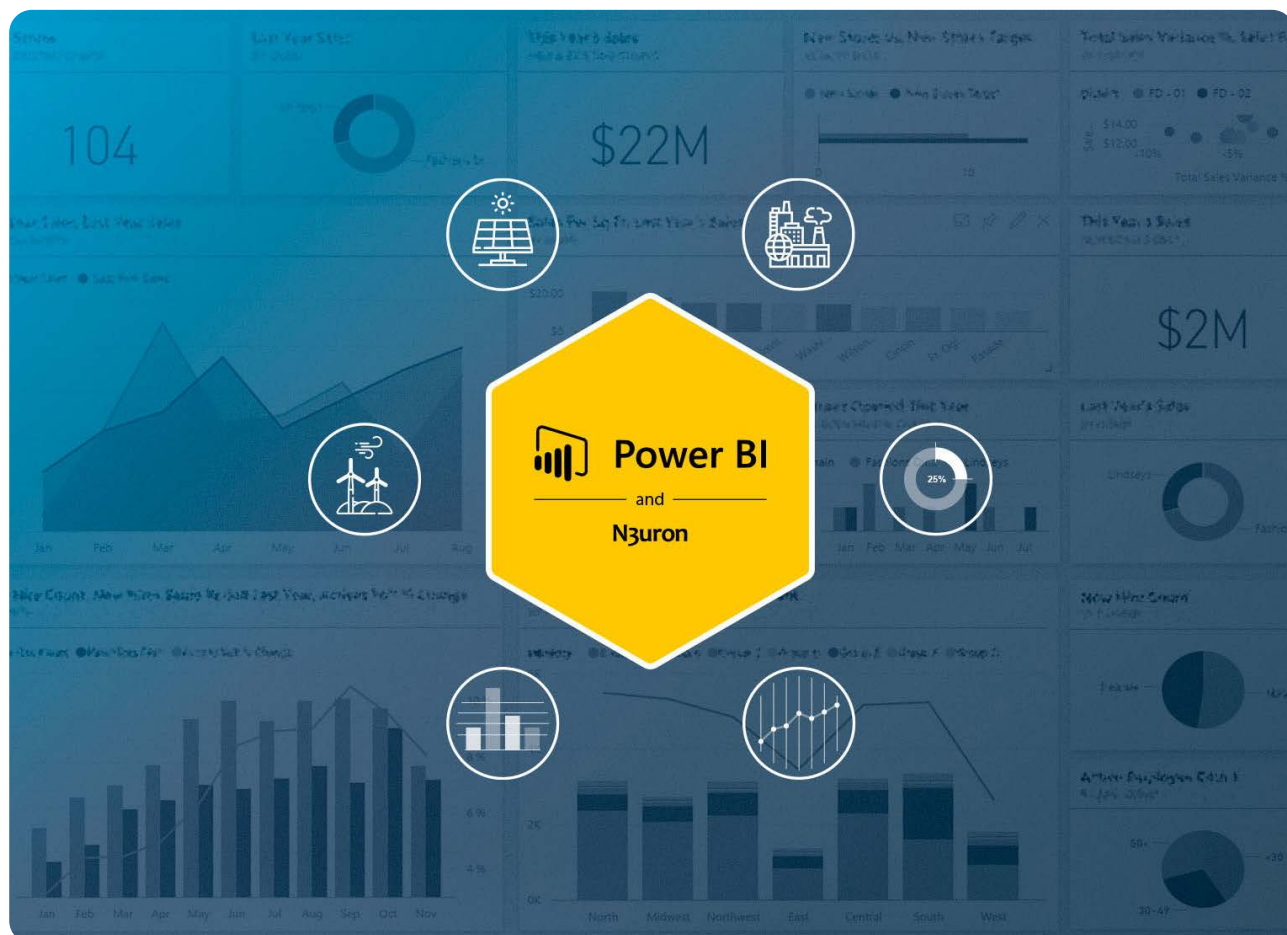


How to squeeze your industrial data with Power BI and N3uron



Power BI and N3uron: Overview

Over the last few years, Power BI has arguably become the preferred SaaS platform offering a data analytics and business intelligence solution for companies worldwide, helping organizations access tremendous amounts of data coming from very diverse sources in order to gain unprecedented insights. It's a modern tool that simplifies the process of creating interactive and comprehensive dashboards and reports that can be shared across the whole organization.

One of the main advantages of Power BI is the vast number of different data sources that can be accessed in order to bring all the necessary data together into one single report for answering any questions related to the business.

The generic workflow for any business intelligence tool, and thus also for Power BI, covers everything from obtaining data up to transforming it into actionable information that can be subsequently shared. This is comprised of the four main phases, as depicted in the below diagram.

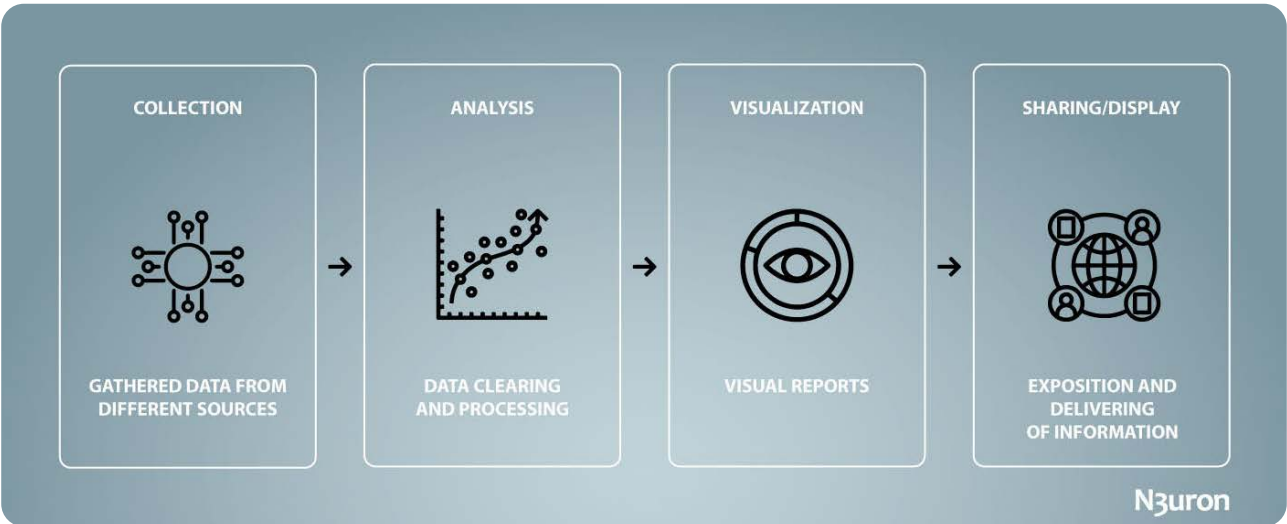


Diagram displaying business intelligence workflow

With an ever-increasing digitalized scenario, whereby numerous data sources such as sensors, control devices, and robots generate vast volumes of data, companies require a solution that allows them to make the most of this data and thus improve their competitiveness. It is precisely this requirement that makes this combined N3uron and Power BI bundle the perfect match.

This article will explain in detail how to extract operational data from industrial assets and make them available to Power BI using N3uron, and in particular, its powerful Historian and Rest Api Server modules.

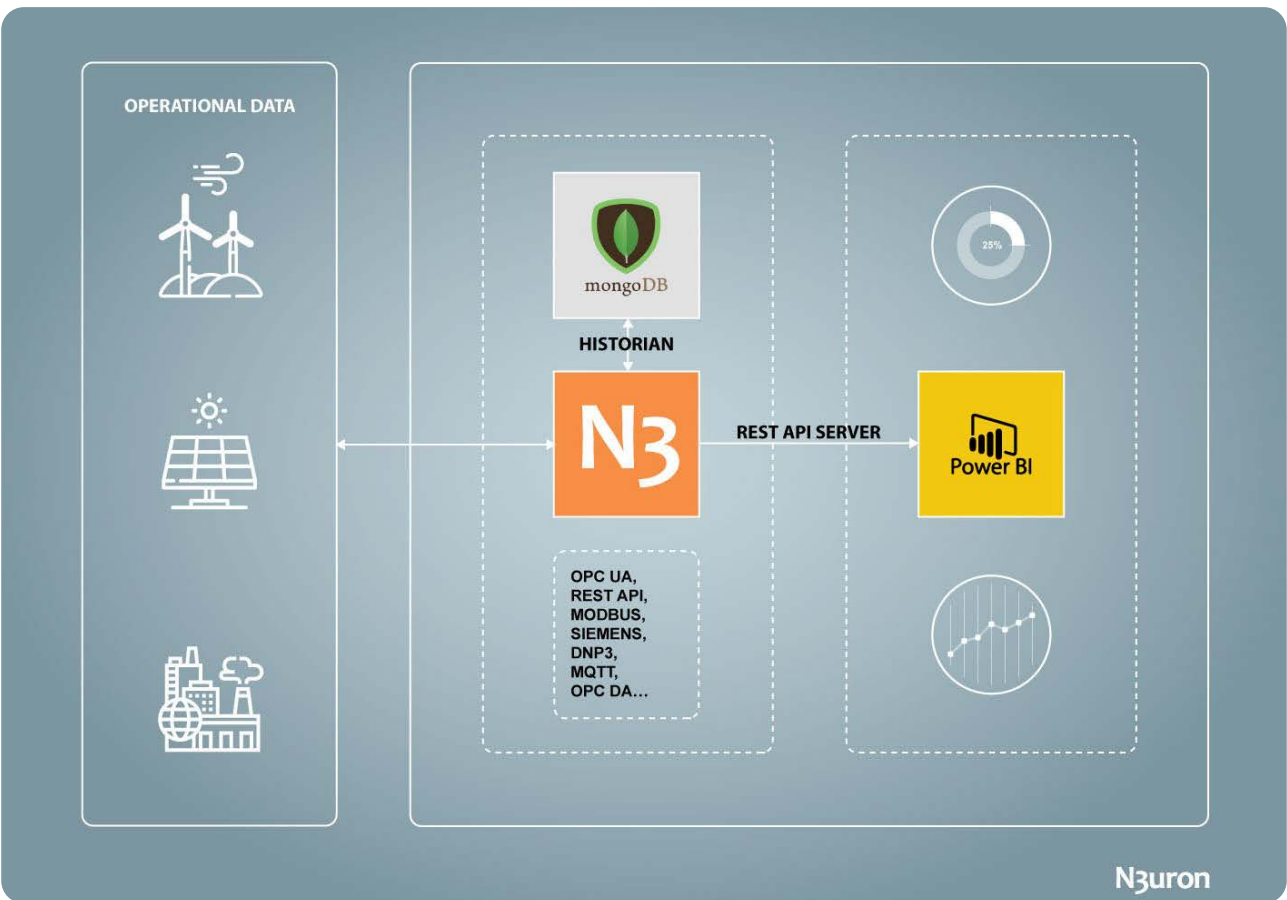


Diagram displaying operational data exchange between OT assets and N3uron using data visualization software Power BI

