INEIGHT

Schedule Resource Spread Data

Download and Transform in Power BI

SCHEDULE

Changelog

This changelog contains only significant or other notable changes to the document revision. Editorial or minor changes that do not affect the context of the document are not included in the changelog.

Rev	Date	Description
1	25-FEB-2026	Initial Release

Contents

Overview

This guide provides details for you to do the following:

- Generate required API keys and permissions
- Connect Power BI to the Schedule Resource Spread API
- Import data using a Power Query script
- Transform the downloaded JSON into a structured dataset

This allows you to automate the download of Activity Resource Spread files directly into Power BI.

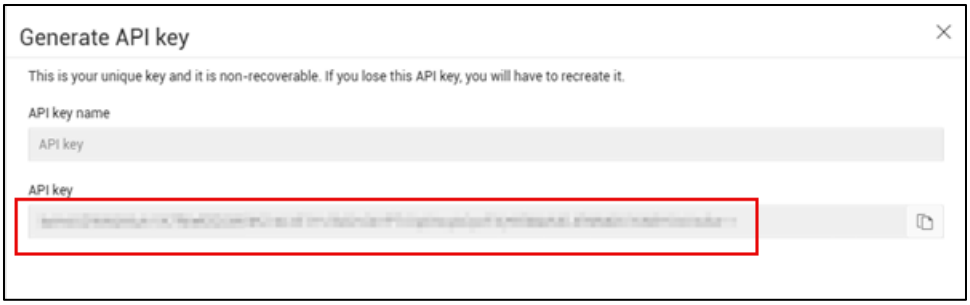
Prerequisites

The following keys is necessary to create your API connection in Power BI. The role selected to generate an API key must be at least a Level 1 – Project Admin with the Download schedule resource spread permission enabled. To verify the permission is enable for the role, go to Suite Administration > Roles and permissions > **Schedule**. Make sure the Download schedule resource spread check box is selected.

Generate an x-api-key

1. From the main menu, go to Suite Administration > **API key Management**.
2. Click the **Add API key** icon at the top of the page. The Generate API key dialog box opens.
3. Complete the information needed to create a specific key for the schedule resource spread data.
    - API key name: Enter a name for the key.
    - Start date: The date when the key is accessible.
    - End date: The date when the key expires.

      When creating the key, the end date can be set for up to one year from the current date. You can click the **Edit API key** icon and extend the end date when needed or use the Extend now icon to set the end date to one year from today's date.
    - Role and Organization/Project: Use the drop-down lists to select a role and the organization or project. The role must be at least a Level 1 – Project Admin with the Download schedule resource spread permission enabled.
4. Click the **Add** icon to add the role. You can add multiple roles to use the API key.
5. Click **Generate**. The key is generated and shows in the dialog box.



IMPORTANT:	This is the only instance that the key is shown. Copy and save it in a location where you can access later.

Generate an APIM subscription key (Ocp-Apim-Subscription-Key)

Refer to the following documentation to generate an APIM subscription key.

- The APIM subscription key is created in API Management: https://developer.ineight.com/
- Instructions to obtain the subscription key: https://developer.ineight.com/getting-started

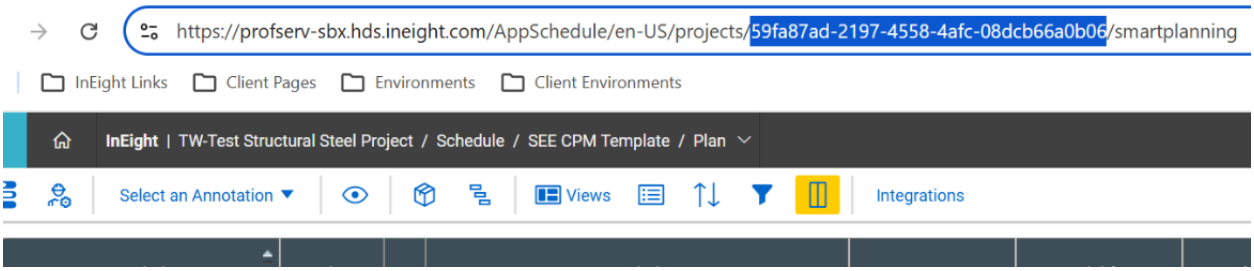
Additional information

- Tenant prefix: The tenant prefix shows in the URL for your environment.

