Kippy API Integration with Power BI using JWT for Authentication

Nauman Khan

13-Mar-2025

Intro

This paper provides details of a proof of concept that integrates a kippy data endpoint using basic authentication and a JWT token from PowerBI desktop. The outcome successfully showed how kippy users could dynamically pull data from kippy to PowerBI.

The example uses the /api/v4/query/ data endpoint, which when called with the parameter “table=user” returns all users in the kippy instance. See https://www.kippy.cloud/api for more information on this and other data endpoints.

Background

What is Basic Authentication?

Basic Authentication is a simple and widely used method for authenticating users on the web. It is a part of the HTTP protocol and involves sending a username and password in the request headers. The basic idea is to include a "Authorization" header in the HTTP request, which contains the word "Basic" followed by a space and a base64-encoded string of "username:password".

Here's a basic example of how it works:

1. The client (such as a web browser or a software application) sends an HTTP request to a server.
2. The server responds with a 401 Unauthorized status code, indicating that authentication is required.
3. The client includes an "Authorization" header in the request with the credentials encoded in base64.

For example:

Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=

In this example, "dXNlcm5hbWU6cGFzc3dvcmQ=" is the base64-encoded form of "username:password".

What is a JWT token?

JWT stands for JSON Web Token, and it is a compact, URL-safe means of representing claims to be transferred between two parties. JWTs are often used for authentication and authorization in web applications.

JWTs are commonly used for authentication by generating a token on the server side, sending it to the client upon successful login, and having the client include the token in the headers of subsequent requests. The server can then verify the integrity and authenticity of the token using the stored secret or public key.

What is PowerBI?

Power BI is a business analytics service developed by Microsoft that allows users to visualize and analyze data from a variety of sources, helping businesses make data-driven decisions. It's a suite of tools designed to transform raw data into interactive dashboards, reports, and visualizations, providing insights for better decision-making.

Here's an overview of some key features and components of Power BI:

Data Connectivity: Power BI connects to a wide range of data sources, including Excel spreadsheets, SQL databases, cloud services, and online platforms. This flexibility allows users to import and combine data from multiple sources.

Data Transformation: The Power Query editor in Power BI allows users to clean, transform, and prepare data for analysis. Users can reshape, filter, and manipulate data as needed before creating visualizations.

Visualizations and Reports: Power BI offers a variety of visualization options, such as charts, graphs, maps, and tables, to represent data. Users can create interactive reports with multiple visualizations, allowing them to explore data and discover insights.

Dashboards: Power BI dashboards are collections of visualizations that provide a high-level overview of key metrics and insights. Dashboards can be customized to focus on specific areas of interest and shared with others.

Power BI Service: This is the cloud-based component where users can publish and share their reports and dashboards. It supports collaboration and sharing within an organization and includes features like scheduled data refreshes, security settings, and mobile access.

Power BI Desktop: A Windows application used to create reports and visualizations. It's the primary tool for data transformation and report design before publishing to the Power BI Service.

Collaboration and Sharing: Power BI enables users to share reports and dashboards with colleagues or teams. It includes features like data-driven alerts, collaboration, and integration with Microsoft Teams.

Data Analysis Expressions (DAX): Power BI supports DAX, a formula language used to create custom calculations, measures, and aggregations, providing advanced analytical capabilities.

Overall, Power BI is designed to make data analysis accessible to a wide range of users, from business analysts to data scientists, and is commonly used in business intelligence and data analytics contexts to support decision-making and strategic planning.

Approach

A kippy instance was integrated with PowerBI Desktop, so that that the information in the kippy instance was pulled via the existing kippy data endpoint APIs – and then used within PowerBI Desktop to create bespoke visualisations on the returned data.

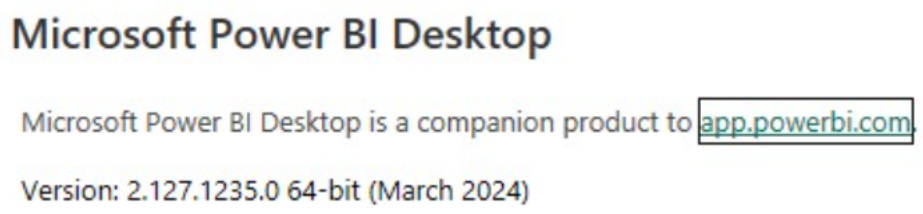
The integration between kippy and PowerBI was done using JWT authentication. Therefore, the JWT token would first be retrieved by PowerBI from the kippy token provider URL – and then passed through on all subsequent calls by PowerBI to the kippy data endpoint APIs.

To do this, PowerBI Desktop needed to be configured in a very particular way, with some bespoke "query" code provided to PowerBI Desktop.

One thing to note is that in a recent version of PowerBI Desktop, Microsoft put in an undocumented restriction, such that the "Authentication" header was no longer sent on calls made from PowerBI Desktop. This approach includes a work around for for that issue.

