

Installation Manual – A Custom Data Connector for Spotify and Power BI

1. Installing Power BI Desktop and create an Application on Spotify for Developers page

To be able to use this connector, you must install Power BI Desktop on your computer.

Then, you must authenticate to <https://developer.spotify.com/dashboard> and create your application, with the name you want (except for Spotify, you must not include that word to comply with their terms). This will generate you a **client_id** and a **client_secret**.

Next, on the dashboard page, under **Edit Settings**, you need to enter the following `redirect_uris`:

- <https://oauth.powerbi.com/views/oauthredirect.html>
- <https://preview.oauth.powerbi.com/views/oauthredirect.html>

2. Setting up the connector

To use the connector, the first thing to do is to copy the **SpotifyConnector.mez** file into a folder in **C:/Users/%username%/Documents/Power BI Desktop/Custom Connectors** (create it if it does not exist).

Then, you need to set up your credentials in the connector *mez* file (*mez* extension is like *zip*). Open it, and then open **Credentials.json** with a text editor. Add your information, such as **client_id**, **client_secret**, and **user_id**. You can retrieve the **client_id** and **client_secret** on the Spotify for Developers page. The **user_id** you can retrieve by opening Spotify, click on your username, and under the three dots, click **Share > Copy Spotify URI**.

2.1. Configure Power BI Settings

After that, open Power BI Desktop, and go to:

- **Files | Options, and Settings | Options**
- Select the **Security** tab
- Under **Data Extensions**, select the latter

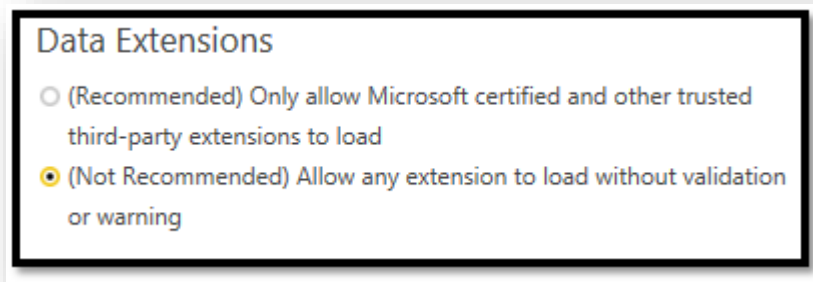


Image 1- Allow any extension to load without validation

- Then go to the **Privacy** tab
- Under **Privacy Levels**, select the latter

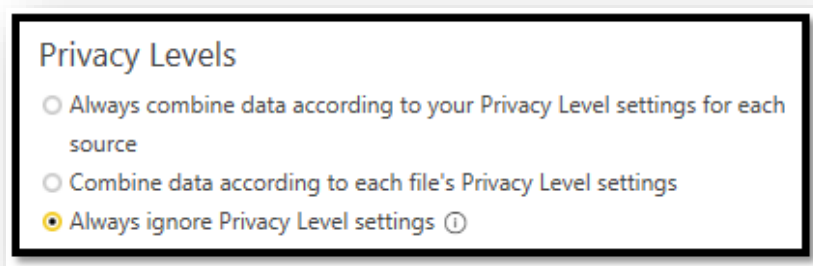


Image 2 - Always ignore Privacy Level settings

- Then go to Data Load in Current File
- Disable Background Data and Parallel loading of tables