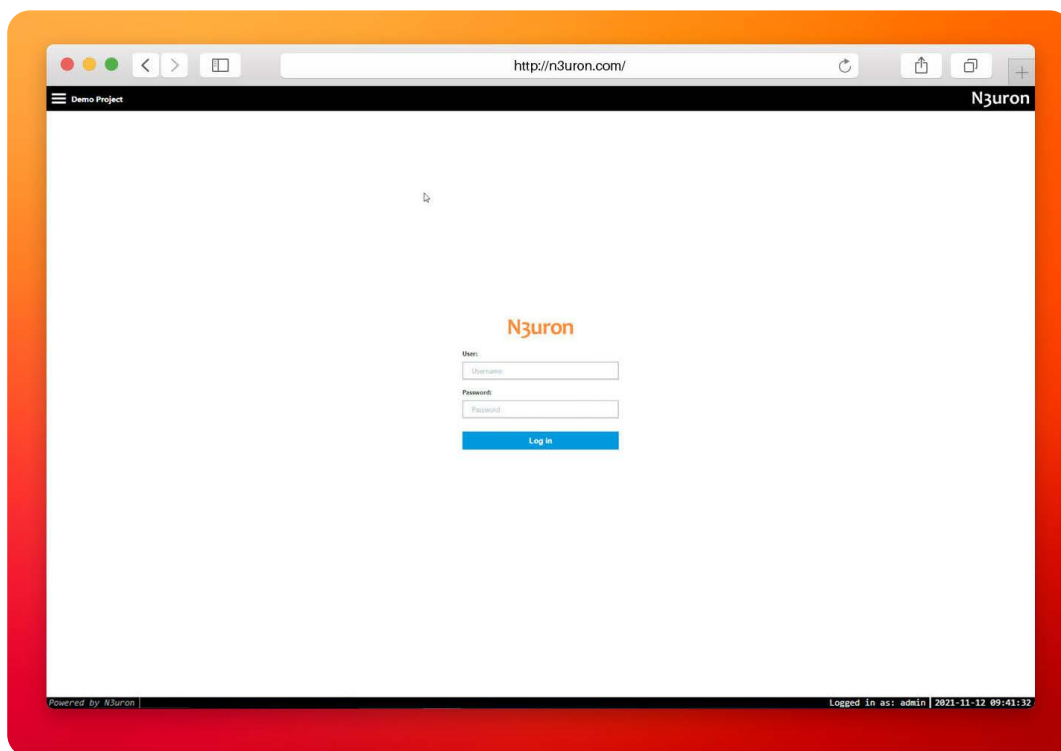
As shown in the image above, operational data is collected and stored in a MongoDB database, via N3uron, and this historical data is then exposed to Power BI Desktop via the Rest Api Server module.

Power BI and N3uron Requirements

It is assumed that you have already installed N3uron, if not you can download it at <https://n3uron.com/downloads/>. If this is the first time you are installing N3uron, our [Quick User Guide](#) will guide you through the entire installation process. Likewise, it is also assumed that you are already historizing some data in a MongoDB database using N3uron's Historian module. For more information on this, take a look at [Historian](#). You will also need to install [Power BI Desktop](#), the free Microsoft application that will allow you to create rich interactive reports with visual analytics.

Configuring the REST API module in N3uron's Industrial IoT Platform

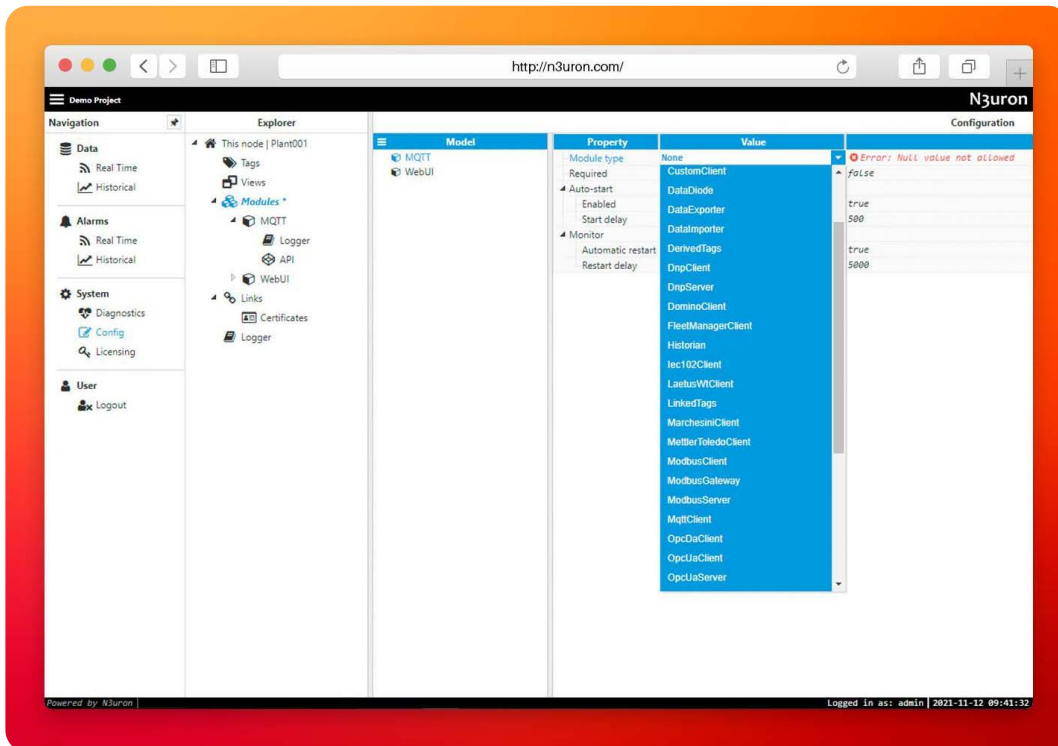
Log into the N3uron Industrial IoT Platform Using a Web Browser If this is the first time you are accessing N3uron, open your web browser and type <http://localhost:8003>. By default, **User** and **Password** are **admin** and **n3uron** respectively.



Screenshot displaying the log-in interface within N3uron's IIoT platform WebUI

Create a Module Instance Within N3uron's WebUI Interface

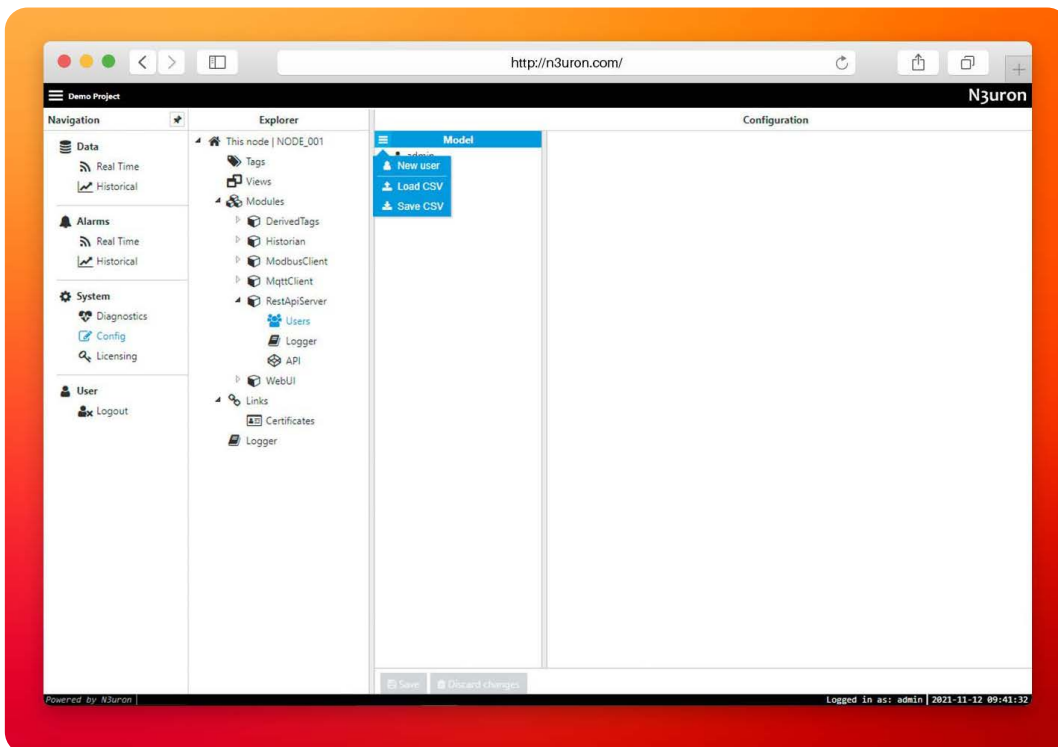
- **Step 01:** In the **Navigation** panel, select **Config**.
- **Step 02:** In the **Explorer** panel, select **Modules**.
- **Step 03:** Click on the **Model** menu and choose **New Module**.
- **Step 04:** The instance can be given any name. For this example, we will use **Rest Api Server**.
- **Step 05:** Set the **Module Type** property to **Rest Api Server** and leave the rest of the properties as they are, including the **Logger** and **API** settings (continue using their default values) and click on **Save**.