Prerequisites

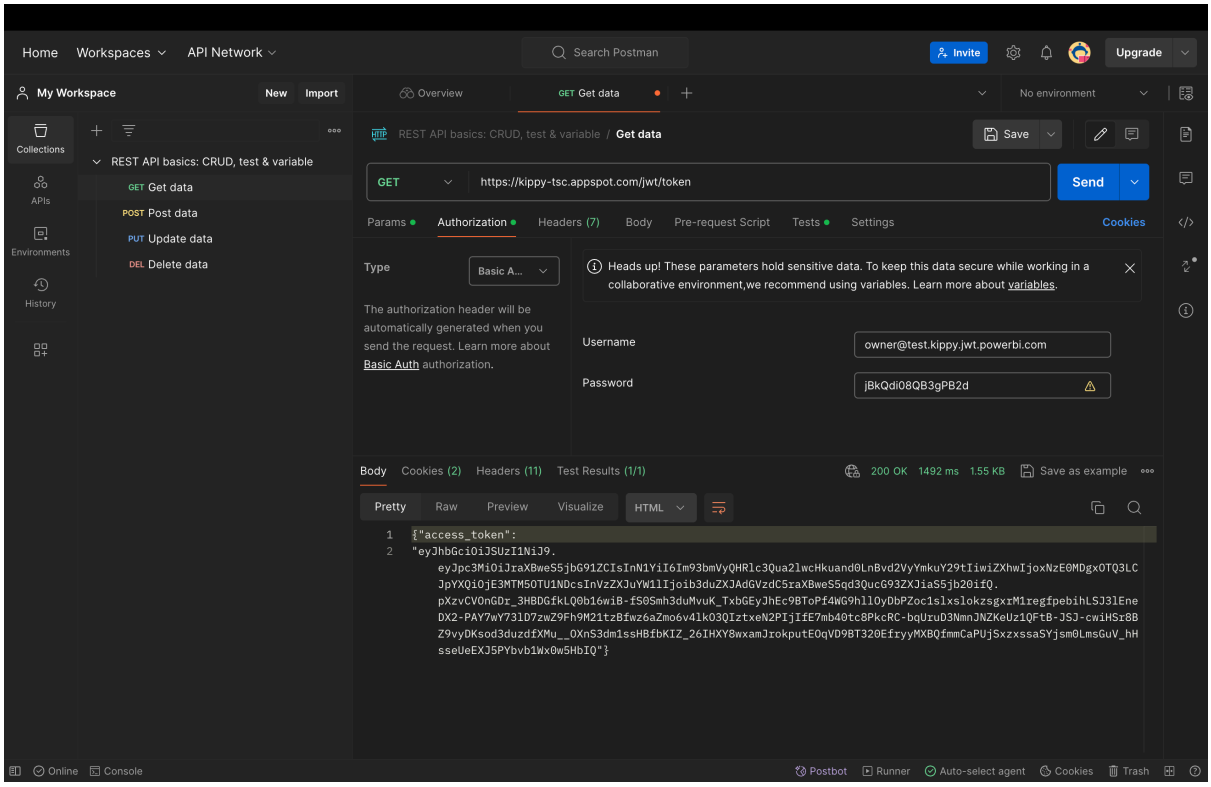
The POC was done using the following version of Power BI Desktop. Screens and functionality may differ for other versions.



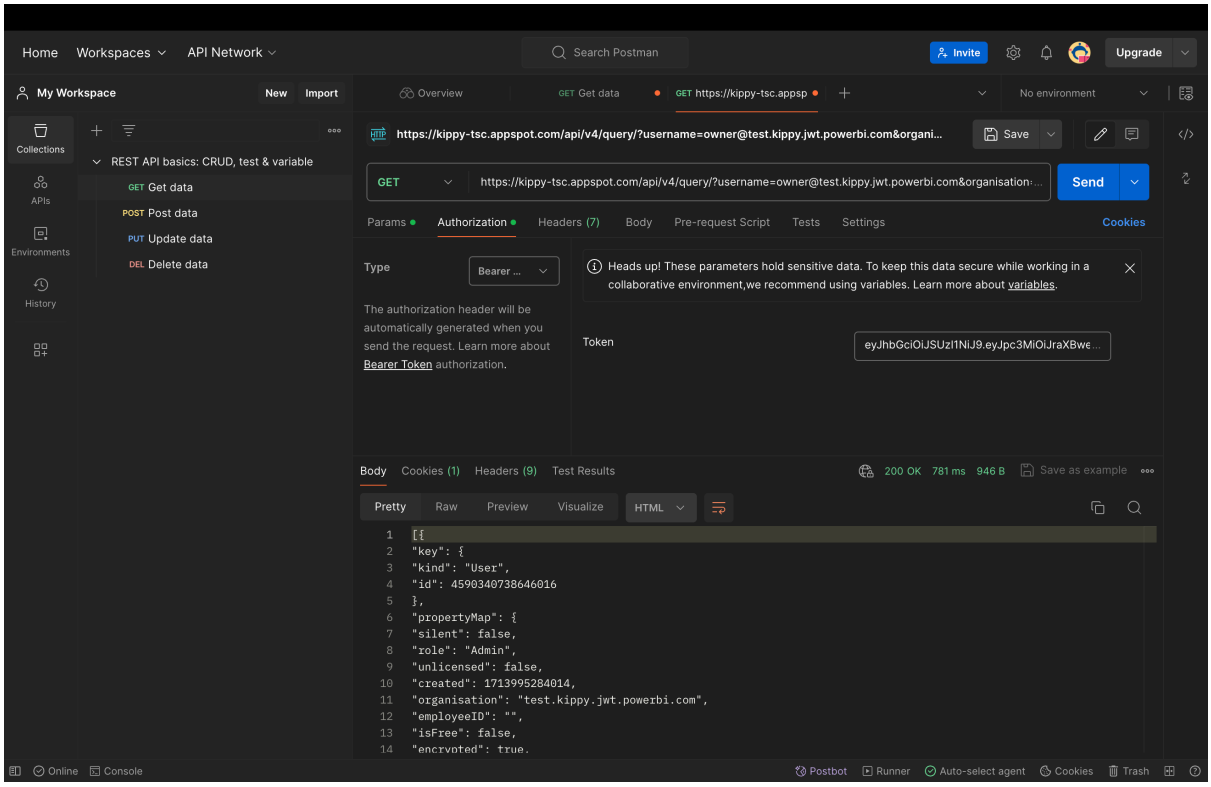
Also, the POC assumes you have access to a kippy instance with some existing data. If you need to, create a new kippy instance at www.kippy.cloud and use that to test your connectivity. The following uses kippy instance with username owner@test.kippy.jwt.powerbi.com and password jBkQdi08QB3gPB2d as an example.

To ensure the connectivity is working as expected, it is highly recommend to use a tool like postman.com to first ensure API access is working as expected outside of Power BI.

First, we call https://kippy-tsc.appspot.com/jwt/token with the username and password using Basic Authentication to get a token.

