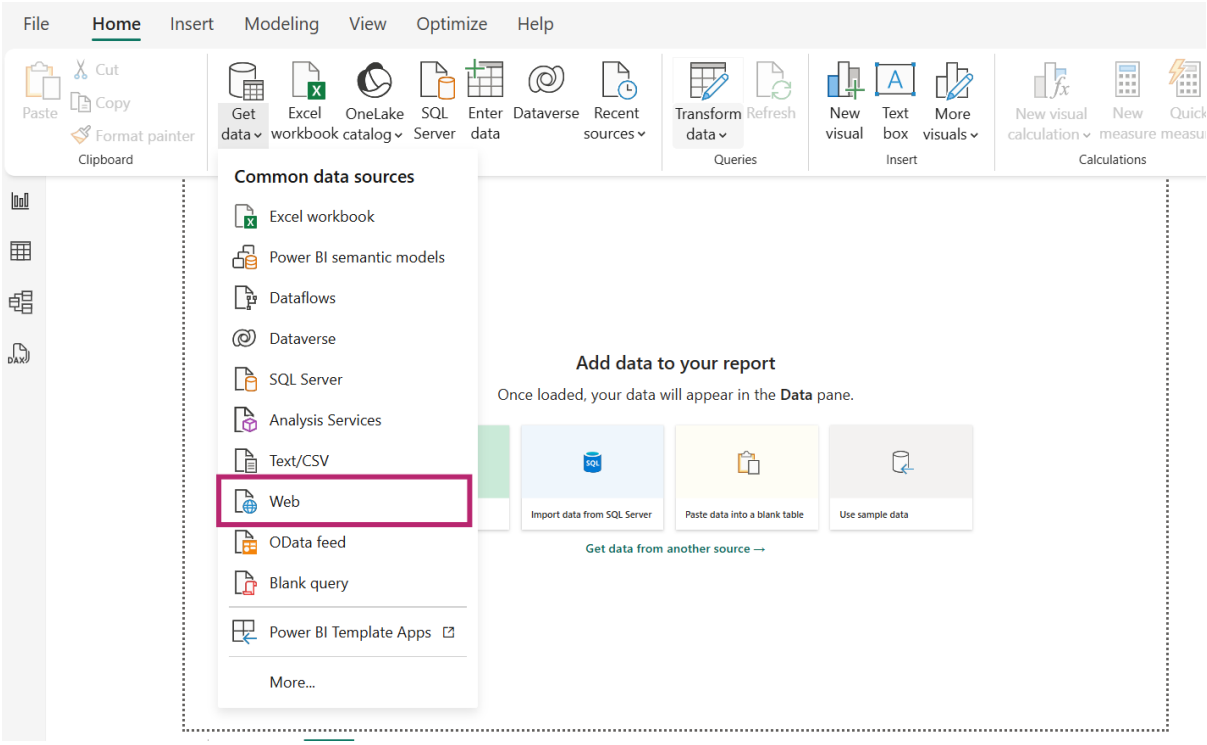
How to set up your Aide API in Power BI

This guide shows you how to connect your Aide CRM data to Power BI using an API. You can follow these steps whether you're adding the data to an existing report or starting a new one. You'll learn how to set up the connection, work with the data (including nested data), and link related tables so you can build useful charts and visuals in Power BI.

Step 1: In the Power BI Desktop application:

1. Go to the **Home** tab
2. Click on the arrow next to the **Get data** command button
3. Select the option **Web**



Step 2: Enter your API information

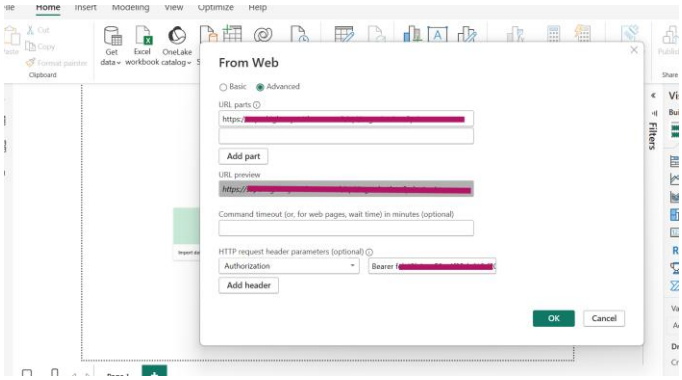
Please refer to Aide CRM's API documentation for more information and email support@aidecrm.co.uk to ask for your **API URL** and **API Key**.