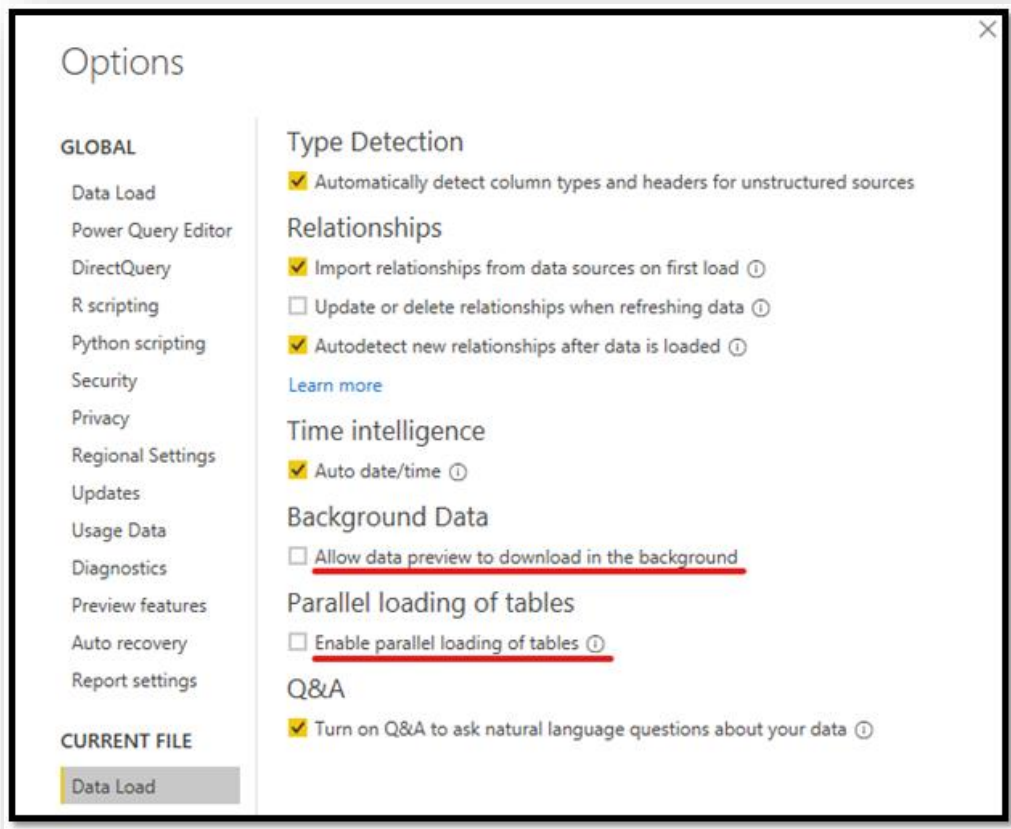


Image 3 - Background Data Settings

- Restart Power BI Desktop

3. Using the connector

To utilize the connector, select Get Data and you can search for Spotify, or access the *Other* tab, and scroll until you find it.

