**Problems faced and solutions found during implementation of Tree Structure**

**P1. How to display the data in json format using jsTree plugin?**

**Sol:** Data in following format will display the data in jsTree format.

{ "attr":{ "id":"rootNodeId" },

"children":[

{

"attr":{"id":" rootNodeId\_subNodeID"},

"children":[],

"data":"subNodeName"

}

],

"data":"rootnodename"

}

**P2. How to traverse through json tree to create new node under specific node?**

**Sol:**

Get the parent node id(**parentNodeID**) of the node under which you want ot create new node and get the new node name. Travrse(loop) through json tree, find id of each node and check if it any id is found with **parentNodeID.** If any node is found, add the new node under the node found with the id **parentNodeID.** If no node id is found, traverse through next node. Continue the same process till the required node is found.

**P3. How to overcome the performance issue in above traversal approach?**

**Sol:**

Use Breadth First Search algorithm that begins at the root [node](http://www.wikipedia.org/wiki/Node_(computer_science)) and explores all the neighboring nodes. Then for each of those nearest nodes, it explores their unexplored neighbor nodes, and so on, until it finds the specified node

Using BFS, we can traverse through all first level child nodes of json tree and find the actual node. The traversal performance can be increased as it won’t traverse through all the unneccesary nodes.