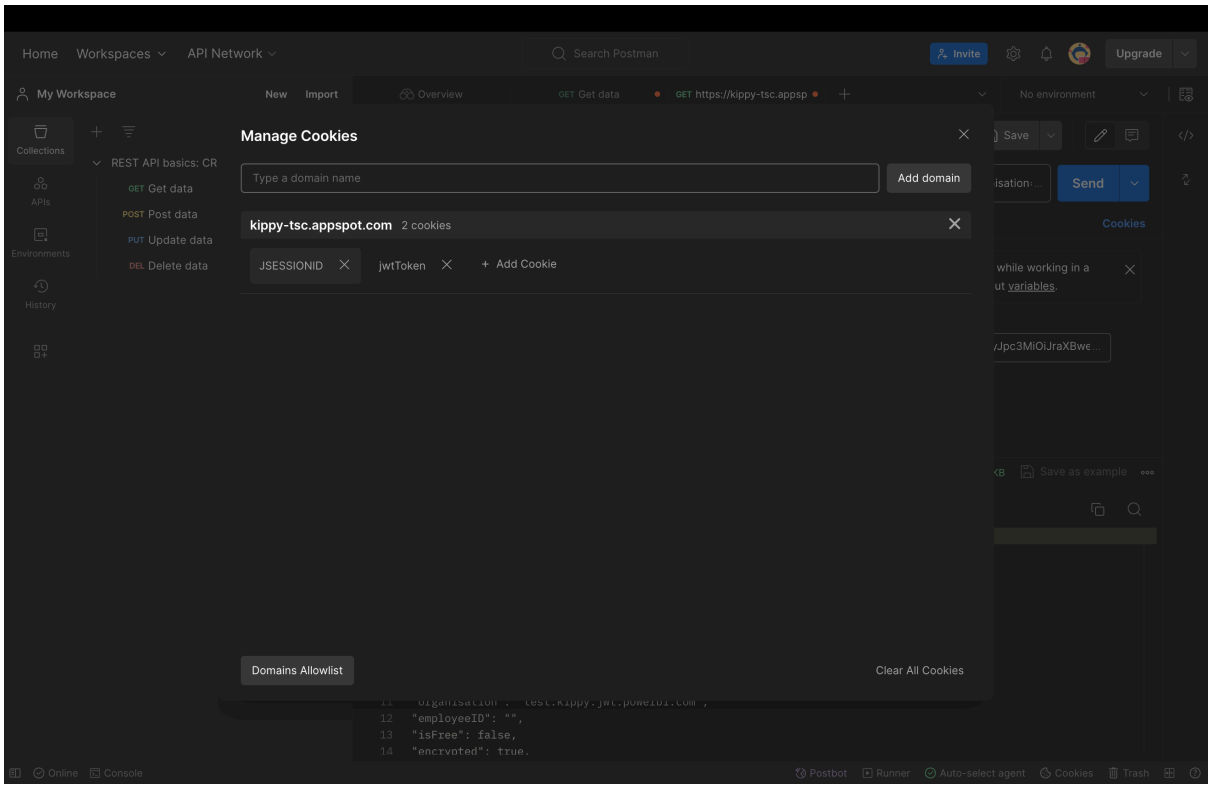


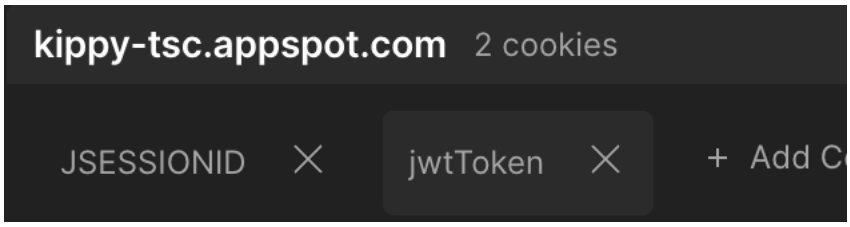
Next, we call the data endpoint API using Bearer token authorization.

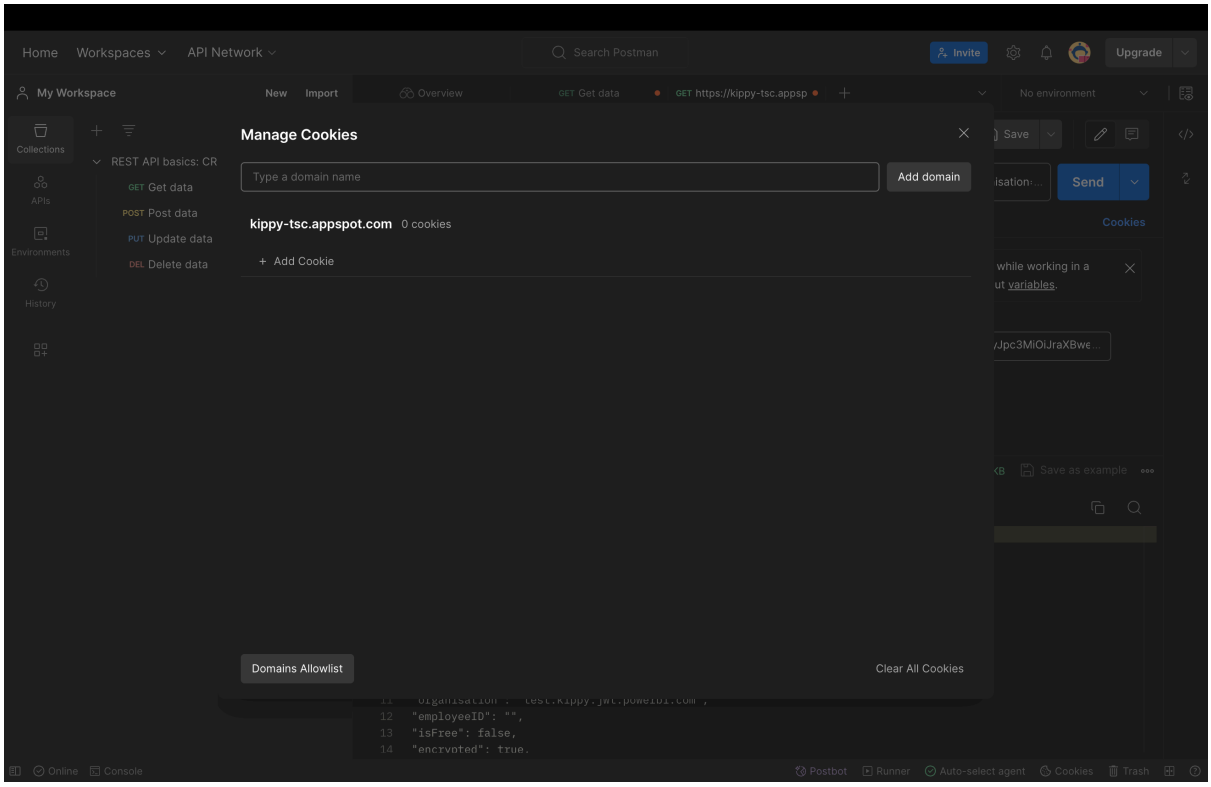


You can see the response is some JSON showing the users in the kippy instance.

