How to connect Power BI to the Spotify API Tutorial

By

Jonathan van der Waard

https://www.linkedin.com/in/jonathan-van-der-waard/

Table of Content

Minute 10

```
let
  Source = Json.Document(

Web.Contents("https://api.spotify.com/v1/search?q=artist%3AHans%20Zimmer&type=artist&limit=
1",
      [Headers=
                  [Authorization=#"Authorization Bearer",
          Accept="application/json",
                    #"Content-Type"="application/json"
                   ]
            ]
           )
  )
in
  Source
```

Minute 18

```
(Artistname as text) =>
let
  Source = Json.Document(
     Web.Contents("https://api.spotify.com/v1/search?q=artist%3A" & Artistname &
"&type=artist&limit=1",
       [Headers=
                    [Authorization=#"Authorization Bearer",
        Accept="application/json",
                       #"Content-Type"="application/json"
                     ]
            ]
          )
  ),
    #"Converted to Table" = Record.ToTable(Source),
    Value = #"Converted to Table"{0}[Value],
    items = Value[items],
    items1 = items{0},
    #"Converted to Table1" = Record.ToTable(items1),
    #"Transposed Table" = Table.Transpose(#"Converted to Table1"),
    #"Promoted Headers" = Table.PromoteHeaders(#"Transposed Table", [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"external_urls", type
any}, {"followers", type any}, {"genres", type any}, {"href", type text}, {"id", type text}, {"images",
type any}, {"name", type text}, {"popularity", Int64.Type}, {"type", type text}, {"uri", type text}}),
    #"Expanded external_urls" = Table.ExpandRecordColumn(#"Changed Type", "external_urls",
{"spotify"}, {"external_urls.spotify"}),
    #"Expanded followers" = Table.ExpandRecordColumn(#"Expanded external_urls", "followers",
{"total"}, {"followers.total"}),
    #"Expanded genres" = Table.ExpandListColumn(#"Expanded followers", "genres"),
    #"Expanded images" = Table.ExpandListColumn(#"Expanded genres", "images"),
    #"Expanded images1" = Table.ExpandRecordColumn(#"Expanded images", "images", {"height",
"url", "width"}, {"images.height", "images.url", "images.width"})
in
  #"Expanded images1"
```

Minute 25

```
(ArtistName as text) =>                                          // This function takes a text input,
ie. Hans Zimmer
let
   BaseURL = "https://api.spotify.com/v1/search",                // This is the Base URL
of the API
   DealWithSpace = Text.Replace(ArtistName," ", "%20"),          // An API can't deal
with spaces, so we'll replace those with %20.
   Request = Json.Document(
      Web.Contents(
         BaseURL,
         [
            RelativePath = "?q=" & ArtistName & "&type=artist&limit=1",     // This is where
we use our input value
            Headers = [
               Accept = "application/json",
               #"Content-Type" = "application/json",
               Authorization = #"Authorization Bearer"                    // This is the Spotify
Authorization Bearer. It is only valid for a limited time. Refresh it before running.
            ]

         ]
      )
   ),
   Response = Request[artists],
   GetTheList = Response[items],
   ConvertToTable = Table.FromList(GetTheList, Splitter.SplitByNothing(), null, null,
ExtraValues.Error),
   Expand = Table.ExpandRecordColumn(ConvertToTable, "Column1", {"followers", "genres", "id",
"images", "name", "popularity"}, {"Followers", "Genres", "ArtistID", "ArtistImages", "Artist Name",
"Popularity"})
                                                                   // Here we only select the columns that we're
interested in and already do some proper renaming
in
   Expand
```