The **API URL** is the unique URL for your AideCRM database table.

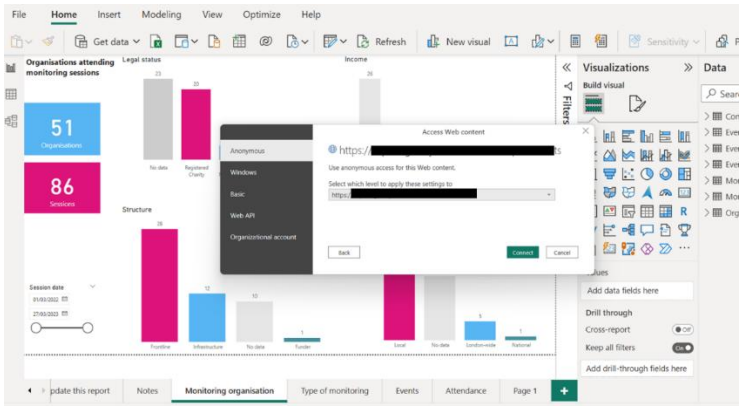
The **API key** is the unique identifier that authenticates requests to your Aide CRM. This is like a password into your database so should be stored in a secure location.

Once you have these 2 bits of information you'll need to enter them as follows:

1. Select **Advanced**
2. Enter the API URL in the field **URL parts**
3. In the **HTTP request header parameters (optional)** field type the word **Authorization** into the first box (don't worry that it isn't on the drop down list, this is just a list of the most frequently used requests but you can simply type in your own request) and in the second box type the word Bearer, then a space, then your API code.
4. Click on **OK**



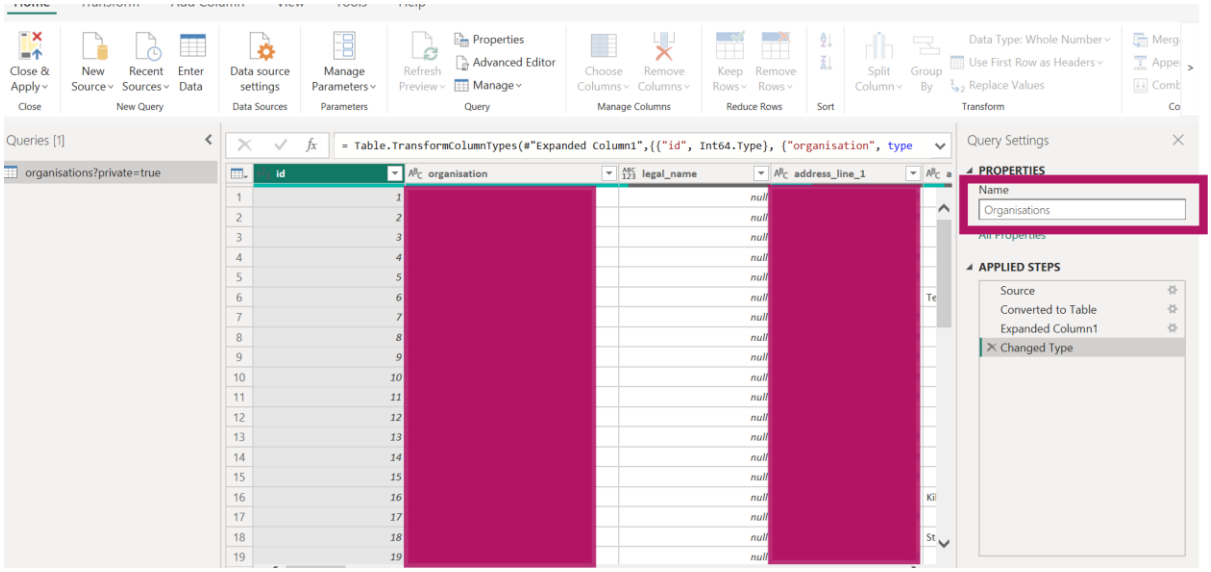
5. Click on **Connect**.