By the way, make sure POSTMAN is not using any cookies when making the calls to the data endpoint API. If it is, delete those Cookies (by click the blue Cookies text near the top right, and delete the cookies by clicking the X next to the cookie names.

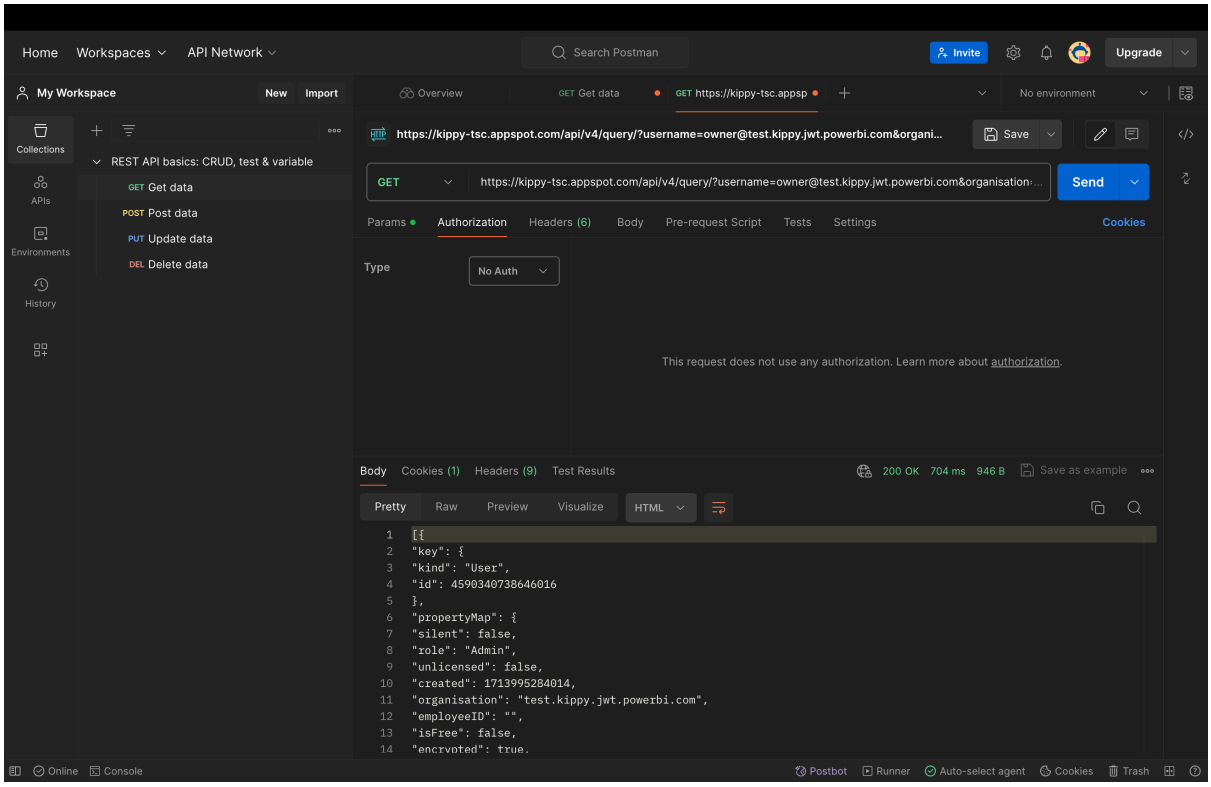




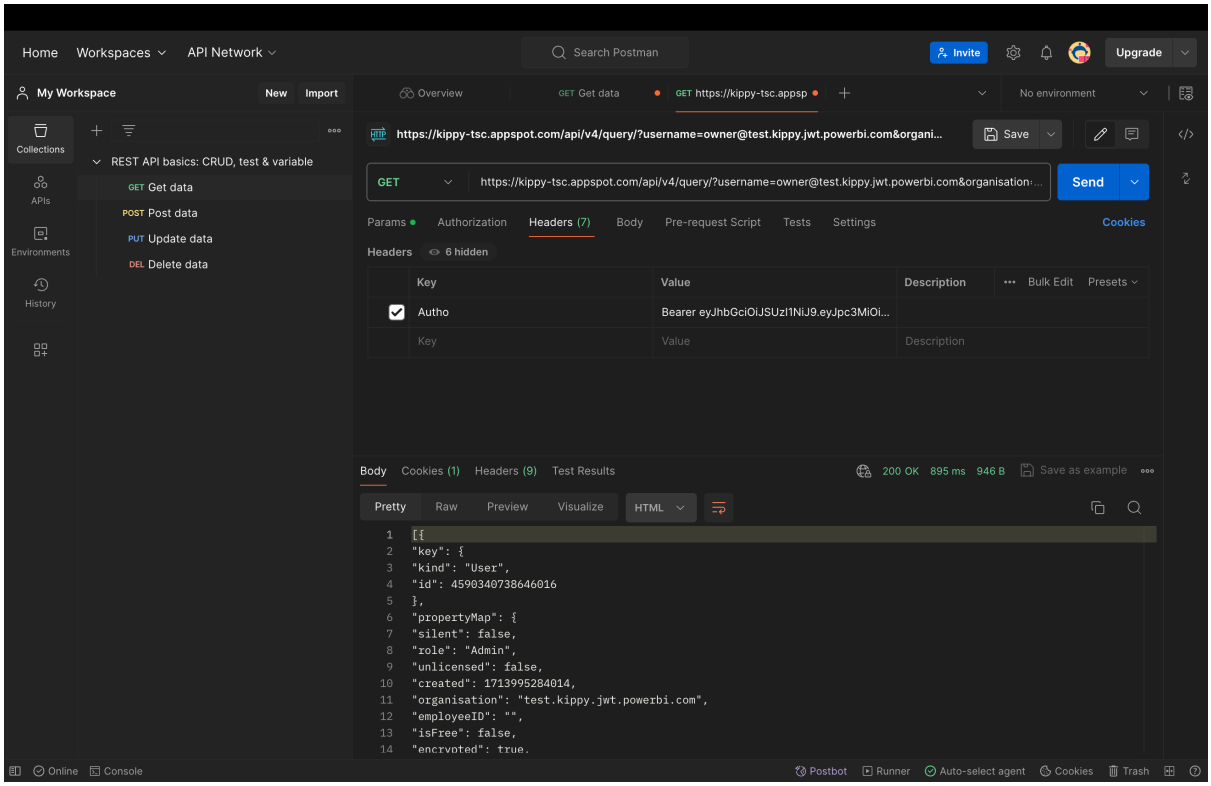


As mentioned earlier, Microsoft recently introduced an undocumented Power BI change that stopped headers called “Authorization” being sent by Power BI. Therefore, as a work around, instead we will pass an Authorization header called “Autho” to kippy, which has made a change to handle this situation.

So, once again, clear all the cookies in Postman and change the Authorization type to No Auth.



Now, pass in a header with the key “Autho” and the value starting with the word “Bearer”, followed by a space, followed by the token.