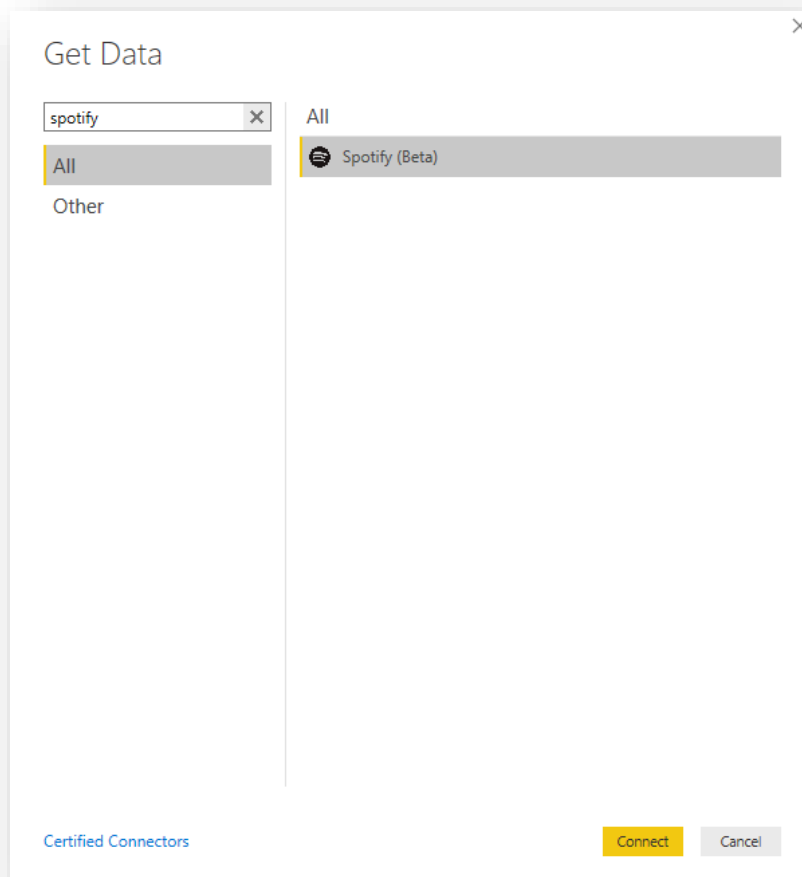


Image 4 – Get Data > Spotify Custom Connector

The connector will prompt a warning as it is still in the Beta version.

It will prompt for your credentials to log in on Spotify. Do it, and accept the terms.

Once you're connected, you'll see the following navigation table:

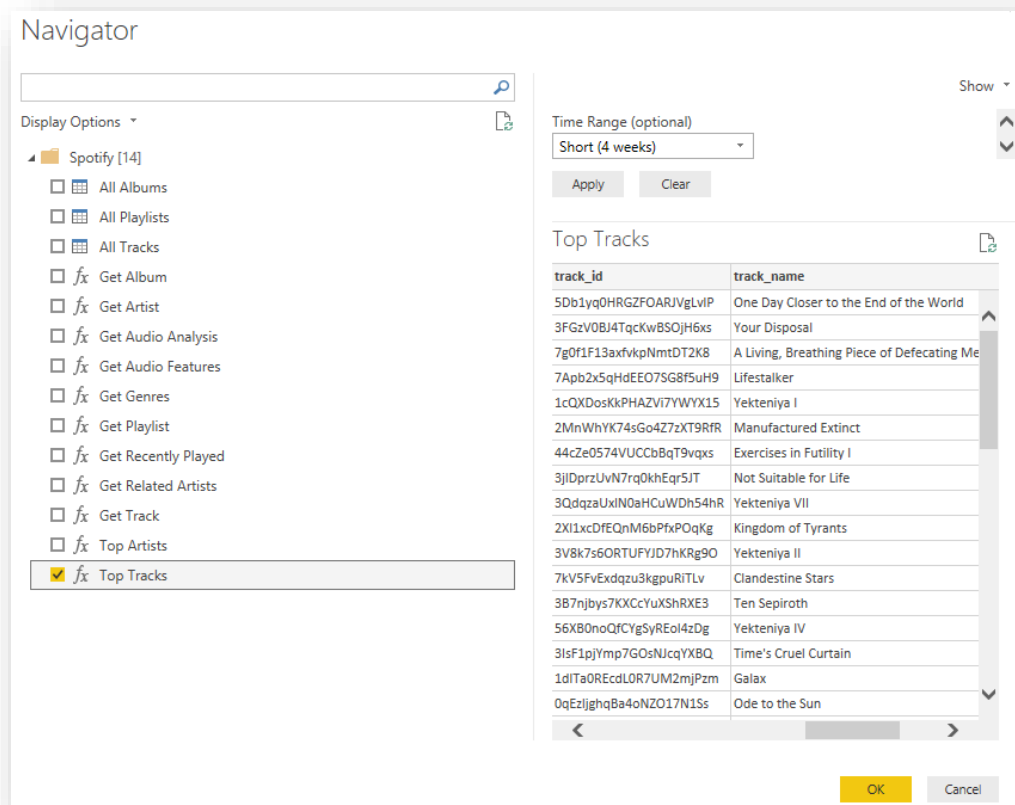


Image 5 – Connector Navigation Table

Note: for the last two queries, **Top Artists** and **Top Tracks**, you can define which time range should be considered:

- **Short (4 weeks)**
- **Medium (6 months)** – this is also the **default** if you do not want to input any value
- **Long (years or overall)**

Note: If you want to keep this simple and use the connector for simple reports and overall stats, the rest of this document is not for you. I will go through some processes to store the actual data in a SQL database and building a fact table. Also, the template provided won't work to its fullest because it has some code for the latter.

