Timesheet Status Report

The delivered functionality of reporting in Project Web App does not include timesheets with at status of “not yet created”. This is interesting because the timesheet tiles you can have from the webpart "track my work" include the counts of timesheets that haven't been created in the unsubmitted counts. Alas, Microsoft Support confirms this is not available as a direct status through reporting.

Prasanna Adavi created an excellent guide that shows you how to display this information in a pivot table: <https://gallery.technet.microsoft.com/projectserver/Step-By-Step-to-create-a-f4eb9270>.

If you’re like my organization, though, you need the information in an actionable form whereby management can be empowered to follow up on timesheets for their departments. A good solution for that is a flat table with filterable columns like department, timesheet manager, week, status, etc. In this way, as more people submit and approve timesheets the list can shrink to only show remaining timesheets that haven’t been submitted.

A little further, this report includes a custom enterprise field called “EmployeeType”. Here we track whether someone is a staff member, consultant, or a student employee/intern. You would need to exclude this from your queries if you don’t also have that field.

# Prerequisites

This report uses Power Query add on for Microsoft Excel. You’ll need to first [get that from Microsoft](https://www.microsoft.com/en-us/download/details.aspx?id=39379&WT.mc_id=Blog_PBI_Announce_DI) and install it.

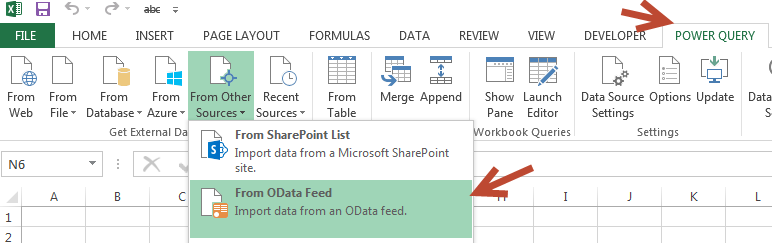
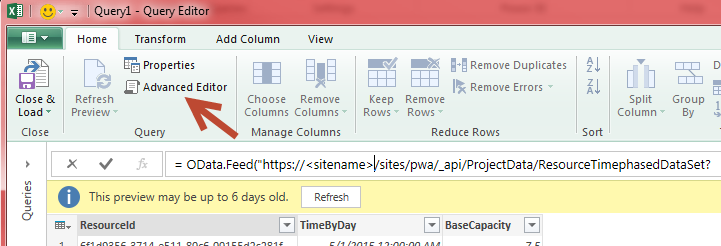
You’ll also need the appropriate reporting permissions.

# Overview

To make this report, we’ll create four queries. The first three we’ll load with a connection only. The fourth and final we’ll load as a table and add it to the data model.

# Details

## ResourcesTimePhasedData

* From the **Power Query** Tab in Microsoft Excel, select **From Other Sources** in the Get External Data section, then click **From** **Odata Feed**.  
  
* The URL will be **https://<sitename>/sites/pwa/\_api/ProjectData/ResourceTimephasedDataSet?$select=ResourceId,TimeByDay,BaseCapacity&$filter=BaseCapacity gt 0**
* Open the **Advanced Editor** from the Query section of the Home tab.   
  
* Paste this code over the code in the Advanced Editor query box then click **Done**:

let

Source = OData.Feed("https://<sitename>/sites/pwa/\_api/ProjectData/ResourceTimephasedDataSet?$select=ResourceId,TimeByDay,BaseCapacity&$filter=BaseCapacity gt 0"),

#"Added Custom" = Table.AddColumn(Source, "Custom", each Date.StartOfWeek([TimeByDay], Day.Monday)),

#"Renamed Columns" = Table.RenameColumns(#"Added Custom",{{"Custom", "StartOfWeekWcapacity"}}),

