[Advanced] Nested Queries for displaying information about Jira Screens & Tabs

This an advanced example that demonstrates the usage of nested Queries.

For this example it is expected that you already have used PocketQuery to execute a simple Query and know the basics.

- Setting up the Datasource
- Setting up the Queries
  - Parent Query: Get the tabs of a Jira Screen
  - Child Query: Get the Fields of a Screen Tab
- Combining the Queries
- Polishing the Template and using AUI

Setting up the Datasource

For this example there needs to be an application link between Confluence and Jira.

If you don't have an application link set up, please see this article for information on how to do that.

Once the application link is set up, head over to the PocketQuery admin and create a new Datasource of type REST Application Link and give it an appropriate name.

To make sure, that PocketQuery can connect to the linked Jira instance, click the "Test Connection" button. You can use "/rest/api/2/serverInfo" as a Test URL.

If PocketQuery tells you that the Connection test was successful, you have successfully completed the Datasource setup!

Additional Information

If you can't create an application link to your Jira, be aware that you can also use a Basic REST Datasource instead. However, as the application link preserves permissions across instances, it is probably the better choice. With an application link, users can only execute queries against Projects, Issues, etc. that they have access to. If you use a Basic REST Datasource with an service account instead, they can also execute REST API calls to which they usually wouldn't have access to.

Setting up the Queries
For this example we want to build a nested Query.

In simple terms, this means that we want to build a Query, that can use it's result to execute another Query.

For this reason we need to create two Queries: a parent Query and a child Query.

**Parent Query: Get the tabs of a Jira Screen**

For our parent query we want to get all tabs of a screen. For this we can hit the endpoint "/rest/api/2/screens/:screenId/tabs".

Simply create a Query with that URL that uses our previously created Datasource.

If we try to execute the Query in this form, we should see a simple table generate from the Result.
We are now able to see a list of all the tabs for that Screen with name and id.

The next step is to get more information about each tab, using a nested Query.

**Child Query: Get the Fields of a Screen Tab**

To get more information about a tab we can call the endpoint "/rest/api/2/screens/:screenId/tabs/:tabId/fields".

Go ahead and create a second query for the same Datasource, using this URL.
If we test this Query with a valid `screenId` and a valid `tabId` we can see that it returns information about the fields of the given tab.

Great! The last step is to combine the Queries together, to dynamically fetch information about all tabs for a given `screenId`.

Combining the Queries
To be able to call the child Query from the parent Query we need to create a custom template.

Inside that template we can use the PocketQuery Velocity API to call and execute other Queries.

Using the PocketQuery Velocity API we have several ways to combine the Queries. In our case we are going to use $PocketQuery.executeQuery().

This API method returns the result of another given Query. Using this, we can loop over the rows of the parent query and execute a nested Query for each tabId.

Since our child query also needs to be passed the screenId from the parent Query, we should read the screenId from the parameters like this: #set ($screenId = $queryParameters.get("screenId"))

The result that we get from the nested Query is again a list/table, so to render it appropriately, we need a second #foreach() loop.

Together with some more slight adjustments, like setting custom column header names, the full Template could look like this:

```velocity
Template

## Save the screen ID in a variable.
#set($screenId = $queryParameters.get("screenId"))

<h2>This are the tabs for Screen ID: $screenId</h2>
<table class="aui confluenceTable pocketquery-table">
   <thead>
      <tr>
         <th class="confluenceTh">Tab ID</th>
         <th class="confluenceTh">Tab Name</th>
         <th class="confluenceTh">Tab Fields (ID, Name, Type)</th>
      </tr>
   </thead>
   <tbody>
      #foreach ($row in $result)
         ## Save Tab Name and ID in variables.
         #set($tabId = $row.id)
         #set($tabName = $row.name)
         ## Executed the nested Query "Fields_For_Tab_Id", passing it screenId and tabId.
         #set($nestedResult = $PocketQuery.executeQuery("Fields_For_Tab_Id", { "screenId" : $screenId, "tabId" : $tabId }))
         <tr>
            <td class="confluenceTd">$tabId</td>
            <td class="confluenceTd">$tabName</td>
            <td class="confluenceTd" style="padding: 0;">
               <table class="aui confluenceTable pocketquery-table">
                  <tbody>
                     #foreach ($row in $nestedResult)
                        <tr>
                           <td class="confluenceTd">$row.id</td>
                           <td class="confluenceTd">$row.name</td>
                           <td class="confluenceTd">$row.type</td>
                        </tr>
                     #end
                  </tbody>
               </table>
            </td>
         </tr>
      #end
   </tbody>
</table>
```