4. Building a Fact Table – Streaming History

For this to be useful in the medium/long term, you will need to store the data. You will need a database to store all your played Songs since the **GetRecentlyPlayed** function only fetches your last 50 recently played songs. The idea is to add them to the database programmatically and build a Fact Table.

I recommend that you save the data on a SQL Server database, that gets **updated** every time you hit **refresh** on the Power BI file or in the service. It can also be refreshed automatically with the Schedule Refresh feature of Power BI Service. We'll talk about it later.

4.2. Optional: Get some historic data

If you do not want to start from zero, request some historic data from your Spotify account, so you can set up your database with some data. They provide user data for the previous 2 years.

To request user streaming history data to Spotify, you must:

1. Access your Spotify Account on the browser
2. Navigate Privacy Settings
3. Scroll down and "Download your Data"

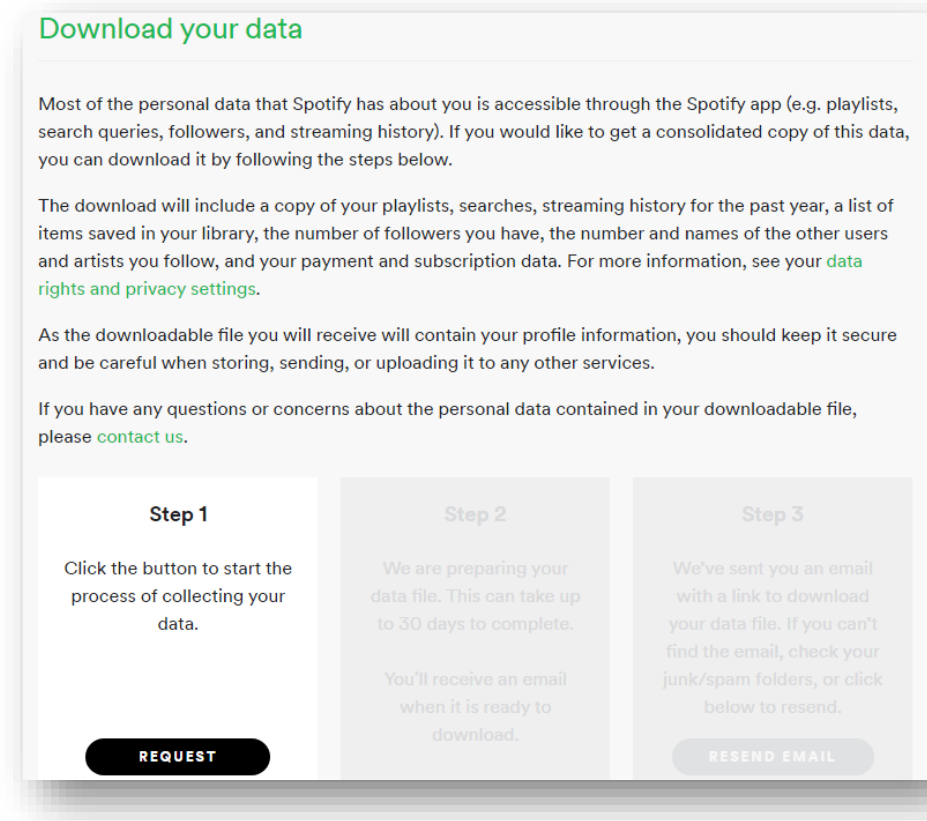


Image 6 – Download your streaming data

After a few days, you will receive an email that you can download the data.

After downloading, extract the json files to a folder that you do not intend to move around.