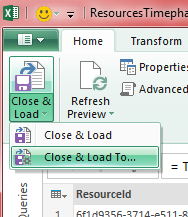
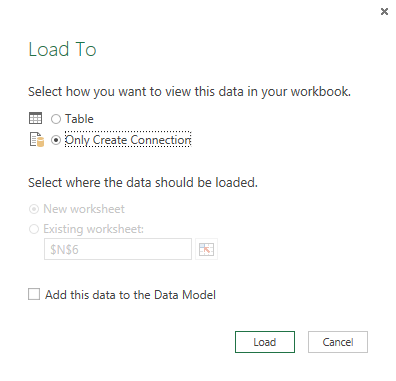
#"Removed Columns" = Table.RemoveColumns(#"Renamed Columns",{"TimeByDay", "BaseCapacity"}),

#"Removed Duplicates" = Table.Distinct(#"Removed Columns"),

#"Changed Type" = Table.TransformColumnTypes(#"Removed Duplicates",{{"StartOfWeekWcapacity", type date}}),

#"Filtered Rows" = Table.SelectRows(#"Changed Type", each [StartOfWeekWcapacity] < DateTime.Date(DateTime.LocalNow()))

in  
 #"Filtered Rows"

* Click **Close and Load To**  
  
* Choose **Only Create Connection**, then choose **Load**.

## Timesheets

* Repeat the same steps as above, substituting the query URL, Advanced Code, and query name as follows:
  + URL

https://<sitename>/sites/pwa/\_api/ProjectData/Timesheets?$select= TimesheetStatusId,StatusDescription,TimesheetOwnerId,StartDate

* + Advanced Editor

let

Source = OData.Feed("https://<sitename>/sites/pwa/\_api/ProjectData/Timesheets?$select= TimesheetStatusId,StatusDescription,TimesheetOwnerId,StartDate"),

#"Changed Type" = Table.TransformColumnTypes(Source,{{"StartDate", type date}})

in

#"Changed Type"

* + Query Name

Timesheets

## Resources

Resources (I have a custom enterprise field called "employmenttype" included. I also wanted to display only the child department of Resource Departments, so there's some splits and removes included.

* Repeat the same steps as above, substituting the query URL, Advanced Code, and query name as follows:
  + URL  
    https://<sitename>/sites/pwa/\_api/ProjectData/Resources?$select=ResourceId,ResourceName,ResourceTimesheetManageId,ResourceDepartments,EmploymentType&$filter= ResourceType eq 2 and ResourceIsGeneric ne true
  + Advanced Editor

let

Source = OData.Feed("https://<sitename>/sites/pwa/\_api/ProjectData/Resources?$select=ResourceId,ResourceName,ResourceTimesheetManageId,ResourceDepartments,EmploymentType&$filter= ResourceType eq 2 and ResourceIsGeneric ne true"),

#"Split Column by Delimiter" = Table.SplitColumn(Source,"ResourceDepartments",Splitter.SplitTextByEachDelimiter({"."}, null, true),{"ResourceDepartments.1", "ResourceDepartments.2"}),

#"Changed Type" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{"ResourceDepartments.1", type text}, {"ResourceDepartments.2", type text}}),

#"Removed Columns" = Table.RemoveColumns(#"Changed Type",{"ResourceDepartments.1"}),

#"Renamed Columns" = Table.RenameColumns(#"Removed Columns",{{"ResourceDepartments.2", "ResourceDepartments"}}),

#"Merged Queries" = Table.NestedJoin(#"Renamed Columns",{"ResourceTimesheetManageId"},#"Renamed Columns",{"ResourceId"},"NewColumn"),

#"Expanded NewColumn1" = Table.ExpandTableColumn(#"Merged Queries", "NewColumn", {"ResourceName"}, {"NewColumn.ResourceName"}),

#"Renamed Columns1" = Table.RenameColumns(#"Expanded NewColumn1",{{"NewColumn.ResourceName", "TimesheetManager"}}),