  Example: https://*tenant prefix*.hds.ineight.com/AppCore/UserLandingView

- Schedule GUID: Open a schedule and the GUID shows in the URL.

  Example: https://profserv-sbx.hds.ineight.com/AppSchedule/en-US/projects/*59fa87ad-2197-4558-4afc-08dcb66a0b06*/smartplanning



Generate resource spread data

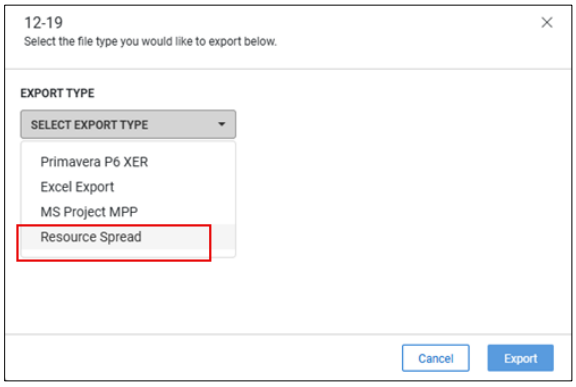
Before the schedule resource spread data can be accessed by the API, it must first be generated and then exported from the Schedule application. You can generate and export the data from either the Project list page or in the schedule.

To export from the Project list page:

1. Open Schedule, the Project list page shows.
2. Select the check box for the schedule that you want to generate resource spread data.
3. In the toolbar, click the **Generate resource spread** icon.
4. A dialog box opens for you to confirm your selection. Click **Confirm**.

Export from a schedule

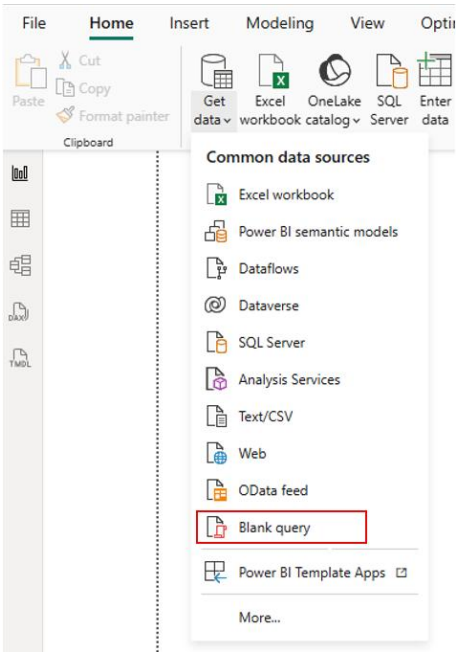
1. From the schedule that you want to export.
2. Click the **Export** icon in the toolbar. The schedule dialog box opens.
3. In the Export Type drop-down list, select **Resource Spread**.