Minute 36

```
(ArtistID as text) =>                                   // This function takes a text input, ie. 0YC192cP3KPCRWx8zr8MfZ (for Hans Zimmer)
let
   BaseURL = "https://api.spotify.com/v1/artists/",                  // This is the Base URL of the API
   Request = Json.Document(
      Web.Contents(
         BaseURL,
         [
            RelativePath = ArtistID & "/albums?offset=0&limit=50",          // This is where we use our input value
            Headers = [
               Accept = "application/json",
               #"Content-Type" = "application/json",
               Authorization = #"Authorization Bearer"                  // This is the Spotify Authorization Bearer. It is only valid for a limited time. Refresh it before running.
            ]

         ]
      )
   ),
   Response = Request[items],
   ConvertToTable = Table.FromList(Response, Splitter.SplitByNothing(), null, null, ExtraValues.Error),
   Expand = Table.ExpandRecordColumn(ConvertToTable, "Column1", {"album_group", "album_type", "artists", "available_markets", "id", "images", "name", "release_date", "release_date_precision", "total_tracks"}, {"Album Group", "Album Type", "Album Artists", "Available Markets", "Album ID", "Album Images", "Album Name", "Release Date", "Release Date Precision", "Total Tracks"})
    // In the above line we only select the columns that we're interested in and already do some proper renaming
in
   Expand
```

Minute 48

```
(ArtistID as text) =>
                                    // This function takes a text input, ie. 0YC192cP3KPCRWx8zr8MfZ (for Hans
Zimmer)
let
   BaseURL = "https://api.spotify.com/",
                                // This is the Base URL of the API
   Request = Json.Document(
      Web.Contents(
         BaseURL,
         [
            RelativePath = "v1/artists/" & ArtistID & "/albums?offset=0&limit=50",                    //
This is where we use our input value
            Headers = [
              Accept = "application/json",
              #"Content-Type" = "application/json",
              Authorization = #"Authorization Bearer"                                  // This is the Spotify
Authorization Bearer. It is only valid for a limited time. Refresh it before running.
            ]

         ]
      )
   ),
   TotalAlbums = Request[total],                                                  // Instead of picking the list, let's
pick up the total
   NumberofOffsets = Number.Round(TotalAlbums/50) - 1,                                  // Divide it by 50
   Record = Record.FromList({NumberofOffsets}, {"Nr"}),
   CreateList = List.Generate(() => NumberofOffsets, each _ > -1, each _ - 1),          // Create a list
starting from zero with the Number of Offsets we got two rows above
   ConvertListToTable = Table.FromList(CreateList, Splitter.SplitByNothing(), null, null,
ExtraValues.Error),
   MultiplyBy50 = Table.TransformColumns(ConvertListToTable, {{"Column1", each _ * 50, type
number}})  // Multiply by 50 to get to the right offsets needed for the API

in
   MultiplyBy50
```

Minute 52

```
(ArtistID as text, Offset as text) =>                                   // This function takes a text input, ie. 0YC192cP3KPCRWx8zr8MfZ (for Hans Zimmer)
let
   BaseURL = "https://api.spotify.com/v1/artists/",                     // This is the Base URL of the API
   Request = Json.Document(
      Web.Contents(
         BaseURL,
         [
            RelativePath = ArtistID & "/albums?offset=" & Offset & "&limit=50",          // This is where we use our input value
            Headers = [
               Accept = "application/json",
               #"Content-Type" = "application/json",
               Authorization = #"Authorization Bearer"                  // This is the Spotify Authorization Bearer. It is only valid for a limited time. Refresh it before running.
            ]

         ]
      )
   ),
   Response = Request[items],
   ConvertToTable = Table.FromList(Response, Splitter.SplitByNothing(), null, null, ExtraValues.Error),
   Expand = Table.ExpandRecordColumn(ConvertToTable, "Column1", {"album_group", "album_type", "artists", "available_markets", "id", "images", "name", "release_date", "release_date_precision", "total_tracks"}, {"Album Group", "Album Type", "Album Artists", "Available Markets", "Album ID", "Album Images", "Album Name", "Release Date", "Release Date Precision", "Total Tracks"})
    // In the above line we only select the columns that we're interested in and already do some proper renaming
in
   Expand
```