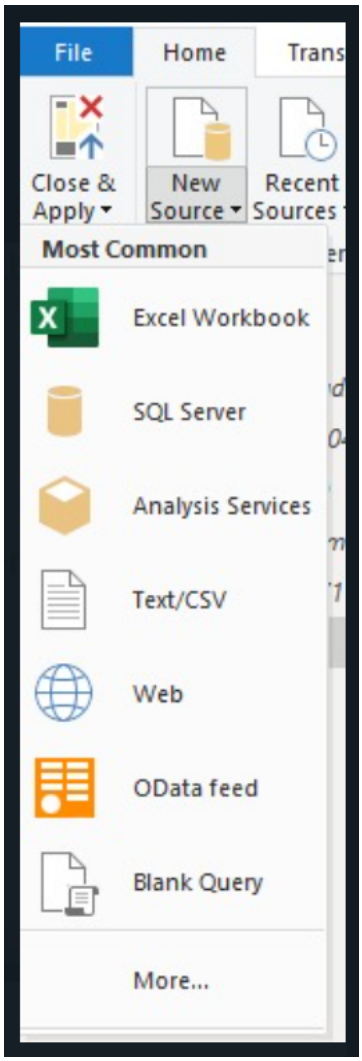
As you can see, the result is returned as before.

These steps confirm that the API is returning as we need it to. Next, we will do the same from within Power BI.

Integration

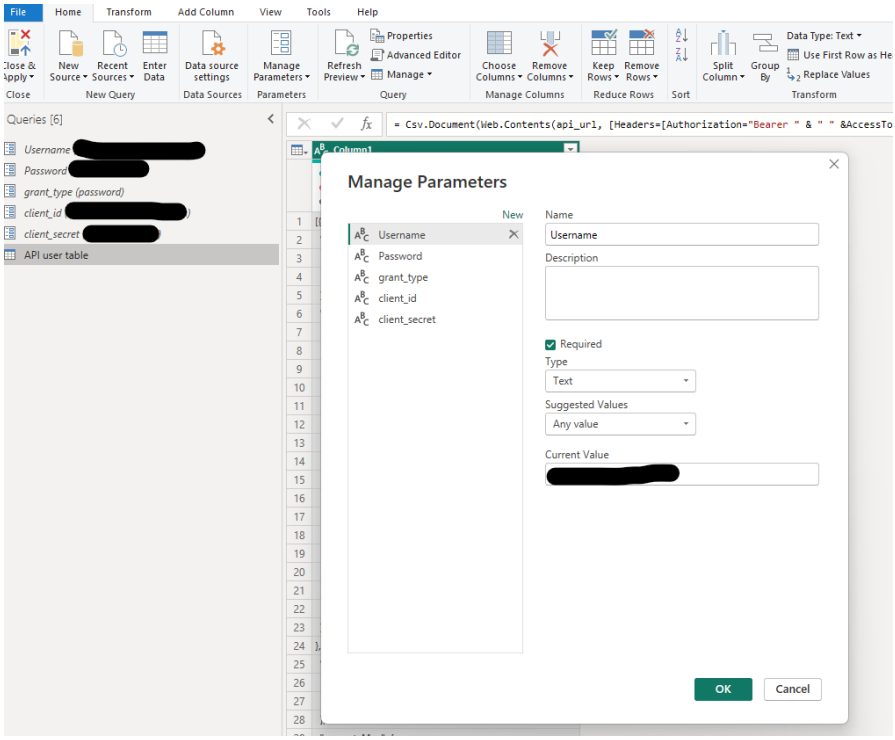
In this section, we will set up Power BI Desktop to connect to the same datapoint API using the same JWT authentication mechanism using a header named “Autho” with the value of “Bearer “ followed by the token.

1) The first thing to do in Power BI Desktop is to create a Blank Query.



2) Next, create the following parameters to be passed to the m query with the following name and values
   - Username = owner@test.kippy.jwt.powerbi.com
   - Password = jBkQdi08QB3gPB2d
   - Grant_type = password
   - Client_id = test.kippy.jwt.powerbi.com