Screenshot displaying how to create an instance using N3uron's Rest Api Server Module panel

Configure N3uron's Rest Api Module within the WebUI's Explorer Panel

- **Step 01:** In the **Explorer** panel, select the **Rest Api Server** instance you have just created and click on the left arrow to expand the options.
- **Step 02:** Click on **Users**, go to the **Model** menu, choose **New user**, and give it a name. In this example we have used **admin**.

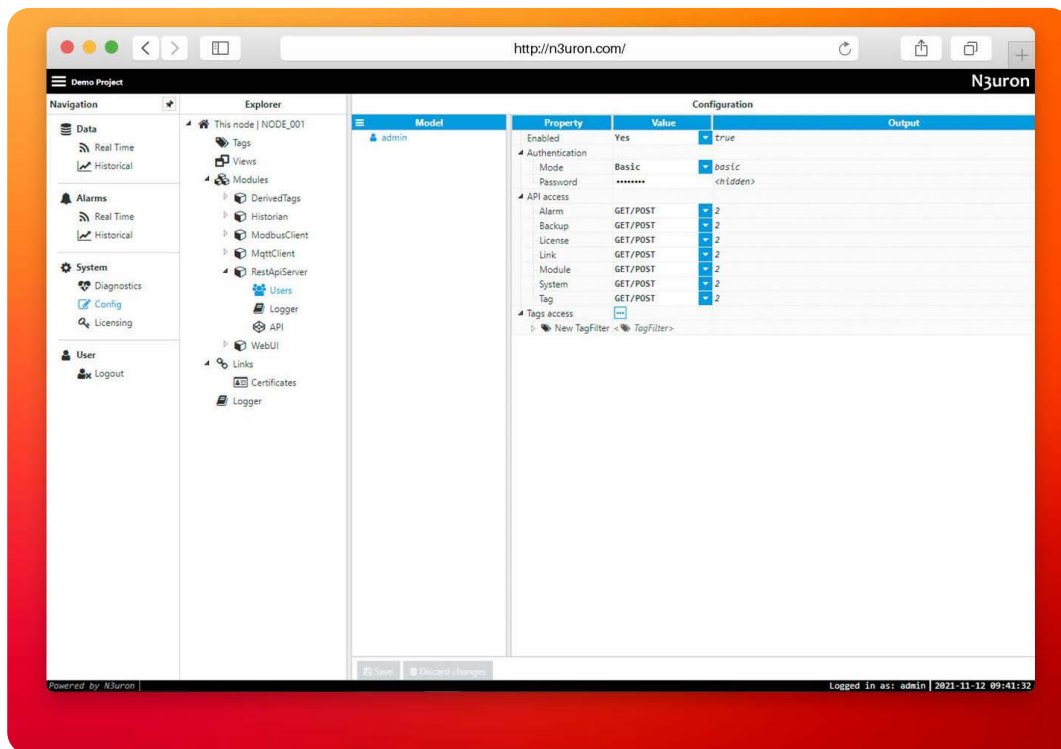


Screenshot displaying how to create a new user in N3uron's WebUI interface

- **Step 03:** Set the **Authentication Mode** to **Basic** and enter a **password of your choosing** in the **Configuration** panel.
- **Step 04:** Expand the **API Access** section, click on the **Tag** dropdown menu, and select **GET/POST**.
- **Step 05:** Click on the **Tag Filter** button, select **New Tag Filter**, and change the default name. In this example, we have used **Filter**. Leave **Mode**, **Path**, and **Regex pattern** as their default values.

With these configuration settings, every tag that is configured in N3uron will now be exposed through the REST API.

Screenshot displaying user configuration settings in N3uron's WebUI interface



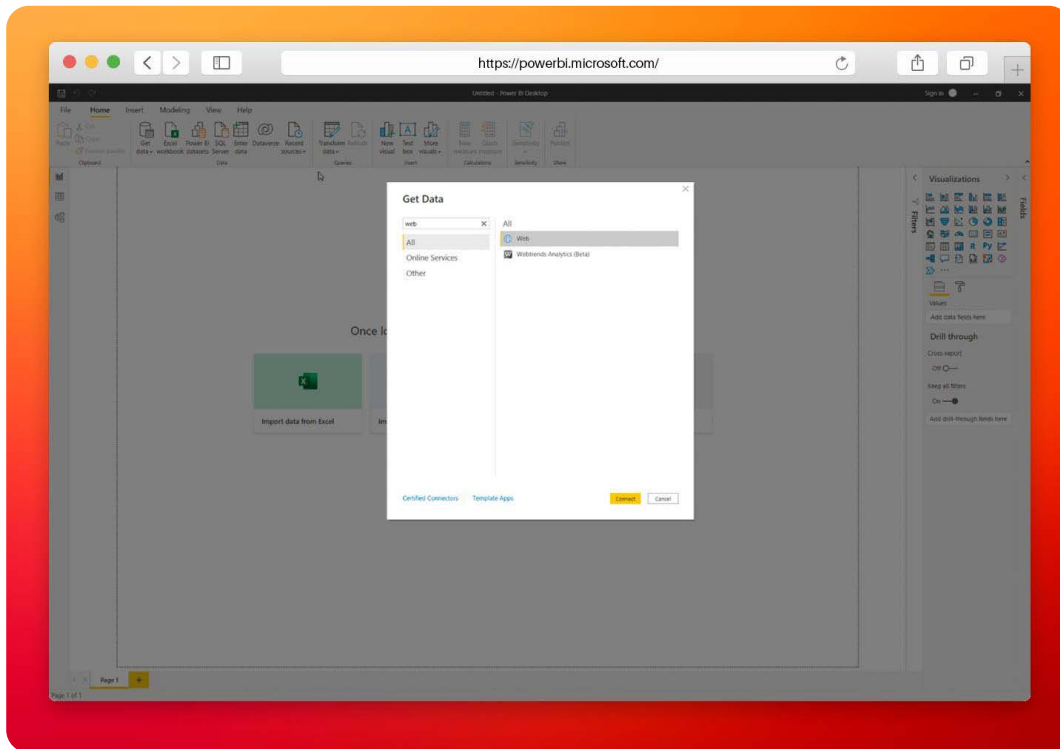
Configure Microsoft Power BI Desktop

Loading data into Power BI data visualization software

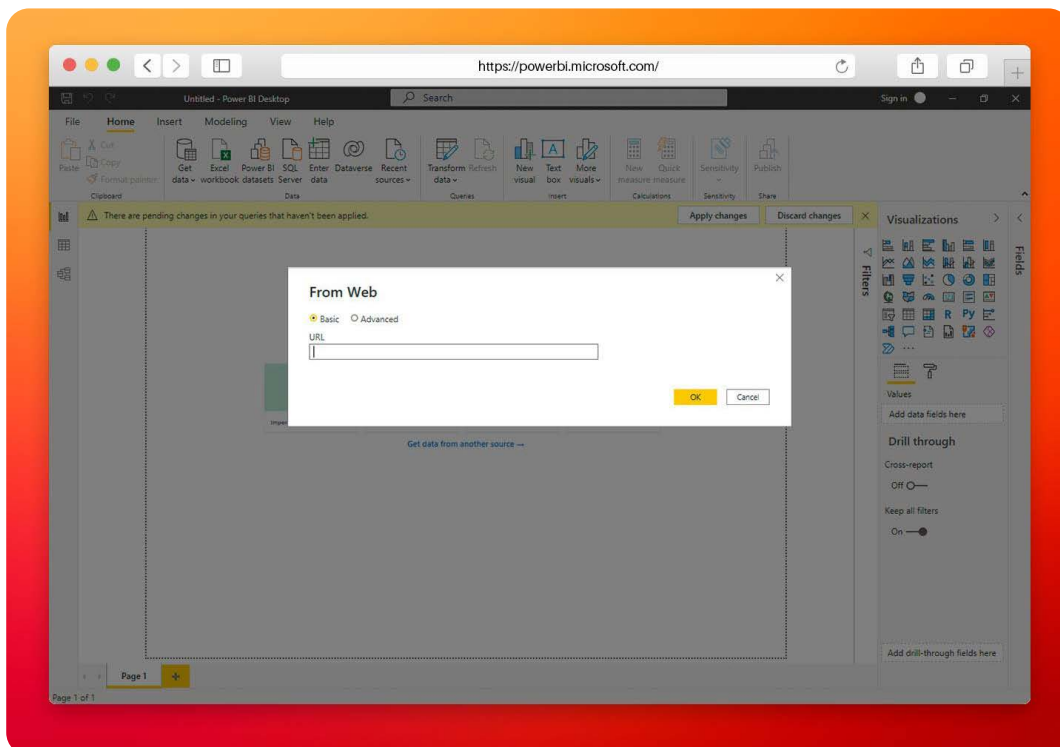
- **Step 01:** Open **Power BI Desktop** and click on **Get Data**.
- **Step 02:** Select **Web** from the available options.
- **Step 03:** We will then be prompted to enter a URL, in our case, given that we want to access historical data, we have entered `https://<IP_address>:<HTTPS_port>/tag?cmd=historyMany&paths=<path>&start=<start_date>&end=<end_date>`. For this example, we are going to use the historical values of two random sine and cosine functions. `http://10.101.3.210:3003/tag?cmd=historyMany&paths=/Derived/cos,/Derived/sin&start=2021-01-01&end=2030-12-31`.