1 Hour 4 Minute x: Top Tracks

```
(ArtistID as text, Offset as text) =>                                   // This function takes a text input, ie. 0YC192cP3KPCRWx8zr8MfZ (for Hans Zimmer)
let
   BaseURL = "https://api.spotify.com/v1/artists/",                     // This is the Base URL of the API
   Request = Json.Document(
      Web.Contents(
         BaseURL,
         [
            RelativePath = ArtistID & "/albums?offset=" & Offset & "&limit=50",          // This is where we use our input value
            Headers = [
               Accept = "application/json",
               #"Content-Type" = "application/json",
               Authorization = #"Authorization Bearer"                  // This is the Spotify Authorization Bearer. It is only valid for a limited time. Refresh it before running.
            ]

         ]
      )
   ),
   Response = Request[items],
   ConvertToTable = Table.FromList(Response, Splitter.SplitByNothing(), null, null, ExtraValues.Error),
   Expand = Table.ExpandRecordColumn(ConvertToTable, "Column1", {"album_group", "album_type", "artists", "available_markets", "id", "images", "name", "release_date", "release_date_precision", "total_tracks"}, {"Album Group", "Album Type", "Album Artists", "Available Markets", "Album ID", "Album Images", "Album Name", "Release Date", "Release Date Precision", "Total Tracks"})
    // In the above line we only select the columns that we're interested in and already do some proper renaming
in
   Expand
```

1 Hour 4 Minute x: Tracks Offset

```
(AlbumID as text) =>
let
   BaseURL = "https://api.spotify.com/v1/",
   Request = Json.Document(
      Web.Contents(
         BaseURL,
         [
            RelativePath = "albums/" & AlbumID & "/tracks?offset=0&limit=50",
            Headers = [
              Accept = "application/json",
              #"Content-Type" = "application/json",
              Authorization = #"Authorization Bearer"
            ]

         ]
      )
   ),

   TotalTracks = Request[total] ,
   NumberofOffsets = Number.Round(TotalTracks/50),
   Step2 = if NumberofOffsets > 0 then NumberofOffsets - 1 else NumberofOffsets ,
   Record = Record.FromList({Step2}, {"Nr"}),
   CreateList = List.Generate(() => NumberofOffsets, each _ > -1, each _ - 1),
   ConvertListToTable = Table.FromList(CreateList, Splitter.SplitByNothing(), null, null,
ExtraValues.Error),
   MultiplyBy50 = Table.TransformColumns(ConvertListToTable, {{"Column1", each _ * 50, type
number}})
in
   MultiplyBy50
```

1 Hour 4 Minute x: Tracks

```
(AlbumID as text, MyOffset as text) =>
let
   BaseURL = "https://api.spotify.com/v1/",
   Request = Json.Document(
      Web.Contents(
         BaseURL,
         [
            RelativePath = "albums/" & AlbumID & "/tracks?offset=0&limit=50",
            Headers = [
               Accept = "application/json",
               #"Content-Type" = "application/json",
               Authorization = #"Authorization Bearer"
            ]

         ]
      )
   ),
   items = Request[items],
   #"Converted to Table" = Table.FromList(items, Splitter.SplitByNothing(), null, null,
ExtraValues.Error),
   #"Expanded Column1" = Table.ExpandRecordColumn(#"Converted to Table", "Column1",
{"available_markets", "disc_number", "duration_ms", "explicit", "id", "is_local", "name",
"track_number"}, {"Track Available Markets", "Track Disc Number", "Track Duration in ms", "Track
Explicit", "Track ID", "Track Is Local", "Track Name", "Track Number"})
in #"Expanded Column1"
```

1 Hour 4 Minute x: Tracks Features

```
(TrackID as text) =>
let
   BaseURL = "https://api.spotify.com/v1/",
   Request = try Json.Document(
      Web.Contents(
         BaseURL,
         [
            RelativePath = "audio-features/" & TrackID,
            Headers = [
               Accept = "application/json",
               #"Content-Type" = "application/json",
               Authorization = #"Authorization Bearer"
            ]

         ]
      )
   ) otherwise null,
   #"Converted to Table" = try Record.ToTable(Request) otherwise null,
   #"Pivoted Column" = try Table.Pivot(#"Converted to Table", List.Distinct(#"Converted to
Table"[Name]), "Name", "Value")  otherwise null,
   #"Removed Other Columns" = try Table.SelectColumns(#"Pivoted Column",{"danceability",
"energy", "key", "loudness", "mode", "speechiness", "acousticness", "instrumentalness", "liveness",
"valence", "tempo", "type", "id", "duration_ms", "time_signature"}) otherwise null
in
   #"Removed Other Columns"
```