4. In the Data Presentation drop-down list, select **Hierarchical**.
5. Click **Export**.

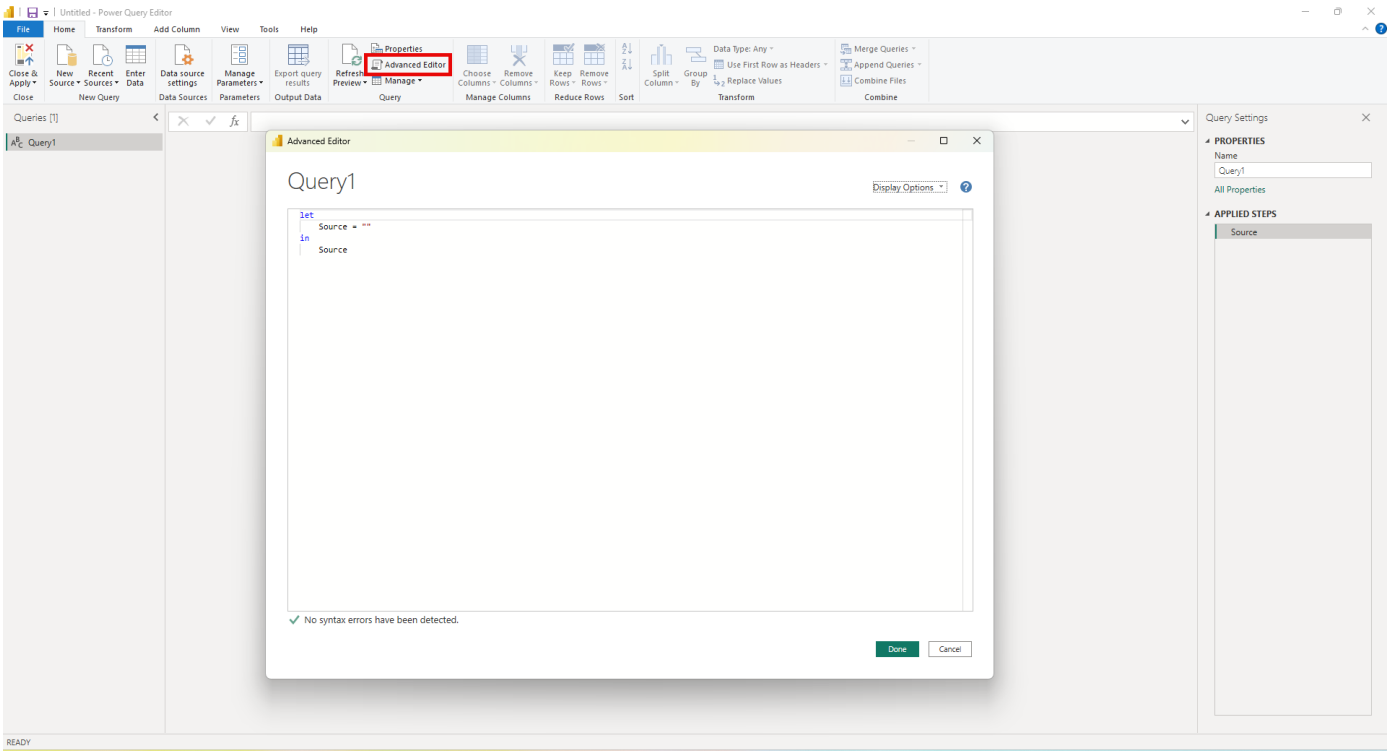
Connecting the API to Power BI

Create a blank query in your Power BI file. On the Home tab, select Get data > **Blank query** in the list of data sources. The Power Query Editor opens.



Open Advanced Editor

On the Home tab, select **Advanced Editor**. This opens a dialog box where you will paste the Power Query M script after removing the existing text.



Load scripts

Basic Import Script (Raw JSON Load)

If you need to download the file as-is, use the following M code and enter the information detailed from Prerequisites in place of the highlighted text.

```
let
   url = "https://api.ineight.com/schedule/ActivityResourceSpread/59fa87ad-
2197-4558-4afc-08dcb66a0b06(your schedule GUID)",
   fileBinary = Web.Contents(url,
      [
         Headers = [
            #"x-api-key" = "your-x-api-key",
            #"X-IN8-TENANT-PREFIX" = "Demo (tenant prefix)",
            #"Ocp-Apim-Subscription-Key" = "your-Ocp-Apim-Subscription-Key"
         ]
      ]),
   Source = Json.Document(fileBinary)
in
   Source
```