Since at the time of writing this article, it is not possible to parameterize the start and end dates in Power BI, we have used a couple of dates that will allow us to load all the historical data every time we refresh the data in Power BI.

For more information about the available commands in **N3uron's Rest Api Server**, take a look at our [Rest Api Server](#) manual and download our [Postman library](#), which includes plenty of examples.

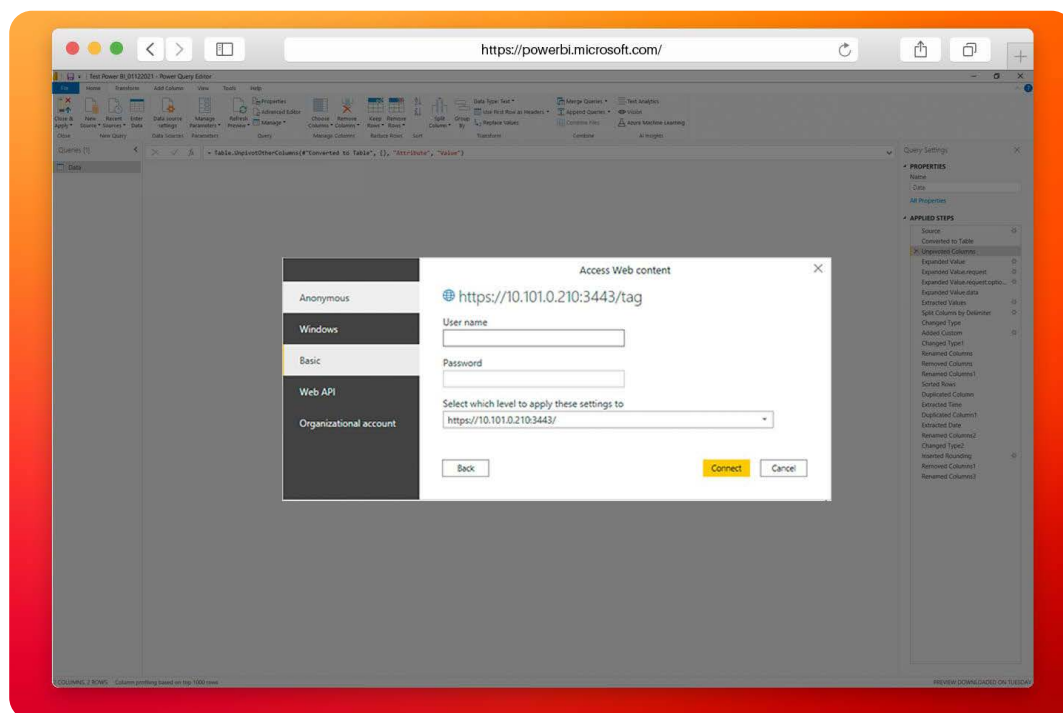


Screenshot displaying the “Get Data” window in the Power BI platform.



Screenshot displaying data import via URL in the Power BI platform.

- **Step 04:** After entering the URL, we will be prompted to enter the authentication credentials we previously created in **N3uron** for the **Rest Api Server** user. Follow this step and click on the Connect button.



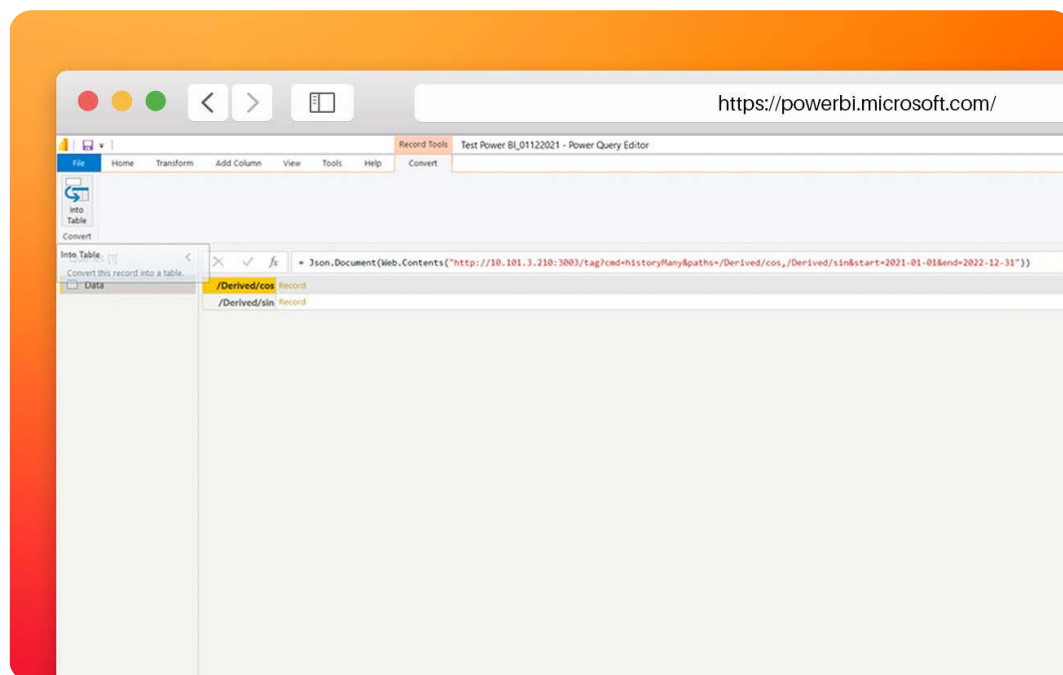
Screenshot displaying authentication process within the Power BI platform.

- **Step 05:** Select the data you want to import and click on **Load** to visualize it in Power BI. From this point on, you will be able to work with the operational data loaded from the Rest API Server in Power BI. See below for a small introduction on how Power BI works using an example report, which will allow us to graphically visualize the data imported from N3uron.

Transforming data using the software Power BI

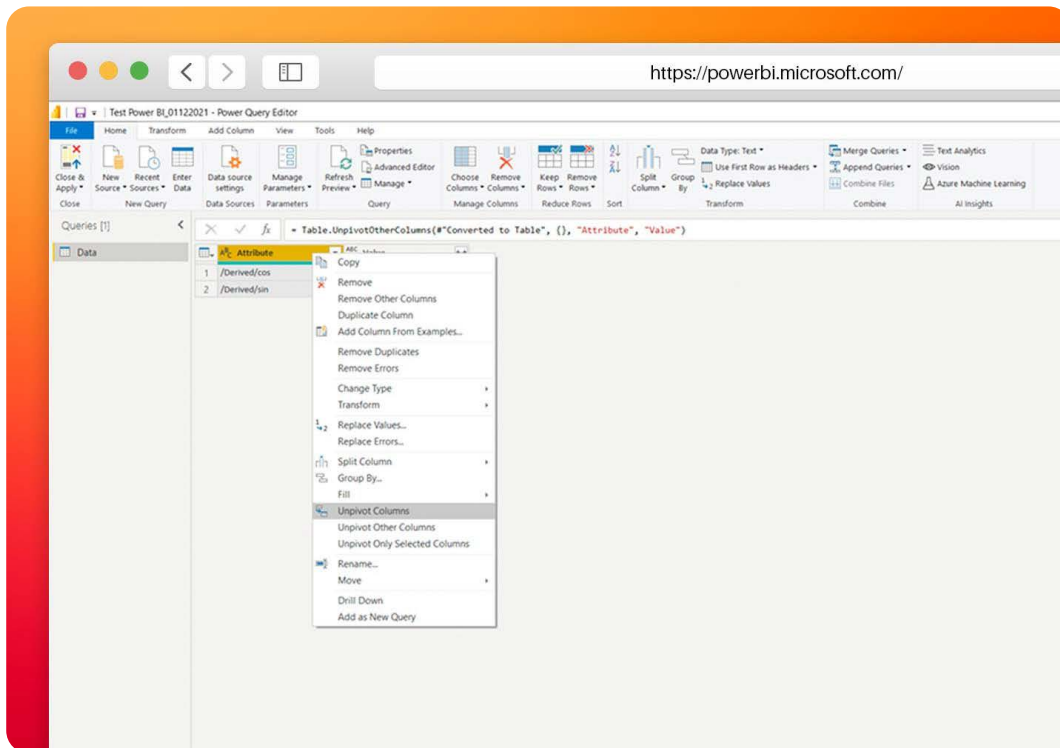
Once the data stored in MongoDB has been imported into Power BI, a few further steps need to be taken in order to transform this data before creating the final report.

- **Step 01:** First, click on the **into table icon** (located in the top left-hand corner) to transform the loaded data into a table.

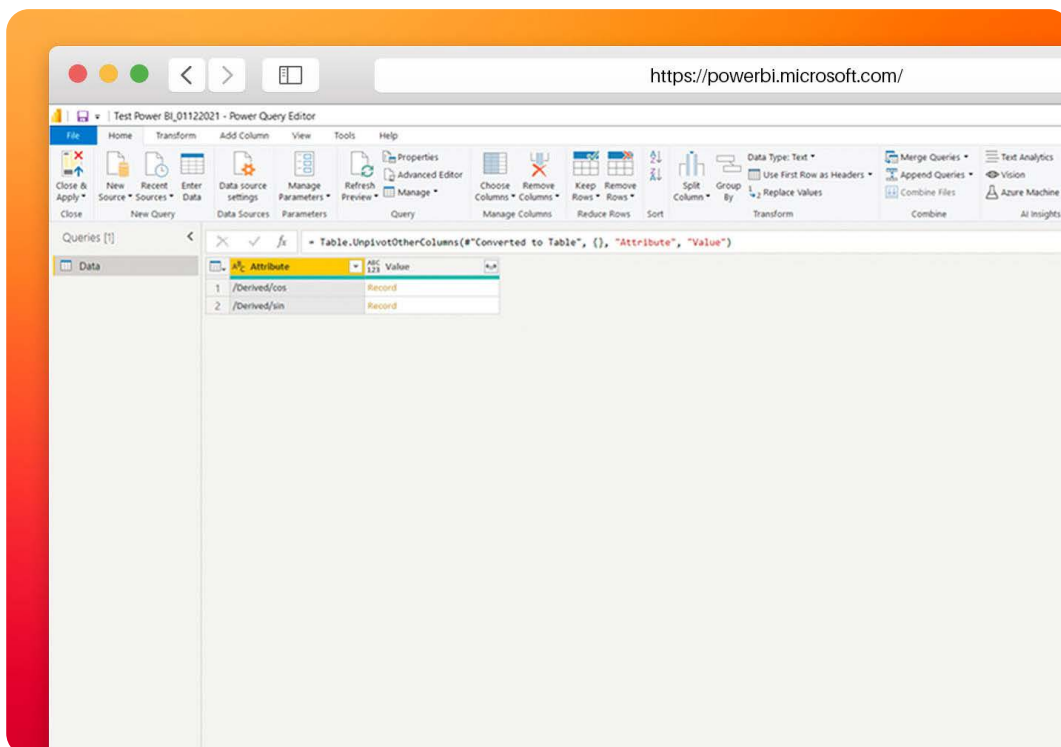


Screenshot displaying data conversion process in the Power BI platform

- **Step 02:** Right click on the first column name and select **Unpivot columns**.

