AnalyseStream.php

The main ‘while’ loop will

get data

parse the json code

put it into an array:

Each element has the following ‘properties’:

[‘id’, ‘parent\_id’, ‘json’]

The ‘id’ and ‘parent\_id’ property was retrieved from the parsed json code

Next, a tree structure is built according to the ‘id’ and ‘parent\_id’ of each tweet:

First, group all tweets having the same parent to the same array, this is a reference to the original array, not a copy:

Array[ ‘parent id’] [ tweet1, tweet2, tweet3, …]

Then, traverse the original array again, comparing current ‘id’ with ‘parent\_id’ from the new array.

If the same, then add the list of child to the new property ’childs’ of the old array

Because they are reference, the new property also pop up in the elements in the new array

Delete all those that already has been added to a parent

Delete property id and parent\_id as we don’t need them anymore

Return the new array

Re-encode the array as json and pass it on to javascript end

First-plugin.php

Main plugin for displaying the data

When the data is retrieved from php, it is passed into a function to recursively access all element and convert them into a different format for storing:

Each status is potrayed as an array with 2 properties:

{ [“status”], [“childs”] }

The status property is actually a Status object, used to store main data about a tweet after converting from the original JSON format.

The Child property will host another array, listing all the tweet that retweet this tweet.

The structure will repeat like that for all level.

In order to easily traverse and find the tweet in the tree, a system of ‘location string’ is used throughout the application.

Basically, the locationString is a string storing information about the location of the tweet in the tree. Ex:

0,2,5,1

This String indicated that first, access the 0th element, then get the 2nd child, then continue to 5th child, then finally 1st child.

There are some utility class for displaying the information uniformly:

Retrieve a html display of a tweet

Draw the whole table base on the array of locationString

Retrieve the tweet object given the locationString

ClassUser.js

Store the class for User and Status (Tweet).

The 2 class is initialized by calling new and pass the parsed json object into it. Ex:

var user = new User( JSON.parse( json ) );

ClassMap.js

ClassChart.js