Make sure you create a new Template with this code and assign it to your parent Query.
If you did everything correctly, the result should look like this:

This are the tabs for Screen ID: 6543210

<table>
<thead>
<tr>
<th>Tab ID</th>
<th>Tab Name</th>
<th>Tab Fields (ID, Name, Type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10300</td>
<td>General</td>
<td>summary, Summary, System field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>description, Description, System field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>reporter, Reporter, System field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>assignee, Assignee, System field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>fixVersions, Fix Version/s, System field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>priority, Priority, System field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>components, Component/s, System field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>environment, Environment, System field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>attachment, Attachment, System field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>labels, Labels, System field</td>
</tr>
<tr>
<td>10301</td>
<td>Test Details</td>
<td>isuelinks, Linked Issues, System field</td>
</tr>
<tr>
<td>10302</td>
<td>Test Sets</td>
<td></td>
</tr>
<tr>
<td>10303</td>
<td>Pre-Conditions</td>
<td></td>
</tr>
<tr>
<td>10204</td>
<td>Test Plans</td>
<td></td>
</tr>
<tr>
<td>10505</td>
<td>Link Issues</td>
<td></td>
</tr>
</tbody>
</table>

Polishing the Template and using AUI

The result is great! We got our nested Queries to work together and can now see all the Tab Fields for a given Screen ID.

However, the presentation might not be the best since nested tables are not easy to read.

But using PocketQuery we can do so much more than just generate tables!

We have full control over the HTML and can use the AUI (Atlassian User Interface) library.

Through making use of AUI we can generate actual tabs from the result that are clickable:
This was achieved using this Template:
## Save the screen ID in a variable.
#set($screenId = $queryParameters.get("screenId"))

This are the tabs for Screen ID: $screenId

```html
<div id="screen-tabs-$screenId" class="aui-tabs horizontal-tabs">
  <ul class="tabs-menu">
    #foreach ($row in $result)
      #set($tabId = $row.id)
      #set($tabName = $row.name)
      <li class="menu-item #if($velocityCount==1)active-tab#end">
        <a href="#screen-tab-$tabId">$tabName</a>
      </li>
    #end
  </ul>
  #foreach ($row in $result)
    #set($tabId = $row.id)
    #set($tabName = $row.name)
    ## Executed the nested Query "Fields_For_Tab_Id", passing it screenId and tabId.
    #set($nestedResult = $PocketQuery.executeQuery("Fields_For_Tab_Id", {
      "screenId" : $screenId,
      "tabId" : $tabId
    }))
    <div class="tabs-pane #if($velocityCount==1)active-pane#end" id="screen-tab-$tabId">
      #if ($nestedResult.size() > 0)
        <table class="aui">
          <thead>
            <tr>
              <th>ID</th>
              <th>Name</th>
              <th>Type</th>
            </tr>
          </thead>
          <tbody>
            #foreach ($row in $nestedResult)
              <tr>
                <td>$row.id</td>
                <td>$row.name</td>
                <td>$row.type</td>
              </tr>
            #end
          </tbody>
        </table>
        #else
        <p>This tab doesn't appear to have any fields.</p>
        #end
      #end
    </div>
  #end
</div>
```

```javascript
AJS.toInit(function() {
  AJS.tabs.setup();
  $('#screen-tabs-$screenId .tabs-menu .menu-item a').on('click', function(e) {
    e.preventDefault();
    e.stopPropagation();
    AJS.tabs.change(e.target, e);
  });
});
```