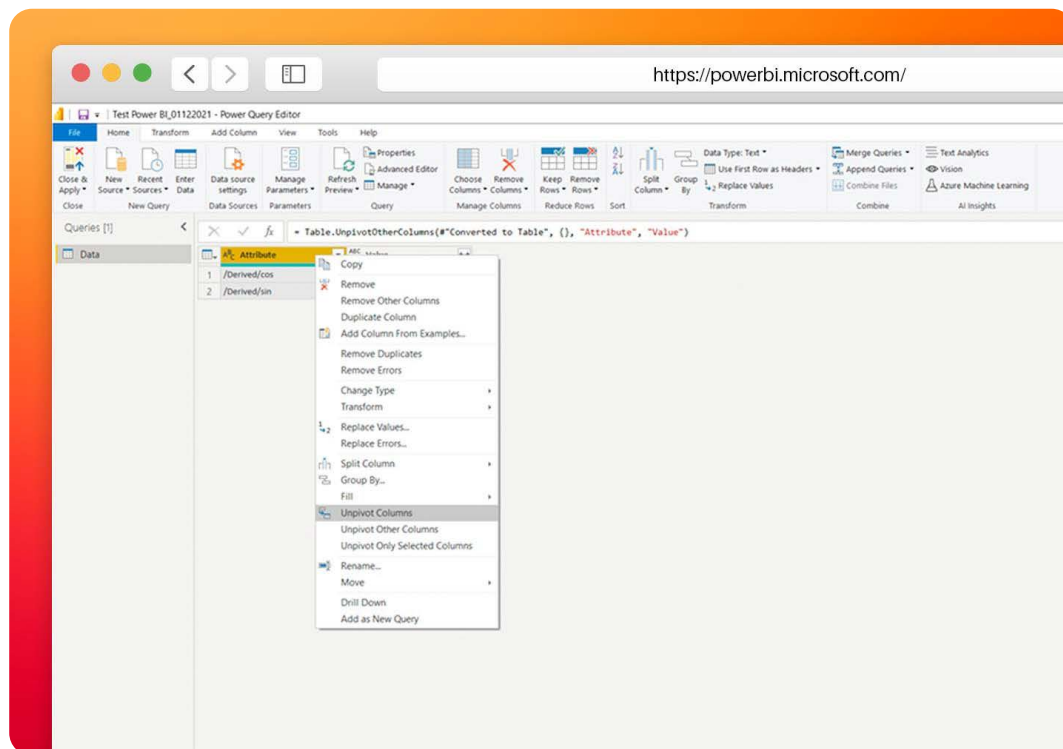


Screenshot displaying the unpivoted columns command within the Power BI platform



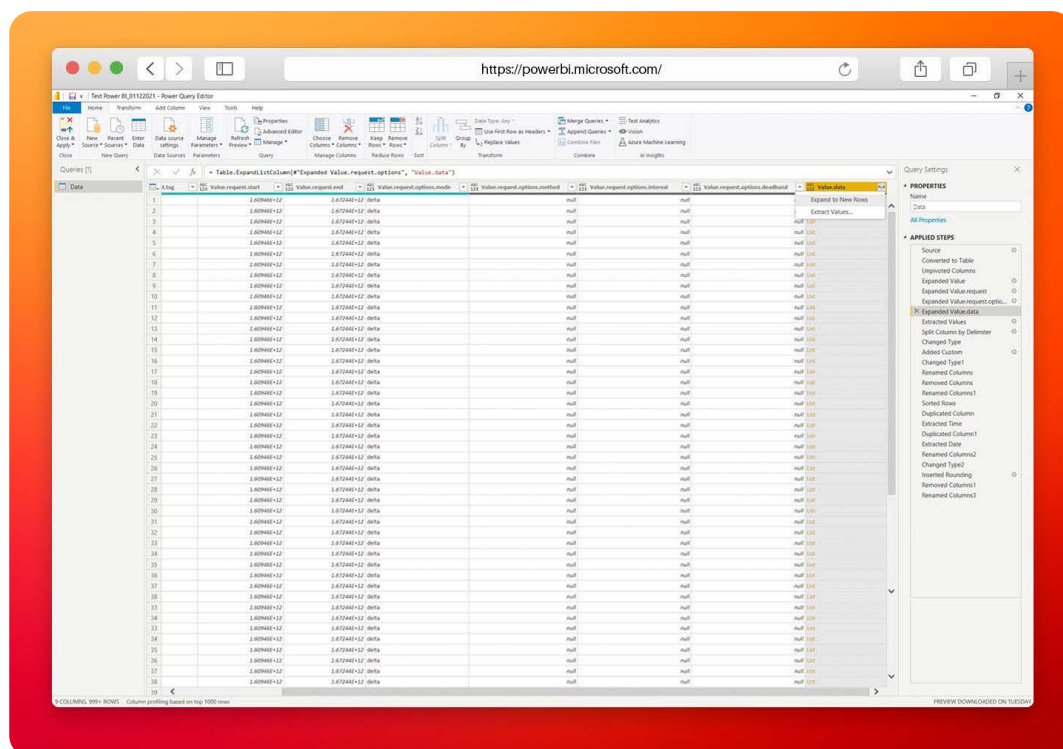
Screenshot displaying the unpivoted columns

- **Step 03:** Click on the **Expand values** button for every column containing **Records and Lists** as many times as necessary until you obtain the **Value.data** column that will show the values **Lists**. The order we followed in the below example to expand values was **Value**, **Value.request**, and **Value.request.options**.

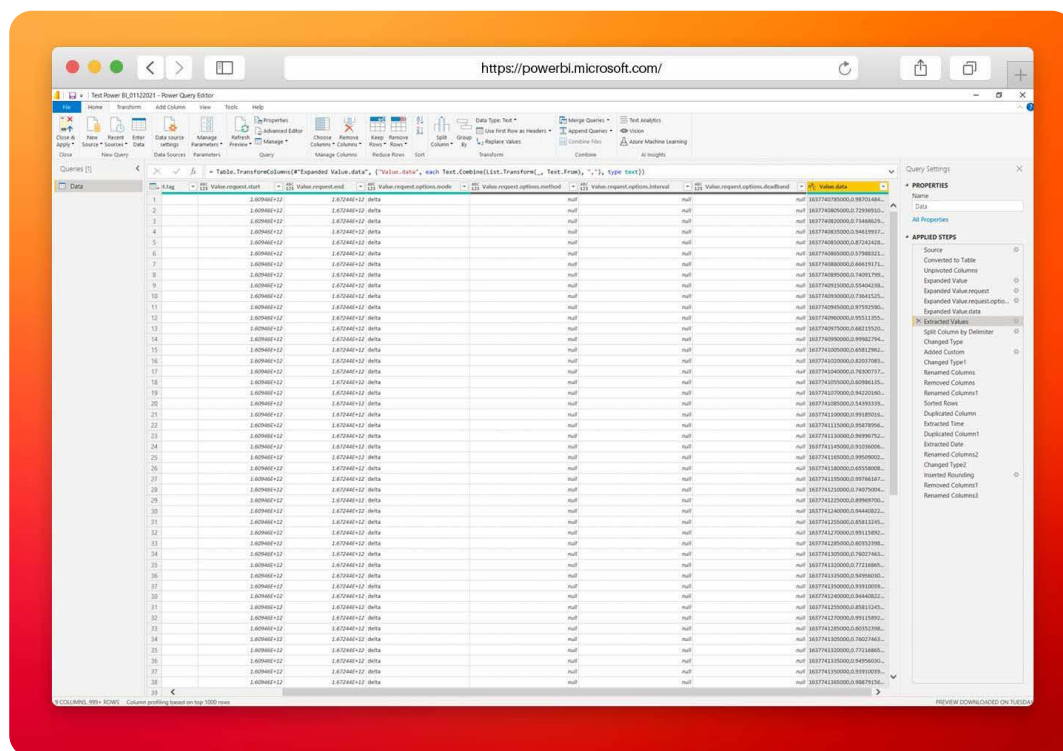


Screenshot displaying expand values action in the Power BI platform

- **Step 04:** Finally, click on the **Expand** button for **Value.data** and select **Extract Values**.

