editor, if so which one? I think we should start using a IDE cause I can tell the text editor env is slowing me down. I’ll search for some later today.

Todo:

-unfollow\_user.php

-unfollow\_event.php

-

--Check this out,about having two relationships with the same name User Follows User

and User Follows Event.

<http://stackoverflow.com/questions/24743539/neo4j-two-different-relationships-with-same-name>

I think we should name it different when User Follows Event , we can do User Follow\_Event Event. It’s less confusing.