Full Transformation Script (Convert JSON → Table)

If you want a fully flattened usable table (Planned, Remaining, Actual, Late spreads by Date), use the extended script:

```
let
   url = "https://api.ineight.com/schedule/ActivityResourceSpread/59fa87ad-
2197-4558-4afc-08dcb66a0b06(your schedule GUID)",
   fileBinary = Web.Contents(url, [
      Headers = [
         #"x-api-key" = "your-x-api-key",
         #"X-IN8-TENANT-PREFIX" = "Demo (tenant prefix)",
         #"Ocp-Apim-Subscription-Key" = "your-Ocp-Apim-Subscription-Key"
      ]
   ]),
   Source = Json.Document(fileBinary),

   ConvertedToTable = Record.ToTable(Source),
   Value = ConvertedToTable{1}[Value],
   ConvertedToTable1 = Record.ToTable(Value),

   ExpandedValue = Table.ExpandRecordColumn(
      ConvertedToTable1,
      "Value",
      {"PlannedResourceSpread", "RemainingResourceSpread",
"ActualResourceSpread", "RemainingLateResourceSpread"},
      {"PlannedResourceSpread", "RemainingResourceSpread",
"ActualResourceSpread", "RemainingLateResourceSpread"}
   ),

   Unpivoted = Table.UnpivotOtherColumns(
      ExpandedValue,
      {"Name"},
```

```
      "SpreadType",
      "SpreadRecord"
   ),

   AddDateValueTable = Table.AddColumn(Unpivoted, "DateValueTable", each
Record.ToTable([SpreadRecord])),
   RemoveSpreadRecord =
Table.RemoveColumns(AddDateValueTable,{"SpreadRecord"}),

   ExpandedDateValue = Table.ExpandTableColumn(
      RemoveSpreadRecord,
      "DateValueTable",
      {"Name", "Value"},
      {"Date", "Value"}
   ),

   Grouped = Table.Group(
      ExpandedDateValue,
      {"Name", "Date"},
      {{"Data", each _, type table [SpreadType=nullable text, Value=nullable
number]}}
   ),

   AddPivotedColumns = Table.TransformColumns(
      Grouped,
      {"Data", each Table.Pivot(_, List.Distinct([SpreadType]), "SpreadType",
"Value", List.Sum)}
   ),

   ExpandedFinal = Table.ExpandTableColumn(
      AddPivotedColumns,
      "Data",
      {"PlannedResourceSpread", "RemainingResourceSpread",
"ActualResourceSpread", "RemainingLateResourceSpread"},
      {"PlannedResourceSpread", "RemainingResourceSpread",
"ActualResourceSpread", "RemainingLateResourceSpread"}
   ),

   ChangedTypes = Table.TransformColumns(
      ExpandedFinal,
      {
         {"Date", each Date.FromText(_, "en-US"), type date},
         {"PlannedResourceSpread", each Number.From(_), type number},
         {"RemainingResourceSpread", each Number.From(_), type number},
         {"ActualResourceSpread", each Number.From(_), type number},
         {"RemainingLateResourceSpread", each Number.From(_), type number}
      }
   )
in
   ChangedTypes
```

The full transformation script produces a table format like this example.

Name	Date	Planned	Remaining	Actual	Remaining Late
…	…	…	…	…	…
…	…	…	…	…	…
…	…	…	…	…	…
…	…	…	…	…	…

Authentication

After saving, Power BI may prompt you to with Edit Credentials.

- Choose **Anonymous** (your authentication is handled using the keys above)
- Organizational privacy level

Troubleshooting

Issue	Resolution
401 Unauthorized	Check API key, APIM subscription key, role permissions.
Anonymous credentials not accepted	Make sure you are using Anonymous, not Organizational or Oauth.
No data returned	Confirm schedule GUID exists and user has permission.
Date conversion errors	Make sure locale = "en-US" in Date.FromText