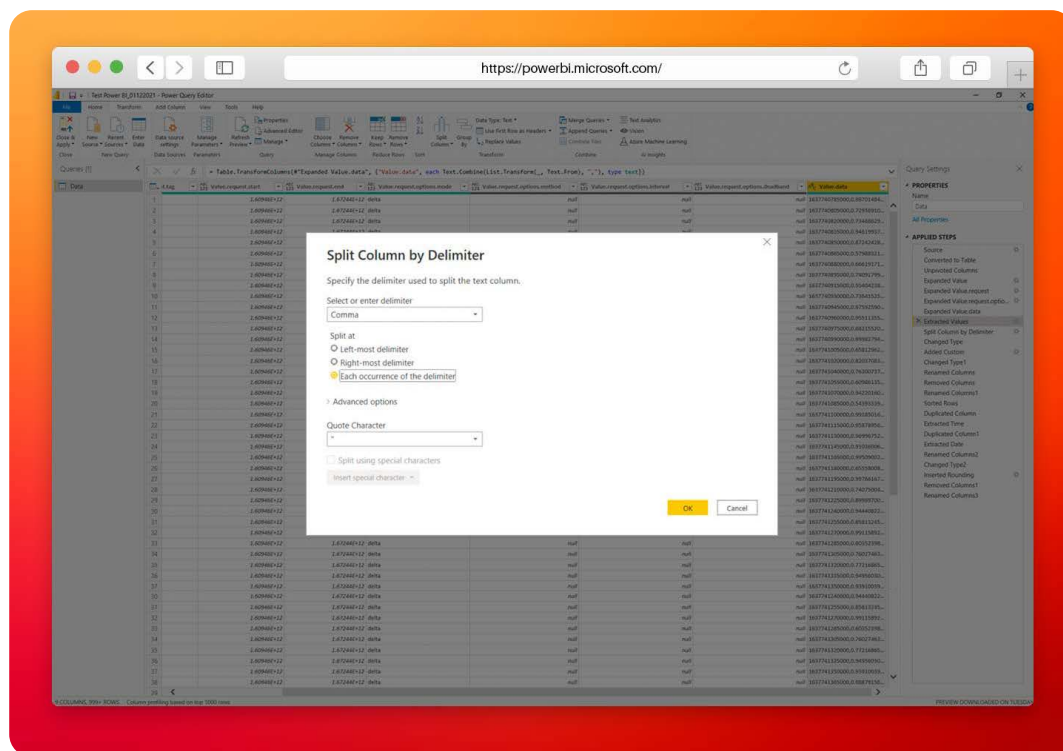


Screenshot displaying value extraction in the Power BI platform



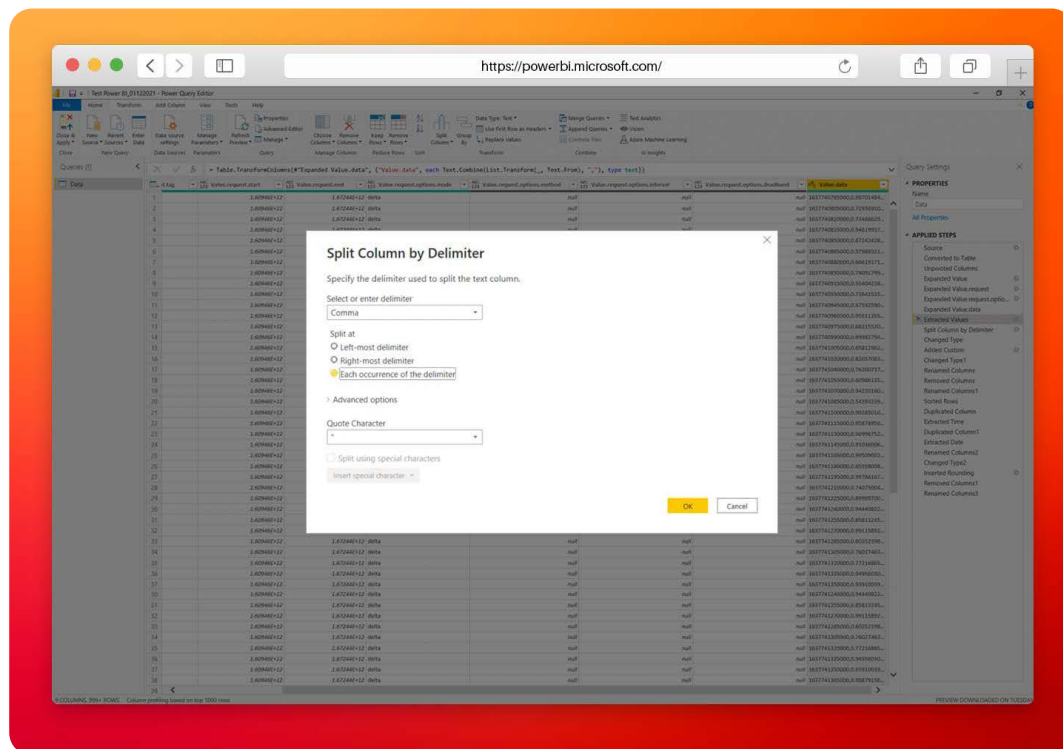
Screenshot displaying extracted values in the Power BI platform

- **Step 05:** The extracted values need to be split in order to separate the value itself from the timestamp. To do so, click on the **Value.data** column and select **Split Column**. In the popup window, select **Comma** as the delimiter and click on the **OK** button. By doing this, two new columns will be created labelled **Value.data.1** and **Value.data.2**.



Screenshot displaying the "split column by delimiter" panel in the Power BI platform

- **Step 06:** The timestamp values from the **Value.data.1** column will also need to be converted from Unix to Date/Time format, following the below steps:
 - A:** In the top menu, select the **Add Column** tab and click on **Custom Column**.
 - B:** Give it a name. In the example, we have used Date/Time.
 - C:** Enter the following formula, `#datetime(1970, 1, 1, 0, 0, 0) + #duration(0, 0, 0, [Fecha Unix]/1000)`.
 - D:** Click on the **OK** button.
 - E:** Change the new column's type to **Date/Time**.



Screenshot displaying the “custom column” panel in the Power BI platform

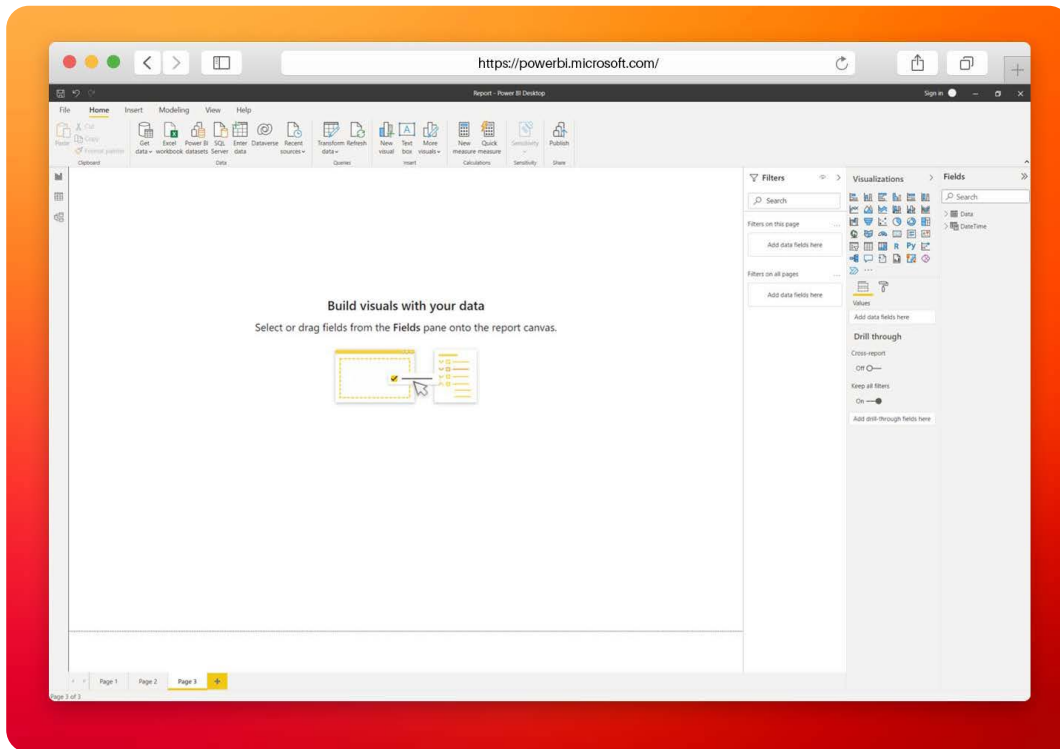
- **Step 07:** Remove all columns containing null values and keep only those which contain useful information, renaming them according to your preferences. In the example, we have only left three columns; **Tag**, **Value**, and **Date/Time**.

Creating a model or report using Power BI software

The next few steps will demonstrate how to create a new model based on the previously imported and transformed data, and finally, how to create a report that will allow users to obtain deeper insights about their business and hopefully come to some actionable conclusions.

