

## Spotify Radar - ES (PDF)

[Spotify Developer](#)

[IDs de artistas](#)

[Telegram Bot](#)

[Make.com](#)

[Nodo 1 - Variable](#)

[Nodo 2 - Iterator \(1\)](#)

[Nodo 3 - HTTP Request](#)


[Nodo 4 - Iterator \(2\)](#)

[Nodo 5 - Mensaje de Telegram](#)

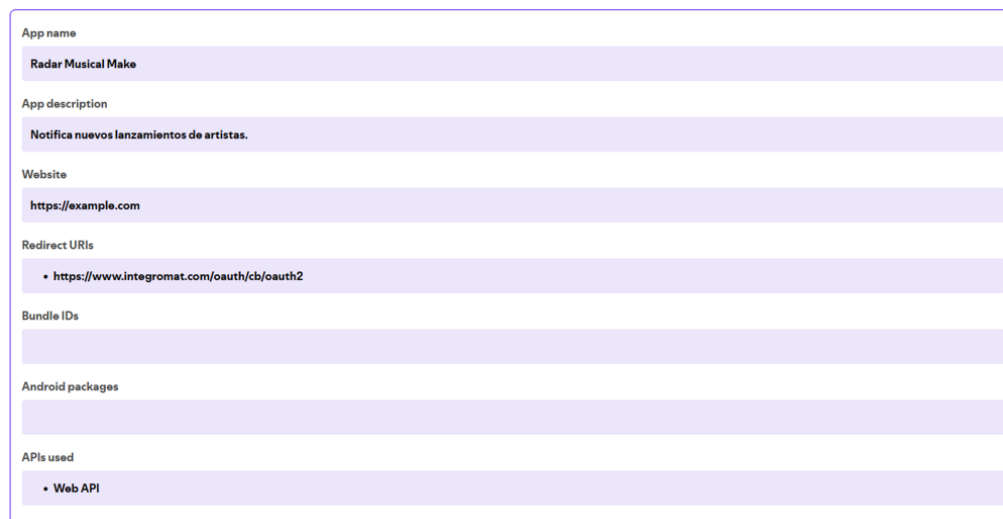
---

## Spotify Developer

✓ Primero, necesitamos una cuenta de Spotify para desarrolladores.

 [Home](#) | [Spotify for Developers](#)

Una vez hecha la cuenta, vamos al [Dashboard](#) y creamos una APP. Aquí debemos poner las siguientes opciones ( **Website** no importa):



The screenshot shows a form for creating a new app on the Spotify Developer Dashboard. The form has several fields, each with a label and a text input area. The fields are: App name (filled with 'Radar Musical Make'), App description (filled with 'Notifica nuevos lanzamientos de artistas.'), Website (filled with 'https://example.com'), Redirect URIs (filled with 'https://www.integromat.com/oauth/cb/oauth2'), Bundle IDs (empty), Android packages (empty), and APIs used (filled with 'Web API').

App name	Radar Musical Make
App description	Notifica nuevos lanzamientos de artistas.
Website	https://example.com
Redirect URIs	• https://www.integromat.com/oauth/cb/oauth2
Bundle IDs	
Android packages	
APIs used	• Web API

**App name** → Cualquier nombre que quieras

**App description** → La descripción que quieras

**Website** → <https://example.com> (esto es de mentira, no importa lo que pongamos pero no puede estar vacío)

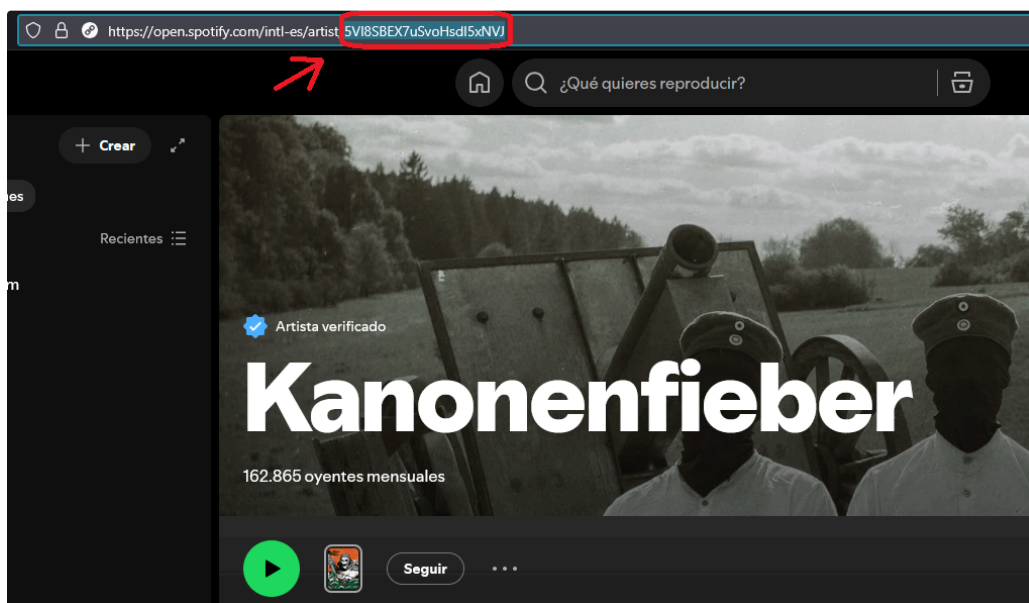
**Redirect URIs** → `https://www.integromat.com/oauth/cb/oauth2`

**APIs used** → Web API

Encima de esta información también veremos el **Client ID** y la **Client Secret**.  
Importante para más tarde , así que hay que anotarlo. Por esta parte ya hemos acabado.