Step 3: Loading the results in Power Query

Assuming all goes to plan the results of your API call should load as a query in the **Power Query Editor**. If you have any problems with it loading please refer to the Aide API documentation which lists the most common error codes you might encounter.

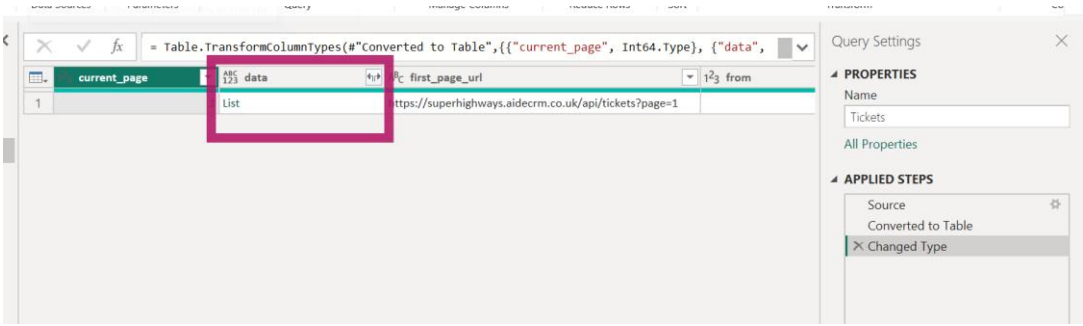
1. **Rename** your query so that it's easy to identify in your data model.
2. **Close and Apply** the query.



Extract nested data in queries

Some of the data might be in a "nested" data structure. This structure keeps everything related to the record bundled together. An example of nested data might be when you have a Project record with multiple organisations associated with it. The Project would be the top level and the organisations underneath would be held as nested records. Using this structure keeps the data clean and organised as well as allowing the API to work faster.

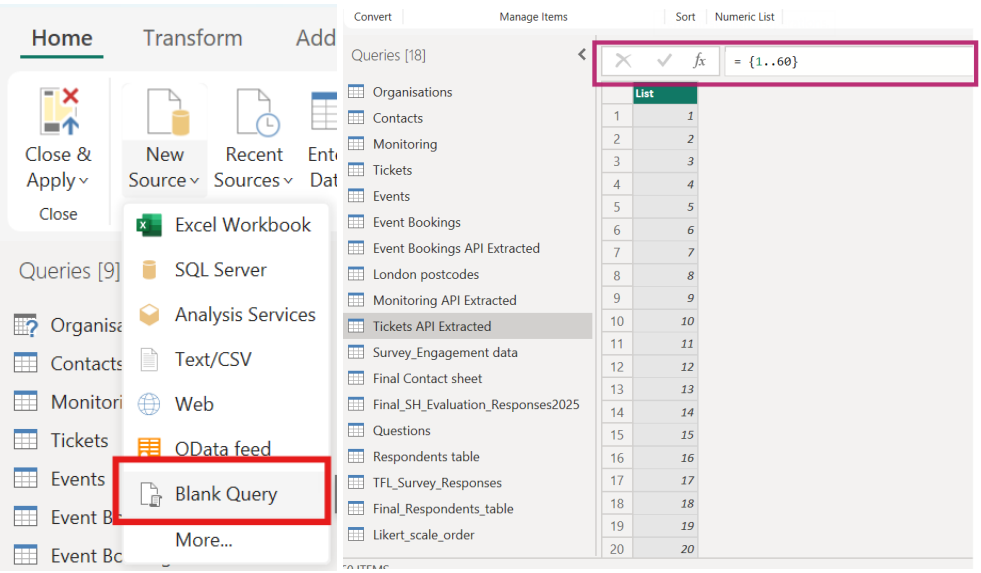
In columns with nested data, instead of a value the word List will appear (screenshot below)



To convert these nested fields into data you can use for reporting you need to create custom columns which extract the data.