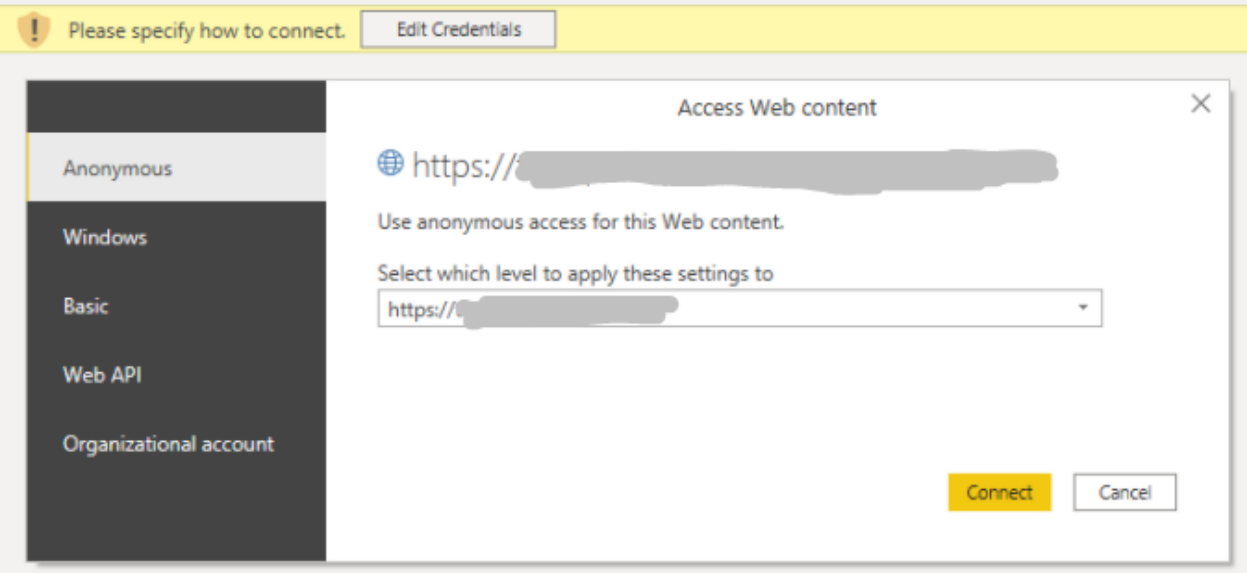
- Client_secret = test.kippy.jwt.powerbi.com.secret



3) Now, paste the following code in to the query.

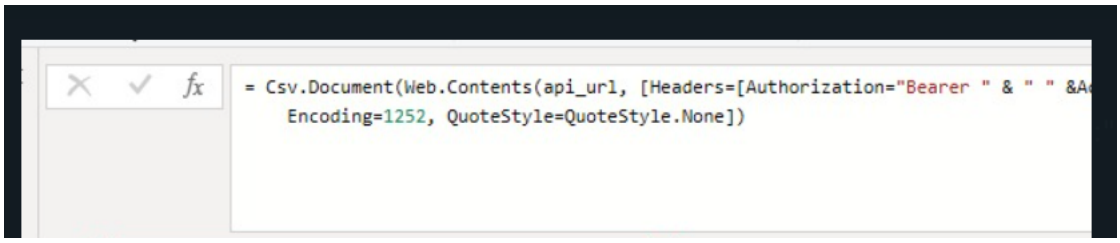
```
let
    //URLs
    api_url = "https://kippy-tsc.appspot.com/api/v4/query/?username=
owner@test.kippy.jwt.powerbi.com&organisation=test.kippy.jwt.powerbi.com&table=user",
//Insert your API endpoint
    token_url  ="https://kippy-tsc.appspot.com/jwt/token", //Insert your token URL
    //Client credentials
    client_credentials = "grant_type=" &grant_type& "&username=" &Username&
"&password="&Password&"&clienttype=User",
    //Get JSON Web Token via API
    EncodedCredentials = "Bearer "& Binary.ToText(Text.ToBinary(client_id & ":" &
client_secret), BinaryEncoding.Base64),
    Token_Response = Web.Contents(token_url, [Headers=
        [Autho=EncodedCredentials, #"Content-Type"="application/x-www-form-
urlencoded;charset=UTF-8"],
        Content=Text.ToBinary(client_credentials)]),
    //Get Access Token
    FormatAsJson = Json.Document(Token_Response),
    AccessToken = FormatAsJson[access_token],
    //Get Data
    GetJsonQueryAPI = Csv.Document(Web.Contents(api_url, [Headers=[Authorization="Bearer " &
" " &AccessToken]]),[Delimiter=";",
    Encoding= TextEncoding.Utf8, QuoteStyle=QuoteStyle.None])
in
    GetJsonQueryAPI
```

4) If Power BI prompt Please specify how to connect.
   a. Click the Edit Credentials button
   b. Make sure to select Anonymous
   c. Click Connect



5) Ensure the encoding is UTF8.

To do this you might have to change “Encoding=1252” to “Encoding= TextEncoding.Utf8”