The most important file is *StreamingHistory.json*. The path of the folder that you save this file will be needed to enter as a parameter when you open the Template. If you do not have it yet, do not worry, type some other thing in there, and come back here when you have the historic data.

4.2.1. Opening the Template

It is time to open the template provided. You should be prompted to enter some parameters.

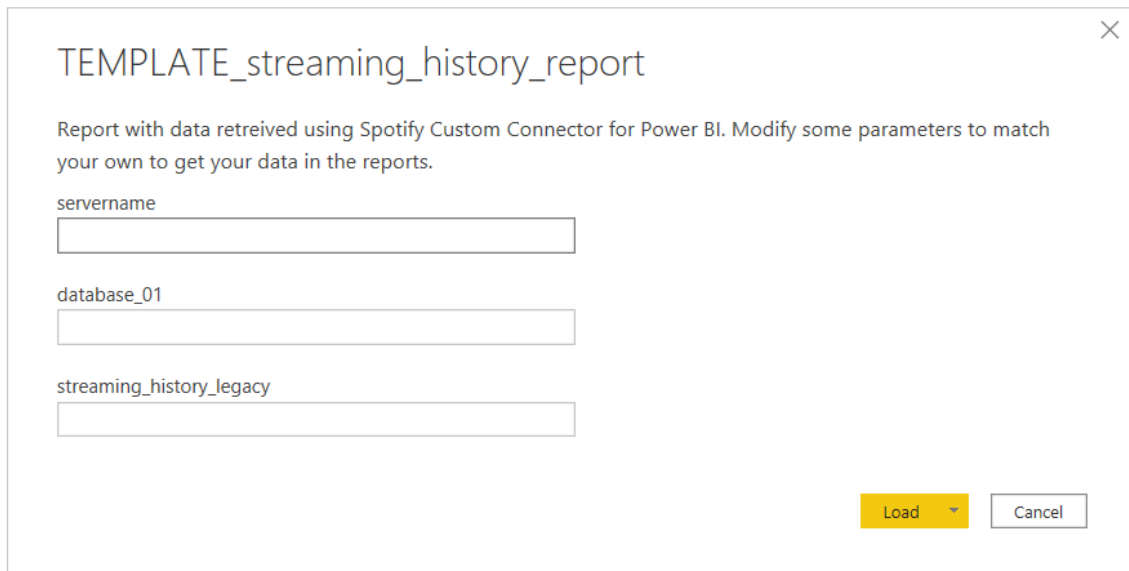


Image 7 – Template_Streaming History Report prompt

The **servername** is your SQL Server name.

You do not need to enter these correctly right away, you can just skip this by typing any characters in the input boxes and click **Load**.

If you want to create a *database* first, go to [section 5](#).

If you already have it, enter the path for your requested historic data *streaming_history_legacy.json* file.

4.2.2. Transforming the Streaming History Legacy table

Since this **data source** is in json format, it needs to be cleaned and transformed using Power Query to comply with the format of the fact table that the connector provides.

I have done the transformations in the template file, so all you need to do is:

- Open the template file and go to Power Query (Transform Data)
- Find the StreamingHistory Legacy query. It should be broken.

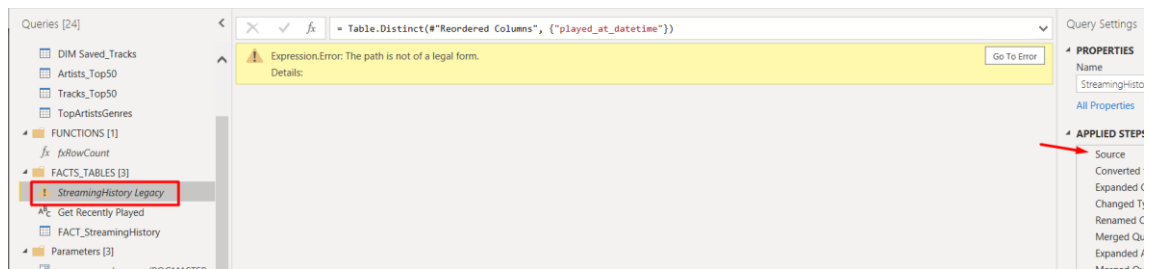


Image 8 – Broken query – StreamingHistoryLegacy

- Now to fix it, we will have to change the source file parameter. Go to Parameters, find the *streaming_history_legacy*, and change its value to **your path of the json file**.