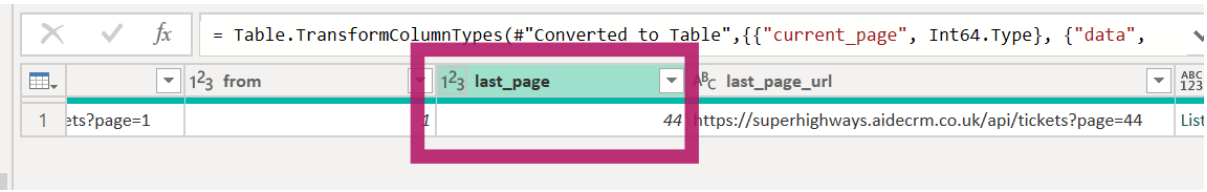
1. Generate a List of Pages

- Use a list expression in Power Query to generate page numbers:



= {1..LAST_PAGE_NUMBER}

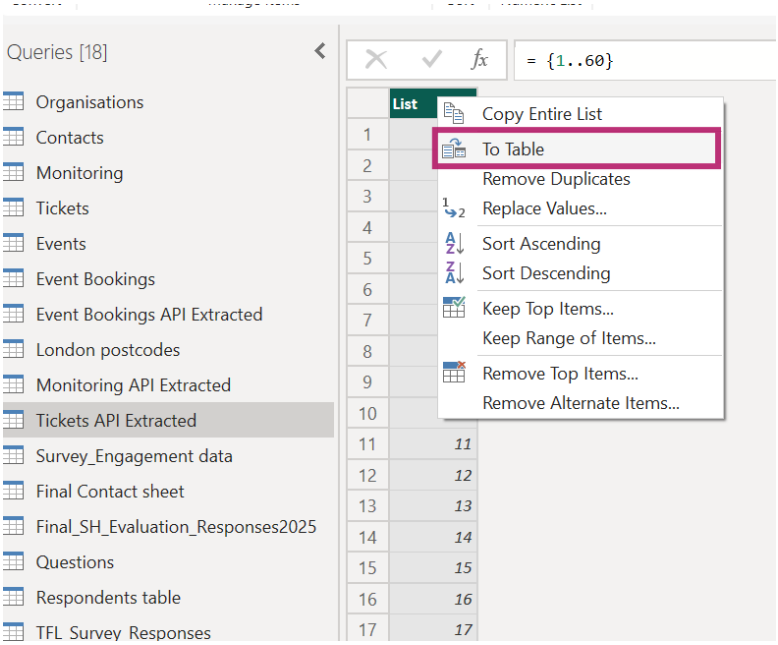
Replace LAST_PAGE_NUMBER with the total number of pages the API provides.

