PSUG National Information Exchange

Users Helping Users

PowerQueries

And the Data Export Manager

By Jim Parsons

jparsons@vcschools.org

goo.gl/J2uyqn

Shared Folder: https://goo.gl/J2uyqn

About Me

Name: Jim Parsons

School: Valley Christian School, Cerritos, CA

Contact: jparsons@vcschools.org

Married 21 Years, 1 Daughter + 2 Grandkids

Teacher for 25 years

Interests: computer programming, preaching, swing dancing, and PowerSchool !

Test Server Available

https://ps9.psugcal.org/admin

Admin User: #aghs1 … where # is a number from 1 to 99

Password: PSaghs1 … same password for all users

Using PowerQueries

Why use PowerQueries?

1. Data Export Manager - custom reports (including scheduled outputs)
2. Internal API - via AJAX
3. External API - from a 3rd Party Server (OAuth)

Demonstrations:

1. Creating a PowerQuery
2. Accessing via Data Export Manager
3. Accessing via AJAX (jQuery or Angular)

Using `tlist_sql` to Get JSON data

```
[
~[tlist_sql;
    SELECT
        lastfirst,
        floor(months_between(sysdate, dob) / 12),
        to_char(dob,'Month D, YYYY')
    FROM
        students
    WHERE enroll_status = 0
        AND floor(months_between(sysdate, dob) / 12) <= ~(gpv.max)
        AND floor(months_between(sysdate, dob) / 12) >= ~(gpv.min)
]
    {
        "lastfirst":"~(name;json)",
        "age":"~(age;json)",
        "birthdate":"~(dob;json)"
    },
[/tlist_sql]{}]
```

Create a PowerQuery

1. PowerQueries are loaded via PlugIns
2. Create three XML files:
   a. ./plugin.xml
   b. ./queries_root/example.named_queries.xml
   c. ./permissions_root/example.permission_mappings.xml
3. ZIP the files, and install the PlugIn.

The “plugin.xml” File

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin xmlns="http://plugin.powerschool.pearson.com"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://plugin.powerschool.pearson.com
plugin.xsd"
        name="Plugin with PowerQuery Example"
        version="1.0"
        description="Show the basics of creating a PowerQuery">
    <oauth></oauth>
    <publisher name="Valley Christian Schools">
        <contact email="support@vcschools.org"/>
    </publisher>
    <access_request>
    </access_request>
</plugin>
```

The Access Request Section

```
<access_request>
    <field table="STUDENTS" field="STUDENT_NUMBER" access="ViewOnly"/>
    <field table="STUDENTS" field="DOB" access="ViewOnly"/>
    <field table="STUDENTS" field="ENROLL_STATUS" access="ViewOnly"/>
    <field table="STUDENTS" field="LASTFIRST" access="ViewOnly"/>
    <field table="STUDENTS" field="GRADE_LEVEL" access="ViewOnly"/>
</access_request>
```

The “example.named_queries.xml” File

```
<queries>
    <query name="org.vcschools.project.students.example"
            coreTable="students" flattened="false">
        <description>(a short description)</description>
            <args>
                (argument definitions go here)
            </args>
            <columns>
                (column definitions go here)
        </columns>
         <sql>
            <![CDATA[
                (your SQL goes here)
            ]]>
         </sql>
    </query>
</queries>
```

PowerQuery Arguments

```
<args>
    <arg name="age_limit"
        description="The maximum age returned."
        type="primitive" required="true" />
</args>
```

Arguments are declared within the SQL as any name preceded by a colon.

```
SELECT
    lastfirst,
    floor(months_between(sysdate, dob) / 12) AS age,
    to_char(dob,'Month D, YYYY') as birthdate
FROM
    students
WHERE
    floor(months_between(sysdate, dob) / 12) <= :age_limit
```

PowerQuery Columns

```
<columns>
    <column column="STUDENTS.LASTFIRST">lastfirst</column>
    <column column="STUDENTS.DOB">age</column>
    <column column="STUDENTS.DOB">birthdate</column>
</columns>
```

Since the SQL returns three columns, you must declare three columns

```
SELECT
    lastfirst,
    floor(months_between(sysdate, dob) / 12) as age,
    to_char(dob,'Month D, YYYY') as birthdate
FROM
    students
WHERE
    floor(months_between(sysdate, dob) / 12) <= :age_limit
    AND schoolid = :school_id
```

PowerQuery SQL Statement

1. Only allows SELECT (or WITH . . . SELECT); read only
2. Columns with calculations MUST be aliased

```
<sql>
    <![CDATA[
        SELECT
            lastfirst,
            floor(months_between(sysdate, dob) / 12) AS age,
            to_char(dob,'Month D, YYYY') AS birthdate
        FROM
            students
        WHERE
            floor(months_between(sysdate, dob) / 12) <= :age_limit
    ]]>
</sql>
```

The “example.permission_mappings.xml” File