#"Removed Columns1" = Table.RemoveColumns(#"Renamed Columns1",{"ResourceTimesheetManageId"})

in

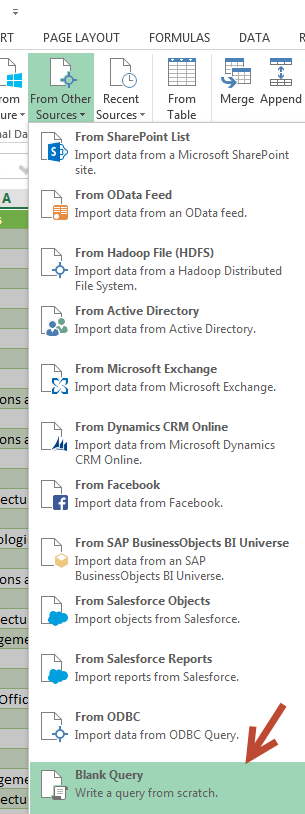
#"Removed Columns1"

* + Query Name

Timesheets

## All\_Timesheets

Bringing all those queries together, ALL\_Timesheets. It includes two columns for status... one detailed, and the other I'm calling summary which groups into approved, submitted, and not submitted (which includes anything not of the first two statuses. This would mean that if you don't force a timesheet manager and have two or more possible approval steps that "acceptable" status would be included in "not submitted" in this report, but you could change that to be whatever you need.)

* For this last query, instead of choosing the source as From Odata, we’ll choose the last option in the dropdown, **Blank Query.**
* Advanced Editor

let

Source = Table.NestedJoin(ResourceTimephasedDataSet,{"ResourceId", "StartOfWeekWcapacity"},Timesheets,{"TimesheetOwnerId", "StartDate"},"NewColumn"),

#"Expanded NewColumn" = Table.ExpandTableColumn(Source, "NewColumn", {"StatusDescription"}, {"NewColumn.StatusDescription"}),

#"Replaced Value" = Table.ReplaceValue(#"Expanded NewColumn",null,"Not Yet Created",Replacer.ReplaceValue,{"NewColumn.StatusDescription"}),

#"Merged Queries" = Table.NestedJoin(#"Replaced Value",{"ResourceId"},Resources,{"ResourceId"},"NewColumn"),

#"Expanded NewColumn2" = Table.ExpandTableColumn(#"Merged Queries", "NewColumn", {"EmploymentType","ResourceName", "ResourceDepartments", "TimesheetManager"}, {"EmploymentType","ResourceName", "ResourceDepartments", "TimesheetManager"}),

#"Removed Columns" = Table.RemoveColumns(#"Expanded NewColumn2",{"ResourceId"}),

#"Renamed Columns" = Table.RenameColumns(#"Removed Columns",{{"StartOfWeekWcapacity", "WeekStarting"}, {"NewColumn.StatusDescription", "Status"}}),

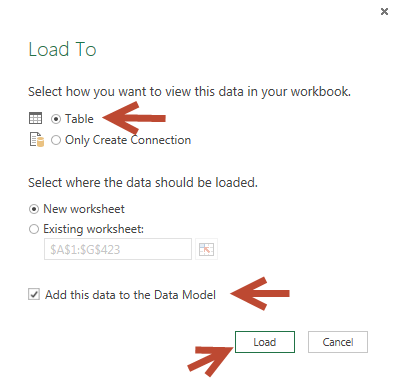
#"Reordered Columns" = Table.ReorderColumns(#"Renamed Columns",{"ResourceDepartments", "TimesheetManager", "ResourceName", "WeekStarting", "Status"}),

#"Sorted Rows" = Table.Sort(#"Reordered Columns",{{"ResourceDepartments", Order.Ascending}, {"TimesheetManager", Order.Ascending}, {"EmploymentType", Order.Ascending}, {"ResourceName", Order.Ascending}, {"WeekStarting", Order.Ascending}}),

#"Added Custom" = Table.AddColumn(#"Sorted Rows", "SummaryStatus", each if [Status]="Approved" then [Status] else if [Status]="Submitted" then [Status] else "Not Submitted")

in

#"Added Custom"

* + Query Name  
    All\_Timesheets
  + Click **Close & Load To**: **Table** & **Data Model**, **Load**

# Final Product

And now, here’s what that looks like as a table:

# 

From there you can sort, filter, etc. to your heart’s desire. Also, because you added the data to the data model, you can easily add pivot tables or Power View reports.