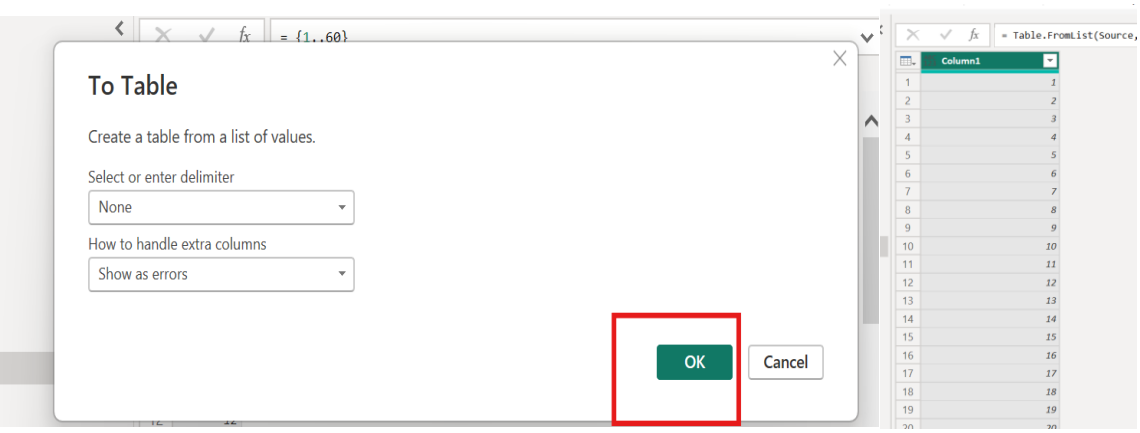


The last page number can typically be found in a column named last_page or something similar. However**, it's recommended to add a buffer of extra pages beyond this number** to accommodate any future data expansion. This helps prevent issues like nesting from reoccurring after page 44 (or the identified last page). In our case, we've added a buffer up to page 60 to ensure stability.

2. Convert List to Table

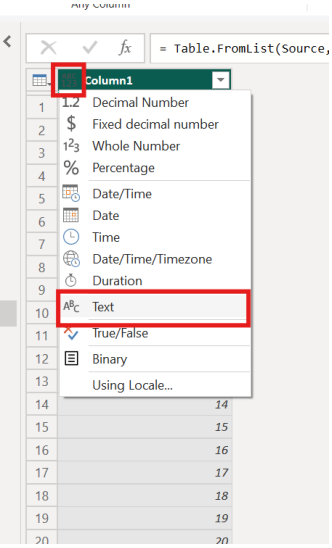
Right-click the list output and choose **Convert to Table**.





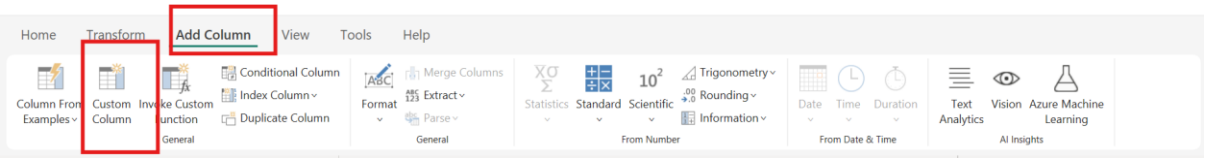
3. Convert Data type of the Column1 to text

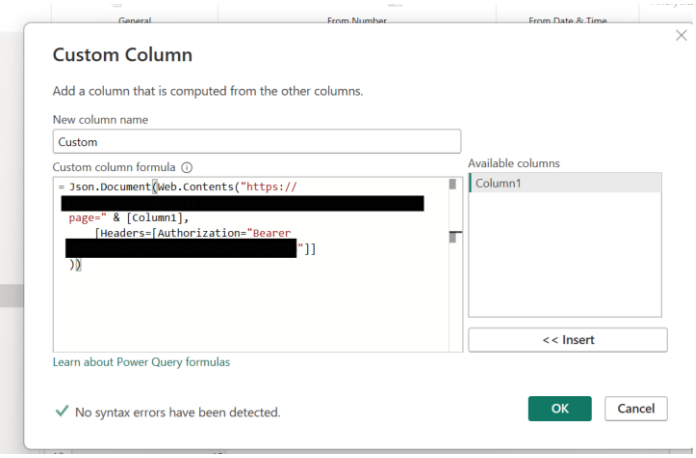
Go to **Column**, click on the drop down and convert to text



4. Add a Custom Column for API Call

Go to **Add Column > Custom Column**, and use the following code:






```
Json.Document(Web.Contents("https://YOUR_API_BASE_URL/api/tickets?reporting=true&page=" & Text.From([Column1]), [Headers=[Authorization="Bearer YOUR_ACCESS_TOKEN"]]))
```

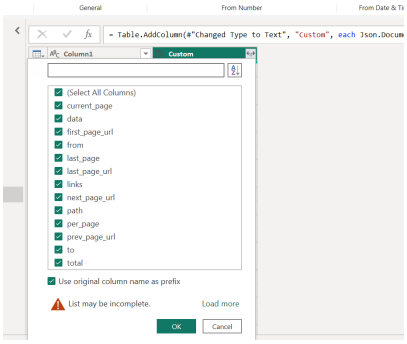
- **YOUR_API_ENDPOINT** : Replace this with the actual API base URL.
- **YOUR_ACCESS_TOKEN** : Users should insert their own API key.
- **[Column1]**: This dynamically inserts page numbers for pagination.