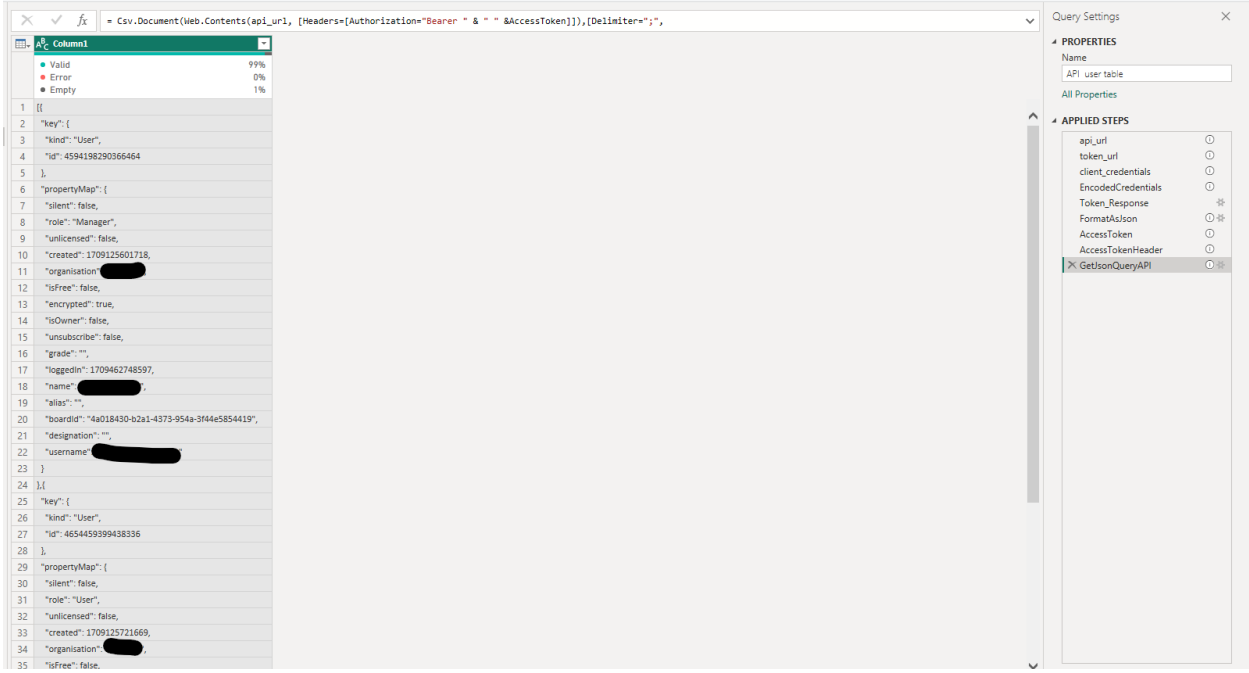


If you get an error like “Expression.Error: The import grant_type matches no exports. Did you miss a module reference?” follow this:

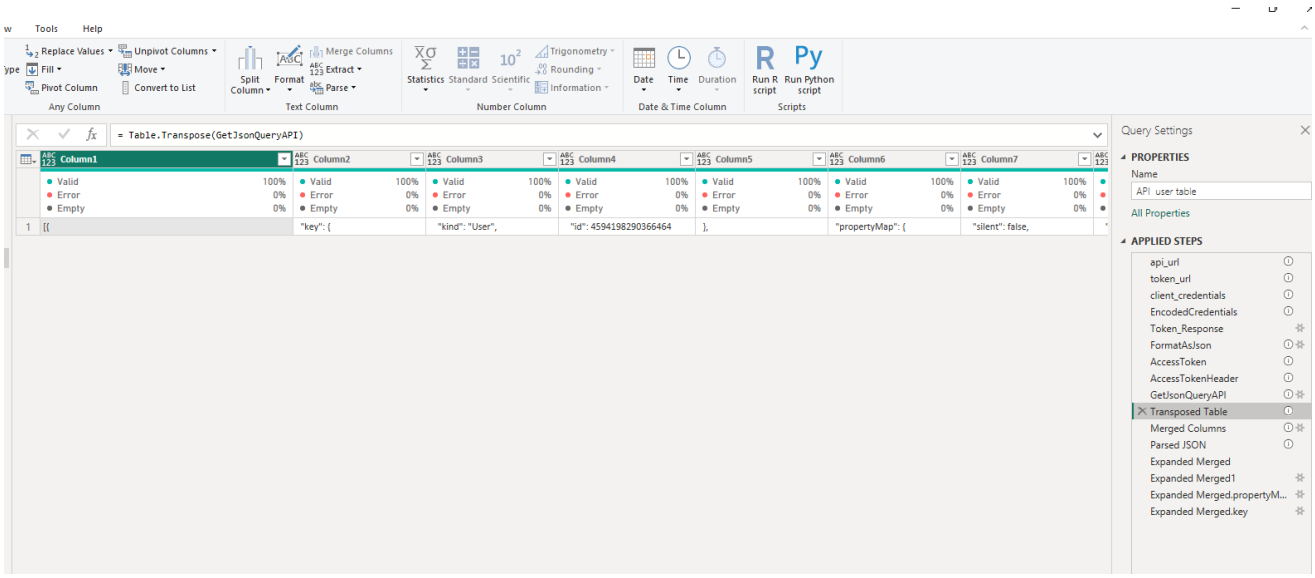
Solution: Ensure that the custom connector (`.mez` file) is placed in the correct directory (`Documents\Power BI Desktop\Custom Connectors`). Additionally, adjust Power BI's security settings to allow loading of custom connectors: help.piwik.pro +1

- Navigate to **File** > **Options and settings** > **Options** > **Security**.
- Under **Data Extensions**, select **(Not Recommended) Allow any extension to load without validation or warning.**
- Restart Power BI Desktop to apply the changes.

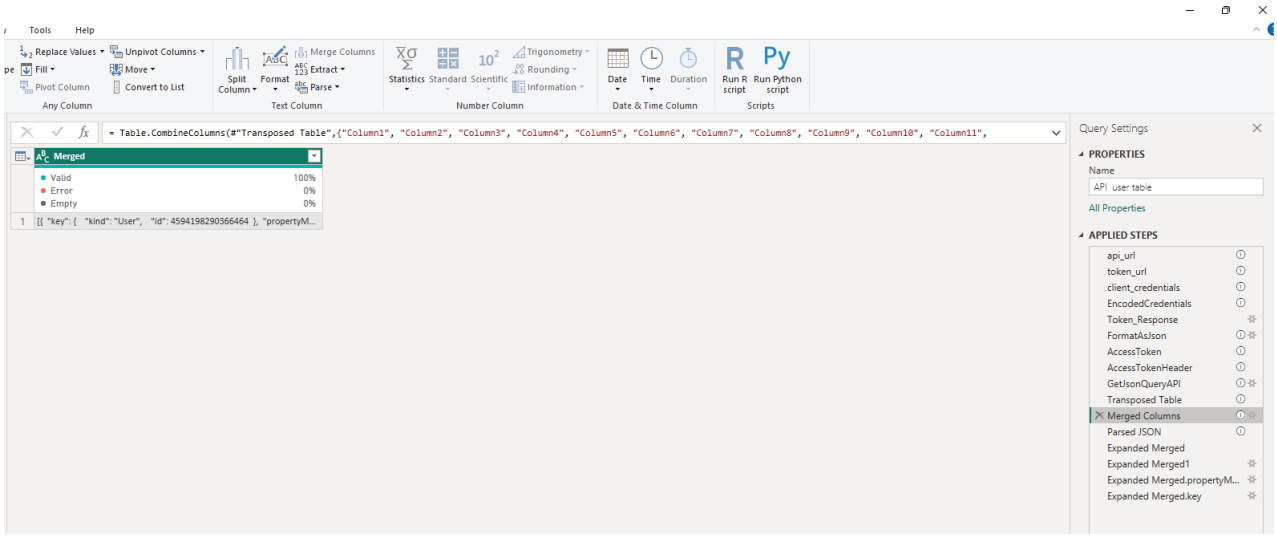
6) The result should be like below:



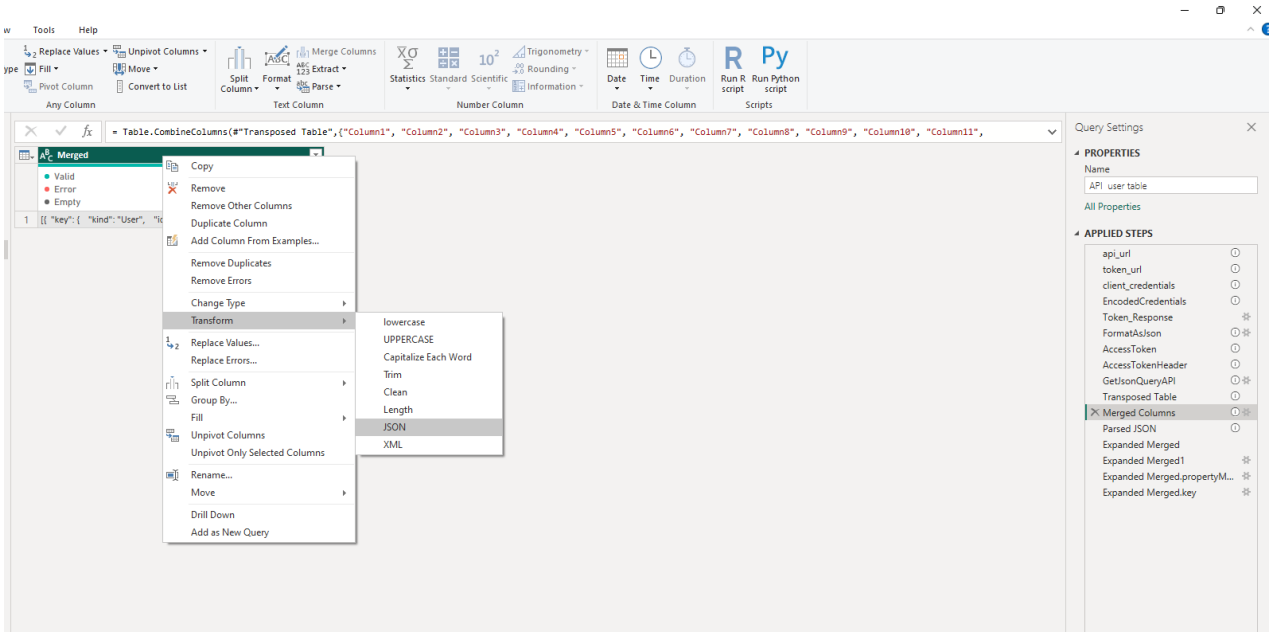
7) Next, we need to transfer csv data format and then to JSON format
   a. Transpose the table like so



   b. Merged the table to one column

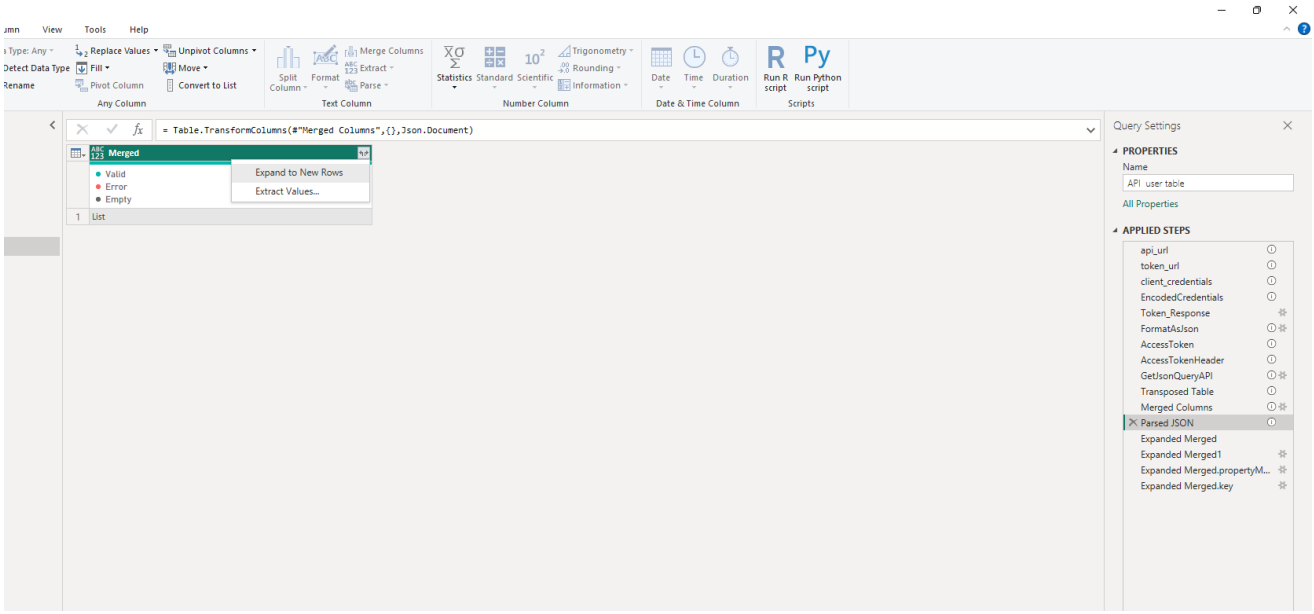


c. Convert the table now to JSON



d. Start expand the table up to your need of data

8) Now you can start use the data for your visualization



The query can now be used to create dashboards and visualisations in Power BI Desktop, which can be 'published' for other Power BI users to use and interact with.

What next?

You can now change the username and password to your live kippy instance to ensure the information is returned.

You can also create other Power BI Desktop queries for different datapoint APIs.

This should allow you to create visualisations based on the information from one or more queries and your other data sources.

Conclusion

Kippy APIs can be called using JWT tokens to pass information to Power BI Desktop.

Other mechanisms are also available for connectivity and access your data in kippy.

Contact us at support@kippy.me for more details.