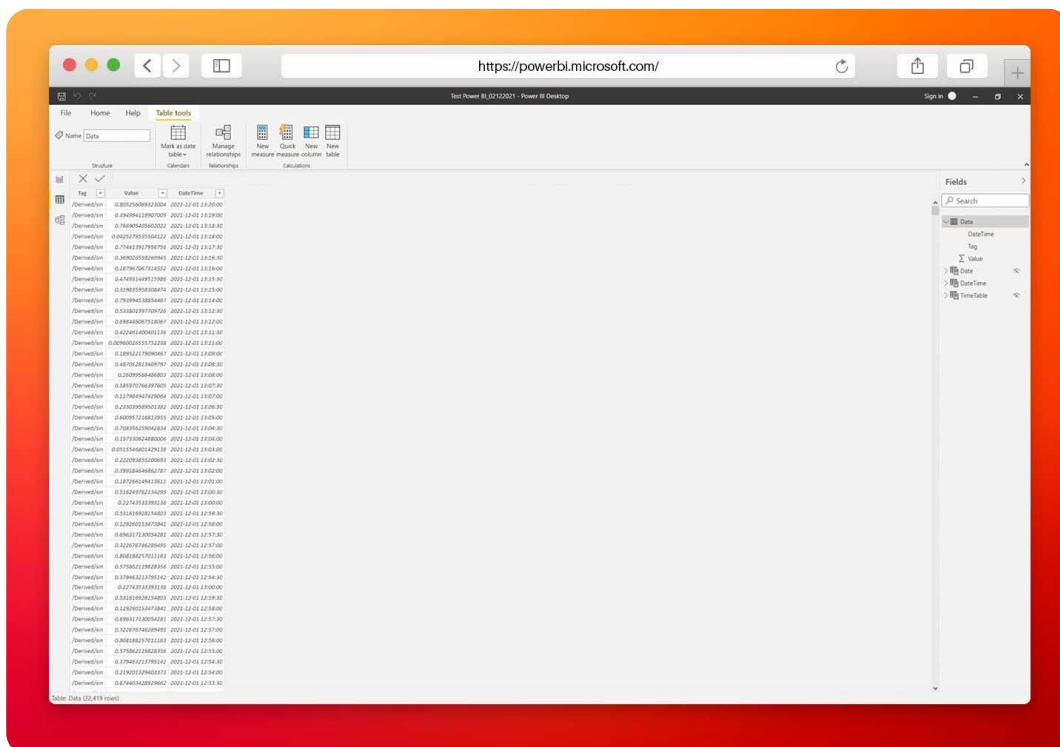
The front panel of Power BI consists of three main sections:

- **Step 01: Report view**, where visualizations and reports are built.



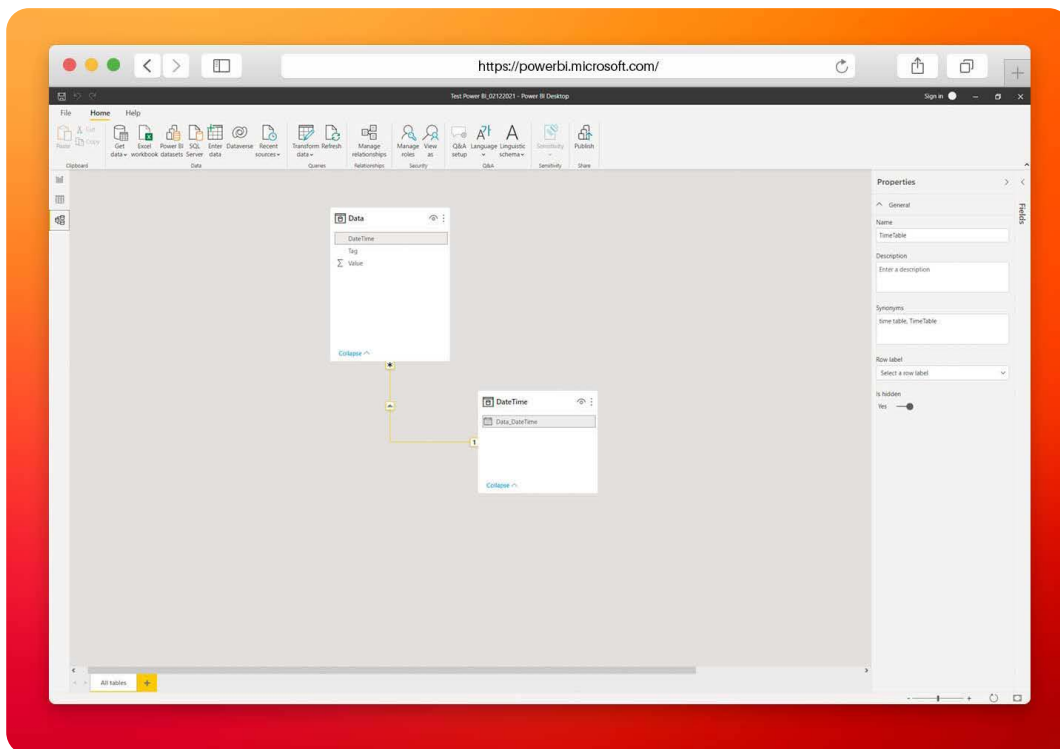
Screenshot displaying the report view within the Power BI platform

- **Step 02: Data view** helps you to inspect, explore, and understand data in your Power BI Desktop model. This is different from how you view tables, columns, and data in the Power BI Query Editor. Data view shows you all data after it has been loaded into the model.



Screenshot displaying data view in the Power BI platform

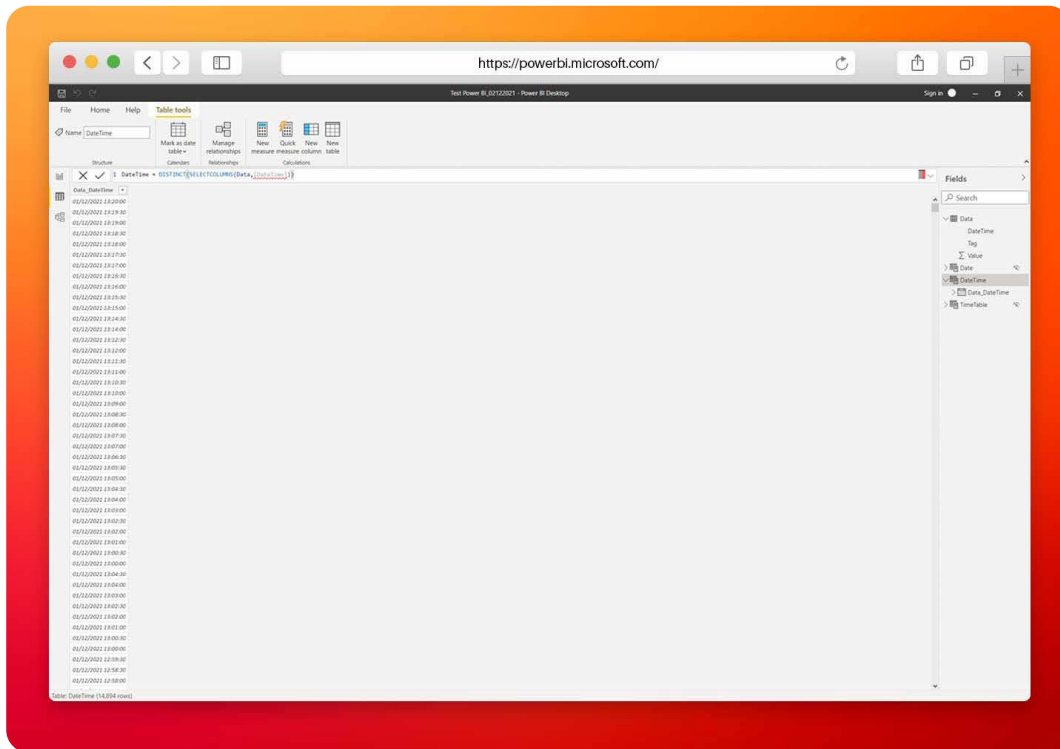
- **Step 03:** Model view is where the model is defined by establishing relations between each of the various uploaded data sets. In this case and following best practices, a very simple start model has been selected using just one dimensional table for Data/Time.



Screenshot displaying model view in the Power BI platform

The result is a **DateTime** table that we have created from our **Data** table, for which we followed the following steps:

- **Step 01:** Go to the **Data** view and from the **Table Tools** menu, click on **New table**.
- **Step 02:** In the **DAX Editor**, enter the following expression and hit enter: `DateTime = DISTINCT(SELECTCOLUMNS(Data, [DateTime]))`
- **Step 03:** Now, in the **Model** view, establish a one to many relationship connecting the DateTime type fields for both tables.