Manage Parameters

New

servername_homepc

database_01

streaming_history_legacy

streaming_history_legacy

Name

streaming_history_legacy

Description

☒ Required

Type

Text

Suggested Values

Any value

Current Value

C:\Users\craop\Documents\KOLETOB\MVData\streaminghist

Image 9 – Parameters in Power Query

There are a few transformations done to the raw table coming from Spotify. At first, it is like this:

ABC 123	endTime	ABC 123	artistName	ABC 123	trackName	ABC 123	msPlayed
	2018-09-14 10:56		Beyond Creation		The Deported		540281

Image 10 – Transforming StreamingHistoryLegacy Table I

But after a few transformations that are already set up in Power Query, it will be in the same format as your FACT table:

played_at_datetime	artist_name	track_name	duration_ms
14/09/2018 10:56:00	Beyond Creation	The Departed	540281

Image 11 – Transforming StreamingHistoryLegacy Table II

You can check the transformations on the **template file**, on the Applied Steps panel:

= Table.Distinct("#Reordered Columns", {"played_at_datetime"})							PROPERTIES Name StreamingHistory Legacy All Properties	
track_id	artist_id	album_id	played_at_datetime	track_name	artist_name		APPLIED STEPS Source Converted to Table Expanded Column1 Changed Type Renamed Columns Merged Queries Expanded All Tracks Merged Queries1 Expanded Top Artists Merged Queries2 Expanded All Albums Merged Queries3 Expanded Top Tracks Added Conditional Column Added Conditional Column1 Added Conditional Column2 Added Conditional Column3 Removed Other Columns Changed Type1 Renamed Columns1 Reordered Columns Removed Duplicates	
1	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	14/09/2018 10:56:00	The Departed	Beyond Creation		
2	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	15/09/2018 11:30:00	The Departed	Beyond Creation		
3	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	17/09/2018 09:21:00	The Departed	Beyond Creation		
4	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	18/09/2018 08:36:00	The Departed	Beyond Creation		
5	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	18/09/2018 17:41:00	The Departed	Beyond Creation		
6	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	19/09/2018 09:22:00	The Departed	Beyond Creation		
7	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	19/09/2018 09:34:00	The Departed	Beyond Creation		
8	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	19/09/2018 13:04:00	The Departed	Beyond Creation		
9	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	19/09/2018 20:01:00	The Departed	Beyond Creation		
10	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	20/09/2018 21:12:00	The Departed	Beyond Creation		
11	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	21/09/2018 09:54:00	The Departed	Beyond Creation		
12	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	22/09/2018 20:18:00	The Departed	Beyond Creation		
13	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	22/09/2018 23:04:00	The Departed	Beyond Creation		
14	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	23/09/2018 13:23:00	The Departed	Beyond Creation		
15	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	23/09/2018 14:55:00	The Departed	Beyond Creation		
16	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	24/09/2018 20:13:00	The Departed	Beyond Creation		
17	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	25/09/2018 10:41:00	The Departed	Beyond Creation		
18	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	25/09/2018 10:54:00	The Departed	Beyond Creation		
19	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	26/09/2018 11:11:00	The Departed	Beyond Creation		
20	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	27/09/2018 13:50:00	The Departed	Beyond Creation		
21	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	28/09/2018 09:25:00	The Departed	Beyond Creation		
22	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	30/09/2018 19:27:00	The Departed	Beyond Creation		
23	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	01/10/2018 10:58:00	The Departed	Beyond Creation		
24	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	03/10/2018 15:11:00	The Departed	Beyond Creation		
25	2xxdTkfABqDa9yNyUR5b2	3HBOxN9Iag9Ot1kThpPX8	OZET76uVmbB8RfiktU4w99	04/10/2018 13:50:00	The Departed	Beyond Creation		

Image 12 – Transforming StreamingHistoryLegacy Table III

Now it is time to import it to a database.