5. Expand the Data

You may need to expand multiple nested records. Follow these steps:

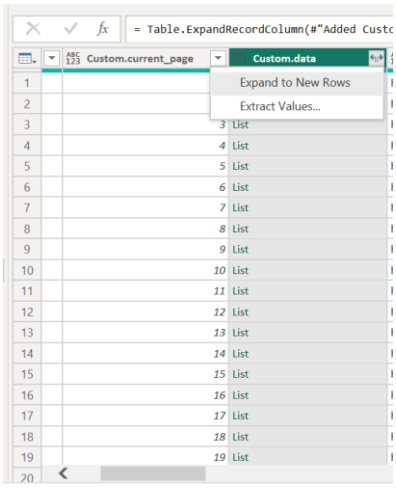
a. Expand the Record Column:

- Click the icon next to the custom column and select all fields or the main list (list, for example).



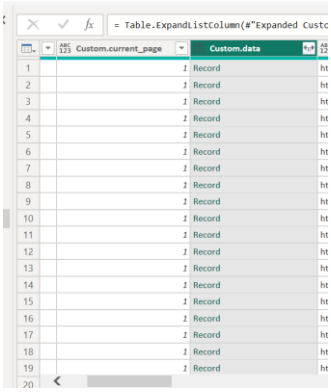
- ### b. Expand the list Column:

  This may contain a list of records. Expand this list.



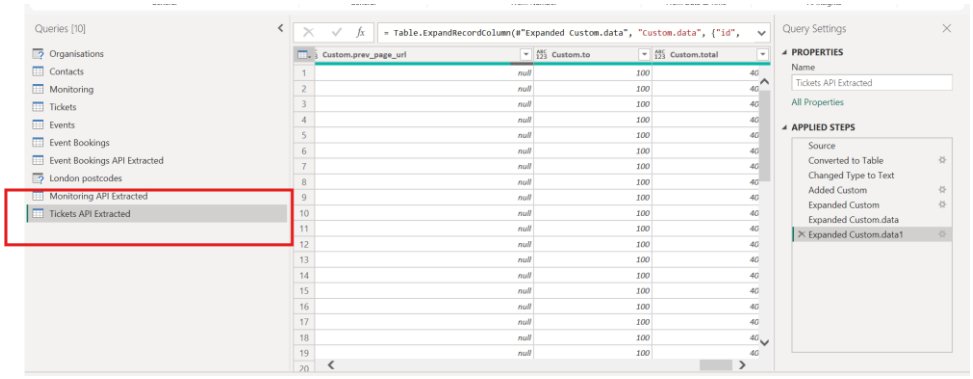
c. Expand Final Record Column:

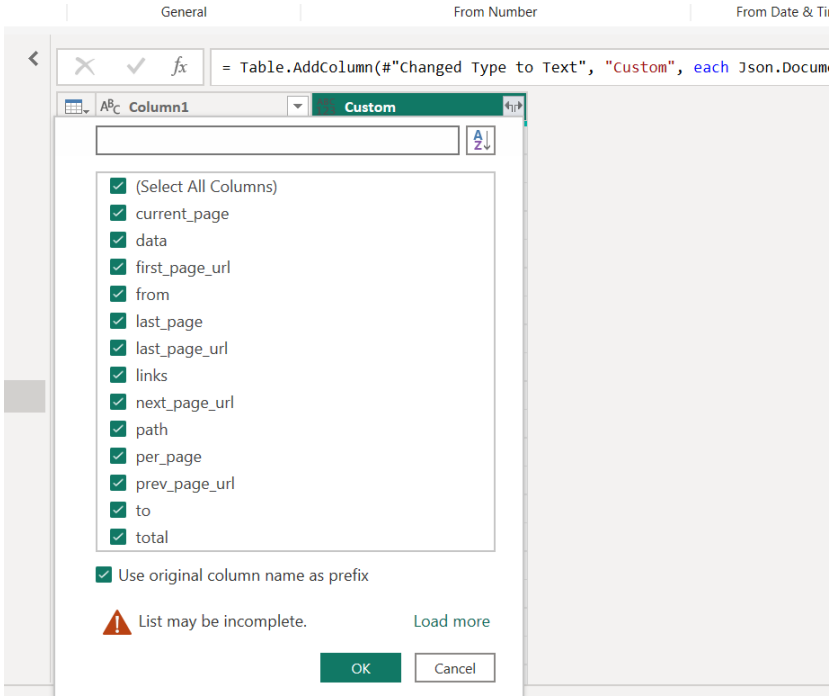
- This is the actual data per ticket. Expand all relevant fields.



6. Rename your table with suitable name

Double click on the table name and rename it and expand the columns.





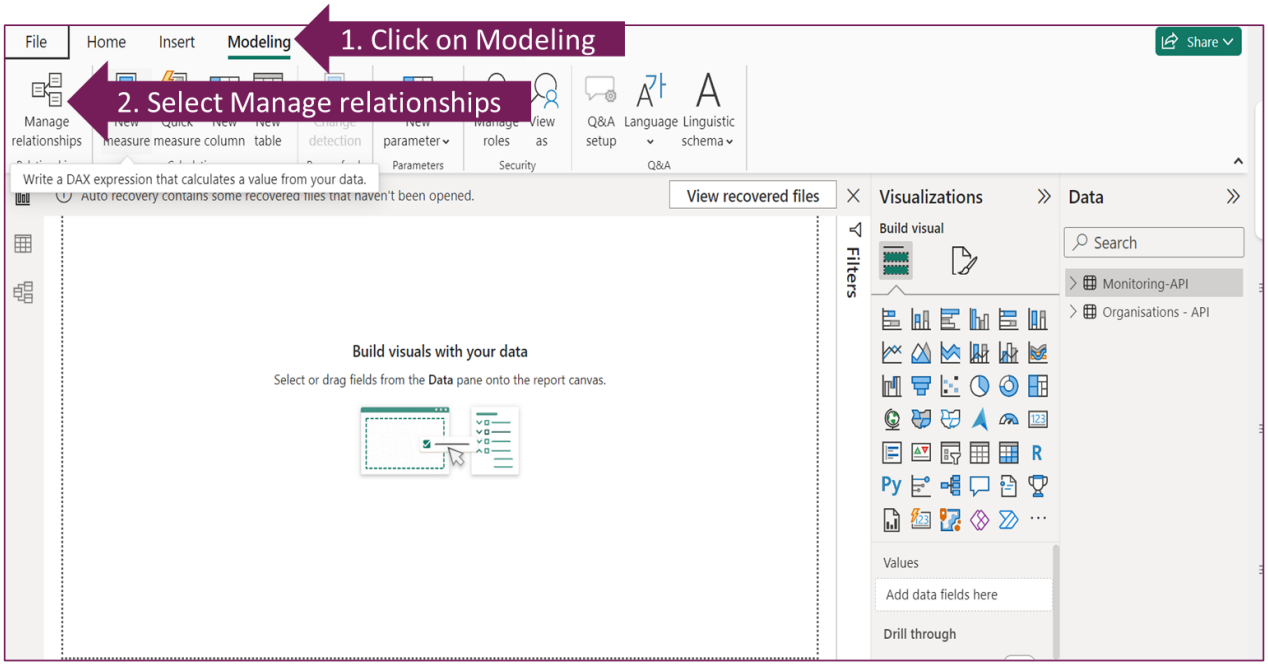
Manage relationships

In PowerBI you will need to recreate the relationships between the entities in the Aide database e.g. link Contact to Organisations. PowerBI will attempt to spot relationships, but you'll need to ensure they are correct.