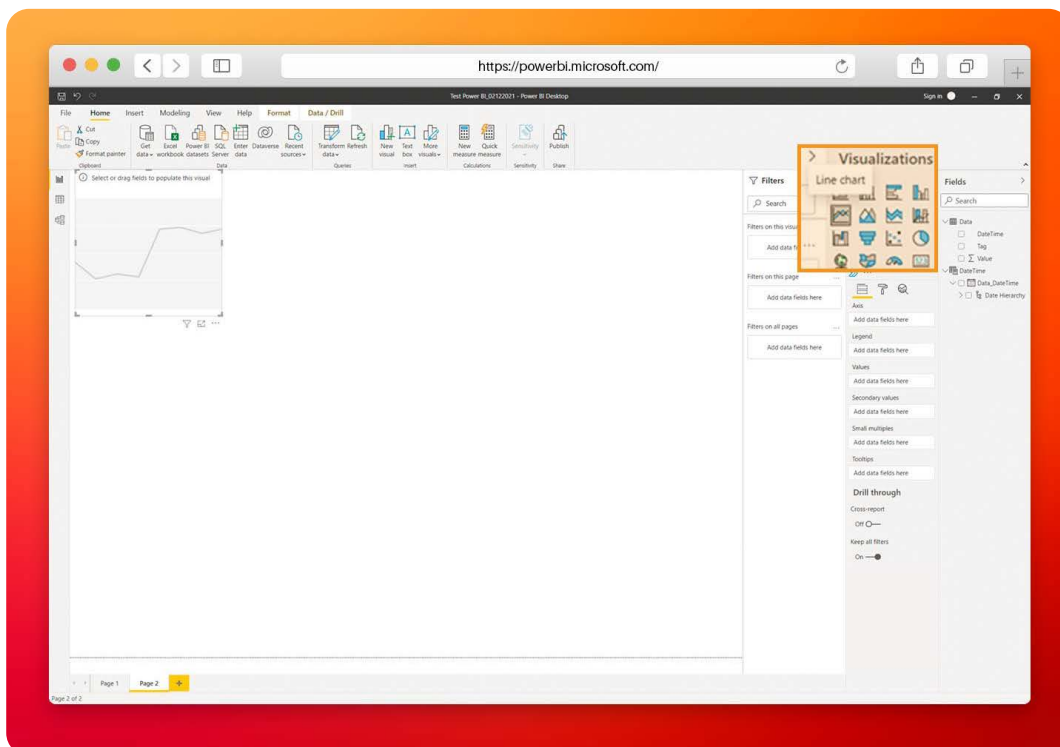


Screenshot displaying how to create a dimensional table in the Power BI platform

Building a report using the Power BI software

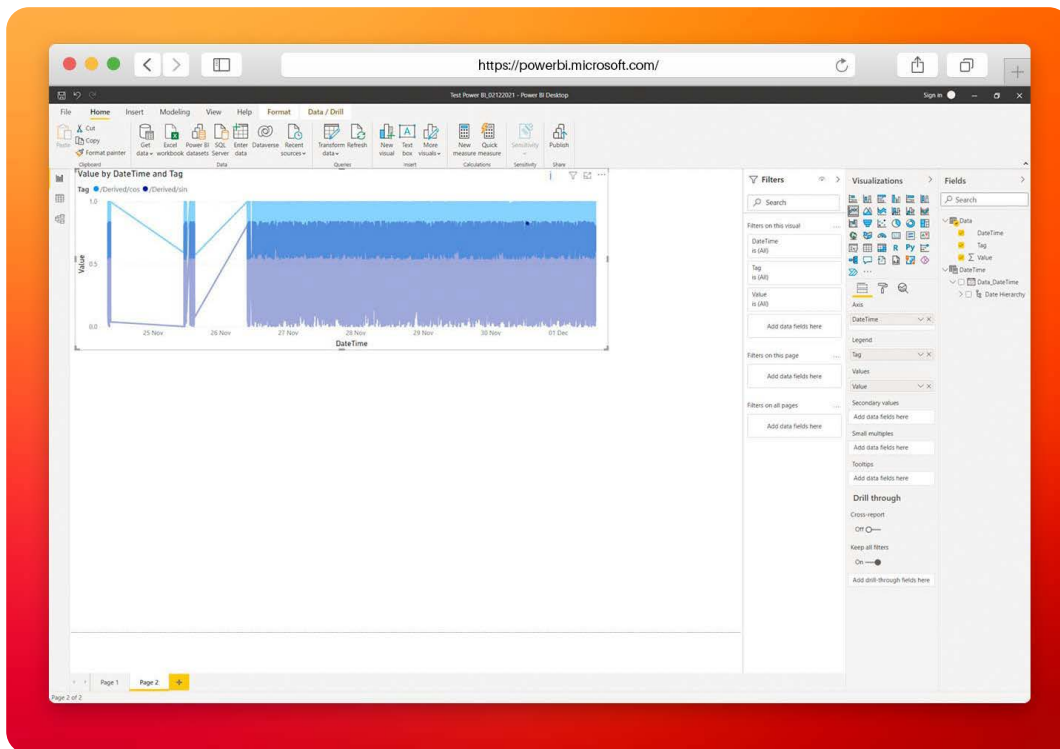
In order to build an example report, you should follow these steps:

– **Step 01:** Go to **Report view** and in **Visualizations**, select a **Line Chart**.



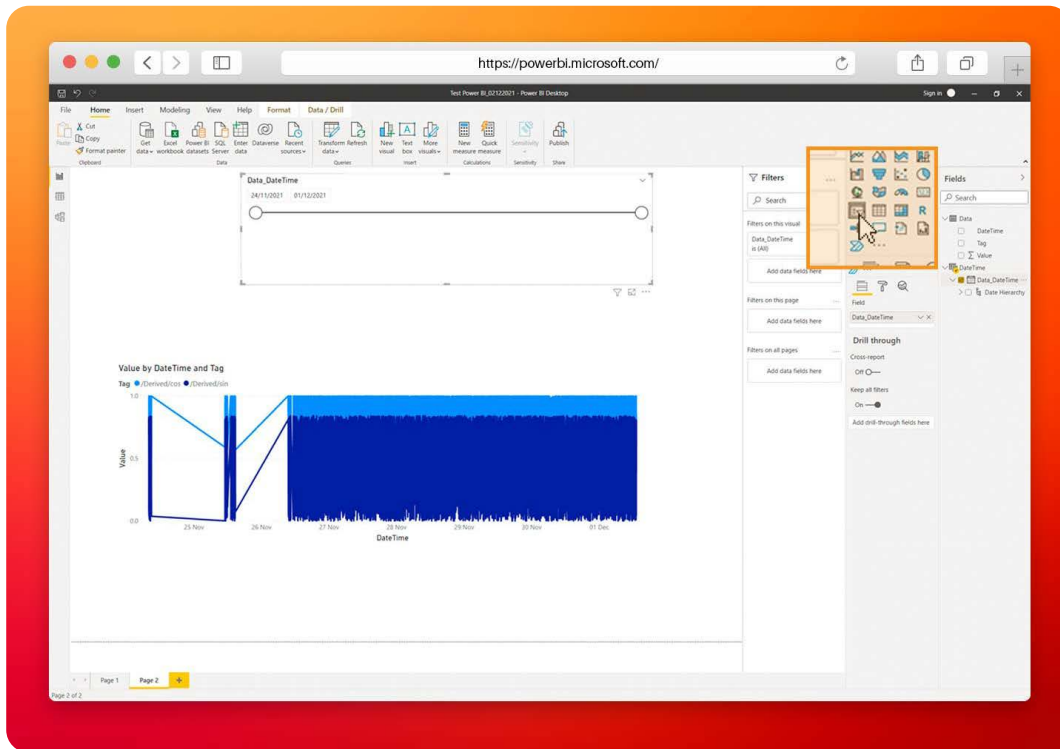
Screenshot displaying how to add a line chart within the Power BI platform

- **Step 02:** Drag and drop the **DateTime** field belonging the **Data** table from the **Field** panel located on the left-hand side, into the **Axis** field of the **Line Chart**.
- **Step 03:** Drag and drop the **Tag** field belonging to the **Data** table from the **Field** panel into the **Legend** field of the **Line Chart**.
- **Step 04:** Drag and drop the **Value** field belonging to the **Data** table from the **Field** panel into the **Values** field of the **Line Chart**.



Screenshot displaying how to configure a line chart in the Power BI platform

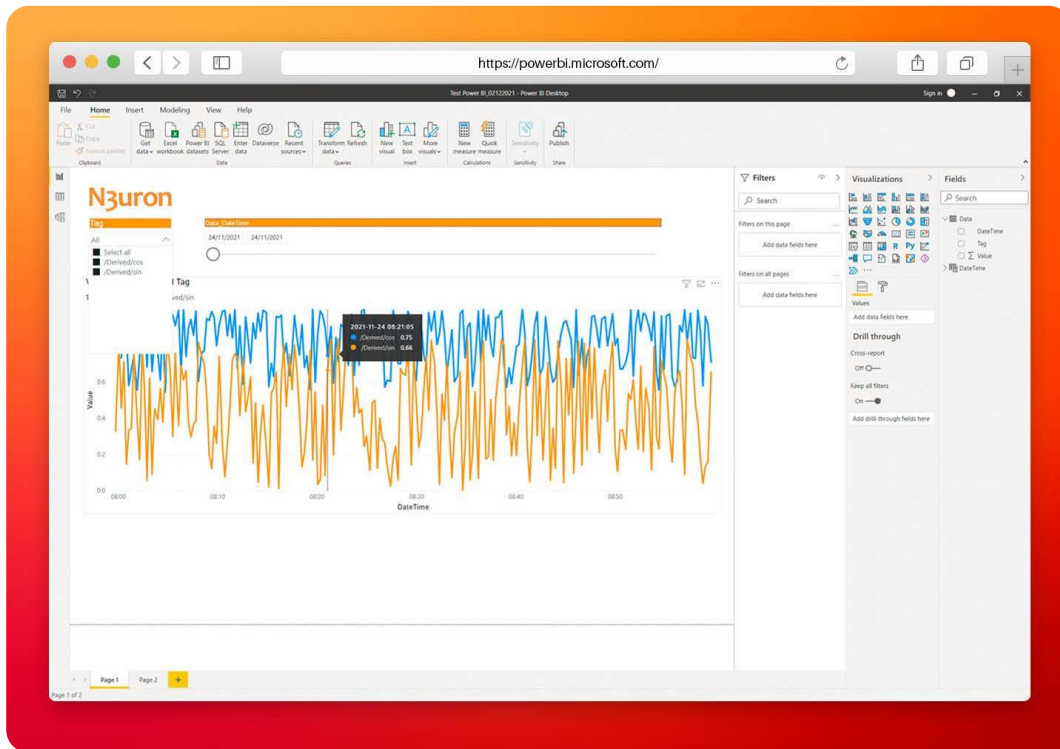
- **Step 05:** Now, let's add in a Date slicer by clicking on the **Slicer** option in **Visualizations**. Select the slicer from the canvas and click on the value that you intend to use for filtering. In this example, we have chosen the **Data_DateTime** field from our dimensional table dataset.



Screenshot displaying how to configure the slicer in the Power BI platform

- **Step 06:** Make sure the Filters section of each chart is properly configured to filter by **DateTime**.
- **Step 07:** Add another slicer, click on it, and select the **Tag** column from our **Data** table. This will allow us to select the data that we want to be displayed in the chart.
- **Step 08:** Finally, customize and rearrange all components in the correct order to achieve a nice interactive dashboard.

At this point, your report should look like something similar to the one depicted in the image below and each time you click on the **Refresh** button in the **Home** menu, your report should automatically be updated.



Screenshot displaying the resulting dashboard in the Power BI platform

Conclusion: How to squeeze industrial data using N3uron and Microsoft Power BI

Connecting your assets to Power BI is extremely easy using N3uron's Rest Api Server module. If you are ready to start using the Rest Api Server, [download N3uron's free trial version](#) and read our [Rest Api Server Manual](#) about how to implement and use N3uron's Rest Api Server module on our communication platform. Screenshot displaying expand values action in the Power BI platform.