There are several ways to import data into SQL Server. I did it using DAX studio so I could import the table exactly as it was on the Power BI template.

5. Save data to a SQL database

5.1. Create the database in SQL Server

Create a database for yourself on SQL Management Studio. I recommend the SQL Server 2019 Express because it is free.

5.2. Import the Historic Data – Streaming History Legacy

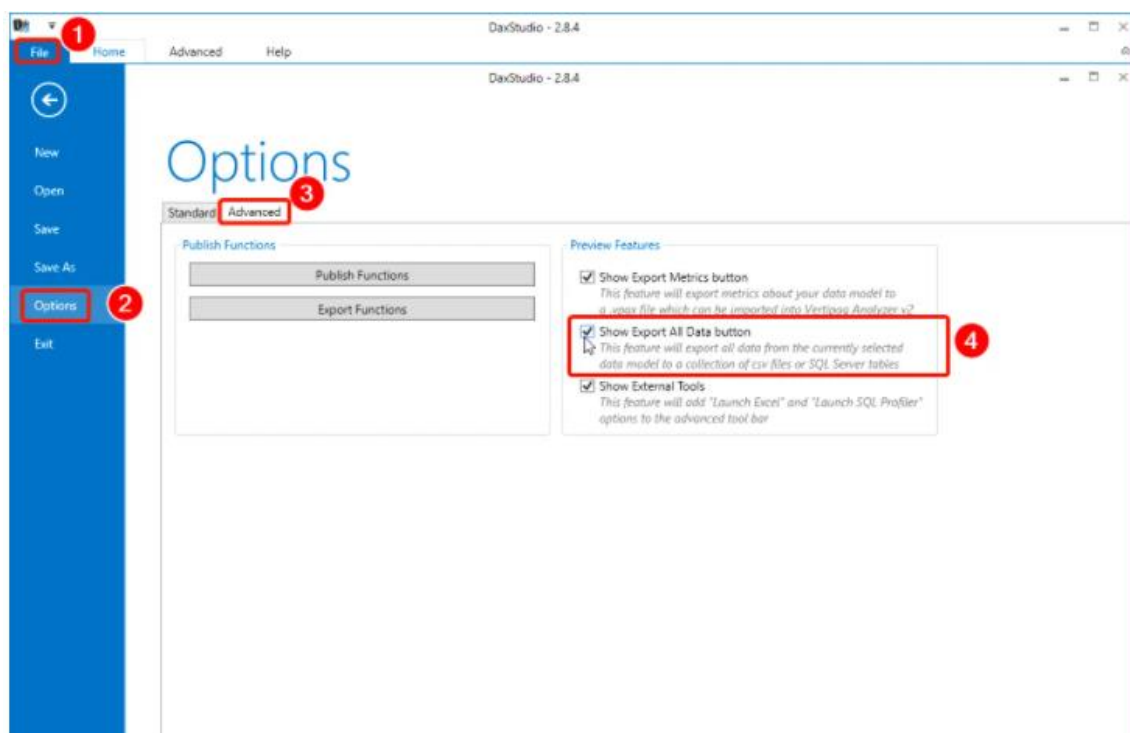
After that, you will want to import the Streaming History Legacy that we transformed earlier.

To do it this way, you'll need to download DAX Studio:

<https://daxstudio.org/>

After downloading it, open it up and do the following changes in settings:

- Go to “File” and click “Options”
- Click “Advanced” and tick the “Show Export All Data button”




The next step is to connect to the template using DAX Studio. The PBI file must be open.

