

NestExample.txt

```

<html>
<head>
  <meta charset="UTF-8">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
  <script src="http://d3js.org/d3.v3.min.js"></script>
  <title>NestExample</title>
</head>
<body>
  <textarea id="example" rows="50" cols="120"></textarea>
  <script type="text/javascript">
    //global variable that will hold the nested data.
    var nested_data = [];
    //The non-nested data has to get loaded first, then it is passed to
function that organize it.
    //When attempting to nest and show the whole dataset a problem
occurred. Apparently the .stringify function couldn't handle the whole dataset and
nothing is shown. But inspecting the nested_data variable on the console I could see
the whole data there and it was properly organized. So it is probably possible that
D3 will be able to handle the whole dataset, in case we need that.

$.getJSON("https://health.data.ny.gov/resource/vbkk-tipq.json?$limit=50000&$select=*
&$where=payer= 'Commercial HMO'and (plan_name='Aetna'or plan_name='CDPHP'or
plan_name='Easy Choice Health Plan of NY') and (measure_description='Colon Cancer
Screening' or measure_description='Adult BMI Assessment'or measure_description='Flu
Shot for Adults'or measure_description='Discussion of Aspirin Risks and Benefits'or
measure_description='Aspirin Use')")
    .done(function( data ) {
      //data(variable) here contains all the values from the
query, that is what gets passed as an argument to the nest function.
      nest(data);
    });

    //After this function runs the nested_data variable now has the
organized data.
    //If you want to visualize the nested_data on the console, and how
it is organized all you have to do is uncomment line 35 and open the file with the
developer tools(USE Chrome!!, tip: press F12) and type is the variable name on the
console. You will be able to click through the array of objects as it is nested.
    function nest(data) {
      //here the you choose which keys you will use to nest. The
hierarchy is by what gets called first. So in this case Payer will be the higher
hierarchy followed by domain and so forth.
      //The last function determines what is the data(non-nested).
      var nested_data = d3.nest()
        .key(function(d) {return d.payer; })
        .key(function(d) {return d.domain; })
        .key(function(d) {return d.sub_domain; })
        .key(function(d) {return d.measure_description; })
        .entries(data);

      //debugger;
      //This changes the inner html of the textarea element.
      //the .stringify function transforms the JSON data to a
string.
      $("#example").html(JSON.stringify(nested_data, null, 3));
    }
  </script>
</body>
</html>

```