There are 2 ways of managing relationships, one list format and one visual. It's worth trying both to see which one you prefer. Both have the same functionality:

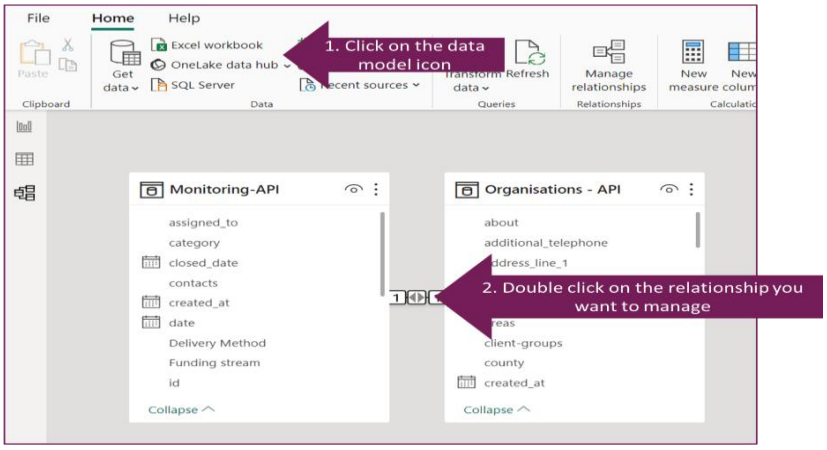
Managing relationships in a list

1. Click on the **Modelling** tab
2. Click on the first button in the ribbon, **Manage relationships**



Managing relationships in a data model diagram

1. Click on the **Data model** icon
2. Double click on the relationship you want to manage



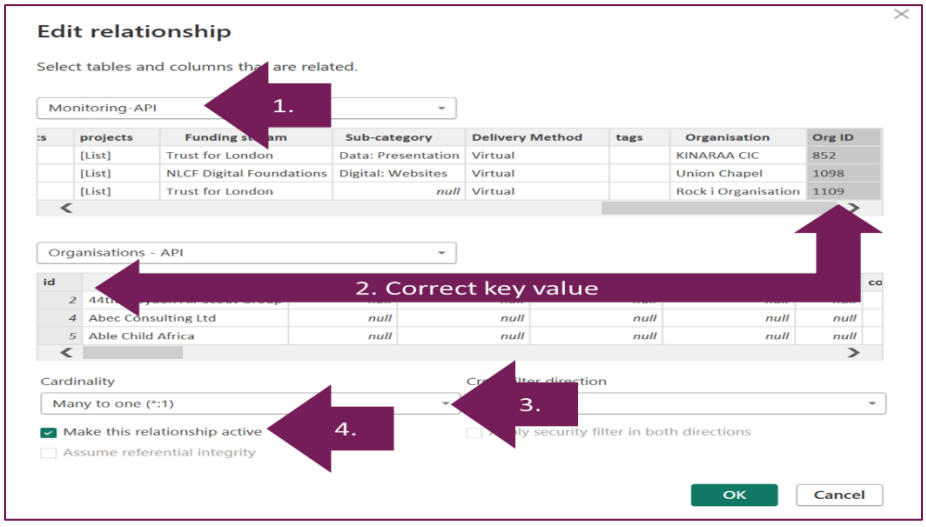
Editing a relationship

The PowerBI resource site has a comprehensive guide on relationships to follow https://learn.microsoft.com/en-us/power-bi/transform-model/desktop-create-and-manage-relationships

There are 4 key things to check when editing a relationship:

1. Are the correct tables selected?
2. Are the correct key value fields selected?
3. Is the relationship type correct e.g. one to one or one to many?
4. Is the relationship active?

The screen shot below shows how the relationship between Monitoring and Organisations should look



Visualising the data

Now you have set up your relationships you can use data from both tables in one visual. Join our PowerBI training or join the Superhighways' Power BI user group to access learning resources and a peer support community