Connect

Data Source

☐ PowerPivot Model [?](#)

☒ PBI / SSDT Model

 pbix_template_2

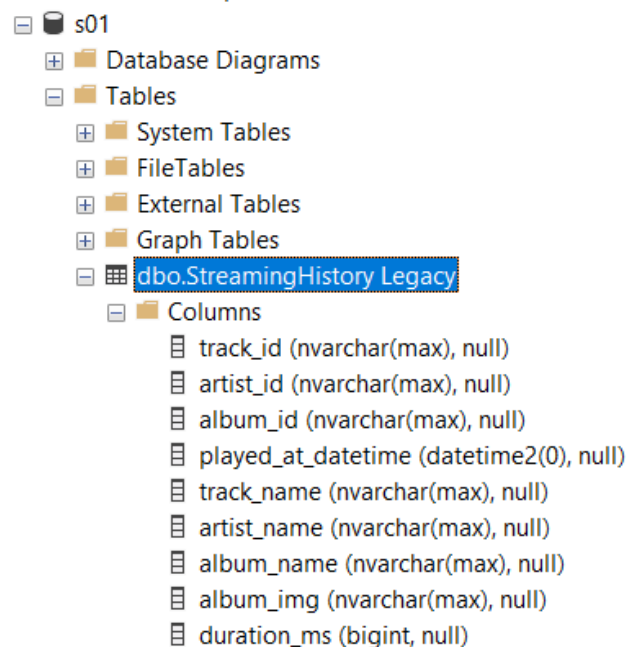
☐ Tabular Server

☒ Advanced Options

Then, go to Advanced and then Export Data. Choose SQL and enter your credentials.

The table you are looking to import is *StreamingHistory Legacy*. The table that we prepared earlier with historic data.

You should now have a table in your SQL Server that looks like this:



To comply with the table name coded on the template, my advice is to change the table name to **"AllData"**.

You can query it and see that it contains your historic data. Cool!

Now we need to set it up for inserting the new data that you will be adding every day!

5.3. Write data to SQL Server

Here I explain a bit about this process. It is based on this blogpost by Parker Stevens from BI Elite:

<https://bielite.com/blog/write-back-to-sql-database-from-power-bi/>

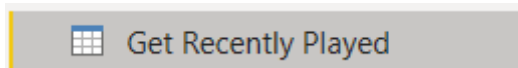
Feel free to check it out so you can understand it better. It also has a video.

Note: This is already implemented in the template file provided – **TEMPLATE_StreamingHistoryReport.pbix**.

The Process

To write data to SQL Server, you will have to do the following:

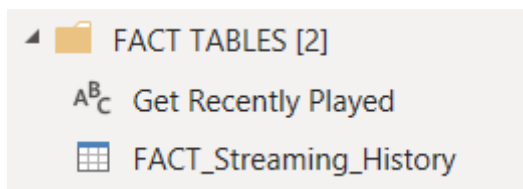
1. Import the **Get Recently Played** table using the connector



2. Add a new column with a code that programmatically will write the rows to your SQL database table if it does not exist already.
3. Add a new query that will SELECT * FROM your table

To implement this process, the first step is to add the code in the template I provided as a replacement for your **Get Recently Played** table.

You will see two separate queries:



The first one is your Get Recently Played query with an extra column that will write to the database.

The second one is the actual table that you want to use as the FACT table.

That is it! Every time you refresh **FACT_Streaming_History**, the data will be written to your database table.

The good news is that it works on the service! You can schedule the refresh on your on-premises data gateway and not worry about going into your report and hit the refresh button.

6. Schedule Refresh in the service

Schedule Refresh of your report (and database) is possible to do using an on-premises data gateway that you can download here:

<https://www.microsoft.com/en-us/download/details.aspx?id=53127>

You need to set up the gateway with your account, and then configure your dataset in the service to use this gateway.

When it is all configured, you can set it up to automatically refresh when you want. As long as the server is online, it will refresh and write the data to your database!

-work in development-