```
<permission_mappings>
    <permission name='/admin/home.html'>
        <implies allow="post”>
            /ws/schema/query/org.vcschools.project.students.birthdays
        </implies>
    </permission>
</permission_mappings>
```

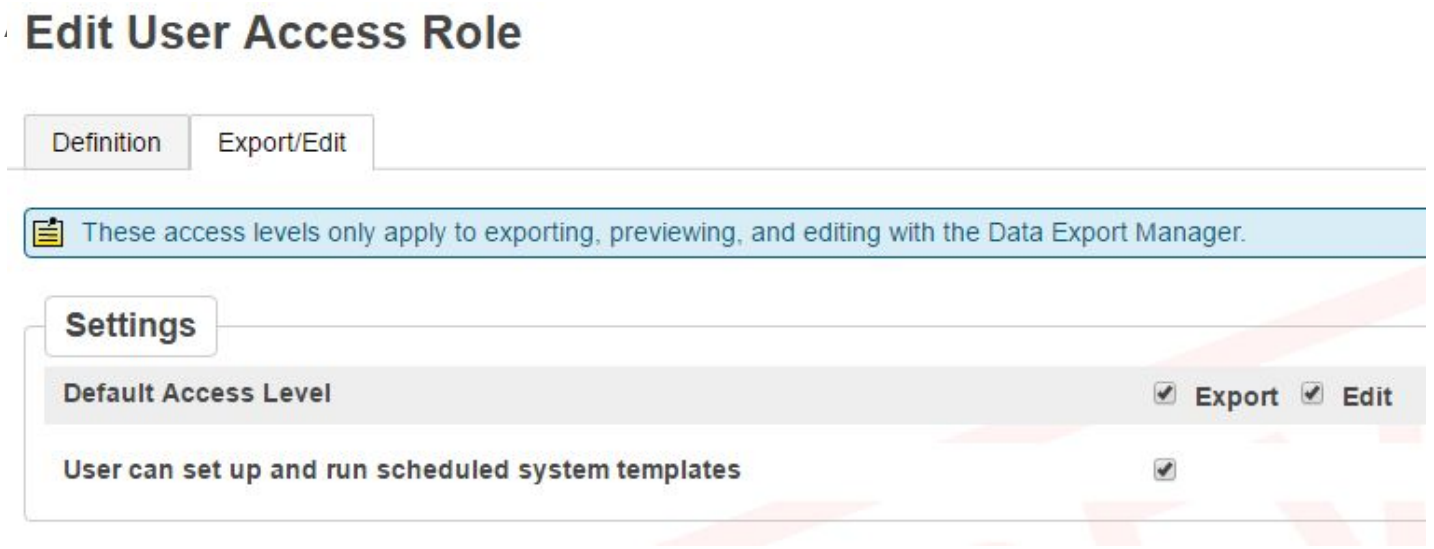
Basically, this says, “Whoever has access to “/admin/home.html” also has access to this PowerQuery. This applies only to internal API access via cookies, and will constrain the results to selected school (and/or exclude records outside the scope of the user’s permissions or restrictions).

Data Export Manager

PowerQueries are available in DEM

1. Exported data is constrained to the scope of the selected school.
2. The PowerQuery must not contain “arguments”
3. Columns with calculated data can NOT be filtered in DEM

Data Export Manager

1. Scheduling saved templates is limited to users with a Role that allows it.
   a. Create a Role or check the box to allow Scheduled Exports in an existing role.
   b. 

2. Add Remote Connections (only SFTP supported)

Data Export Manager

Depending on the user's permissions, the Data Export Manager will constrain the results to the scope of the current school and the permissions mapping settings.

These constraints are not enforced on scheduled exports.

Using AJAX (jQuery Example)

```
<div class="box-round">
    <label for="select-box”>Drop Down Selector: </label>
    <select id="select-box">
        <option value=""></option>

    --- additional <options> inserted by jQuery script ---

    </select>
</div>
```

Using AJAX (jQuery Example)

```
var url = "/ws/schema/query/org.vcschools.project.students.sample";
var payload = { age_limit_min: 12, age_limit_max: 16 };
$j.ajax({
    method: 'POST',
    url: url,
    dataType: 'json',
    data: JSON.stringify(payload),
    contentType:"application/json",
    success: function( resp ) {
        for ( i=0; i < resp.record.length; i++ ) {
            $j('#select-box').append('<option value="' +
            resp.record[i].lastfirst + '">' +
            resp.record[i].lastfirst + '</option>');
        };
    }
});
```

Using AJAX (AngularJS Example)

```
<div class="box-round" data-ng-controller="myController" data-ng-
cloak>
    <label for="select-box”>Drop Down Selector: </label>
    <select id="select-box">
        <option
            data-ng-repeat="item in list"
            value="{{item.lastfirst}}"
        >{{item.lastfirst}} ({{item.age}})</option>
    </select>
</div>
```

Using AJAX (AngularJS Example)

```
myApp.controller('myController2', function($scope, $http) {
    $scope.list = [];
    $scope.age_limit_min = 12;
    $scope.age_limit_max = 16;
    var url =
        "/ws/schema/query/org.vcschools.project.students.agelimit";
    var payload = { age_limit_min: $scope.age_limit_min,
                    age_limit_max: $scope.age_limit_max };
    $http.post (
        url,
        Payload,
        { headers:{'Content-Type':'application/json'}}
    ).then( function ( response ) {
        $scope.list = response.data.record;
    });
});
```

Hands on Time - Lab

- Create a PowerQuery
- Install the PQ (via Plug-In)
- Access PQ via Data Export Manager
- Access via REST Client (using Oauth)

Questions?

Thank you for attending.

Resources:
https://support.powerschool.com/developer - a must for the PowerSchool customizer

Jim Parsons
jparsons@vcschools.org