## IDs de artistas

Antes de ir a [Make.com](https://make.com), hay que ver los IDs de los artistas que nos interesan en Spotify. Para ello, hay que entrar en la página cada artista en Spotify y en la URL, después de `/artist/` estará el ID.



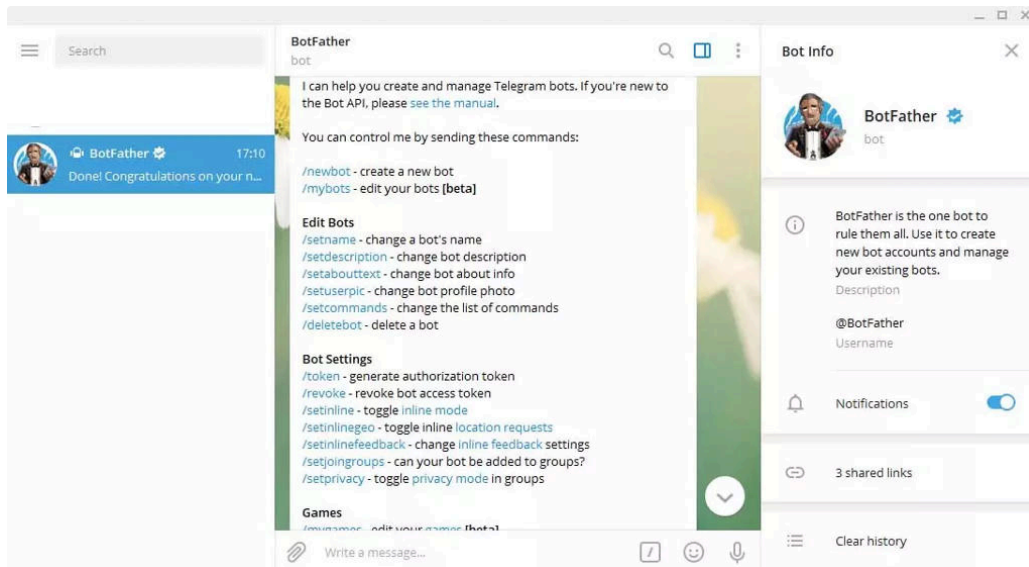
Estos IDs hay que copiarlos y hacer una lista en este formato (cada ID debe pegarse sin los corchetes):

```
1 [Artist_ID1],[Artist_ID2],[Artist_ID3]
2 --- Ejemplo ---
3 5VI8SBEX7uSvoHsdI5xNVJ,05fG473iIaoy82BF1aGhL8,240iw7B1v01BETecDLJt6m
```

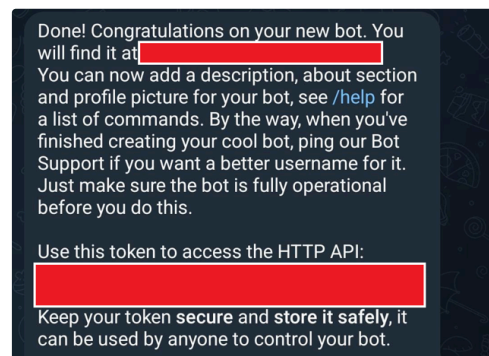
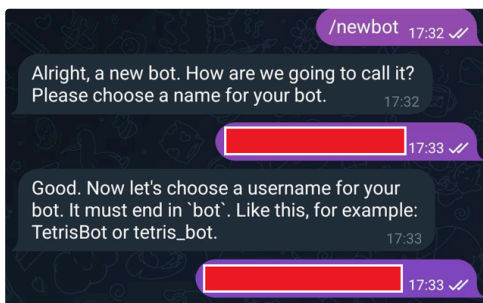
---

## Telegram Bot

Para crear un bot en Telegram, iniciaremos un chat con **@BotFather**. Se puede buscar con ese nombre exacto.



- Pulsamos en **Start** para iniciar la conversación.
- Como podemos comprobar, el comando para crear un nuevo bot es **/newbot**.
- Por lo tanto, usaremos dicho comando dándole un **nombre y un nombre de usuario**.
- Automáticamente Telegram nos dará un token o código de autorización. Este código es necesario para que pueda usar la API Bot de la app de mensajería. Por otro lado, este Token tendremos que guardarlo a buen recaudo, ya que si otro usuario accediera a él podría controlar ese bot.



Ahora necesitaremos el Chat ID del bot. Para ello, debemos hablarle al bot que hemos creado. Y después, entrar en un navegador a la siguiente página reemplazando **[TOKEN]** por el token que recién se ha generado:

```
1 https://api.telegram.org/bot[TOKEN]/getUpdates
```

Copiaremos el número que hay después de **id:** y lo guardaremos junto al token.

## Make.com

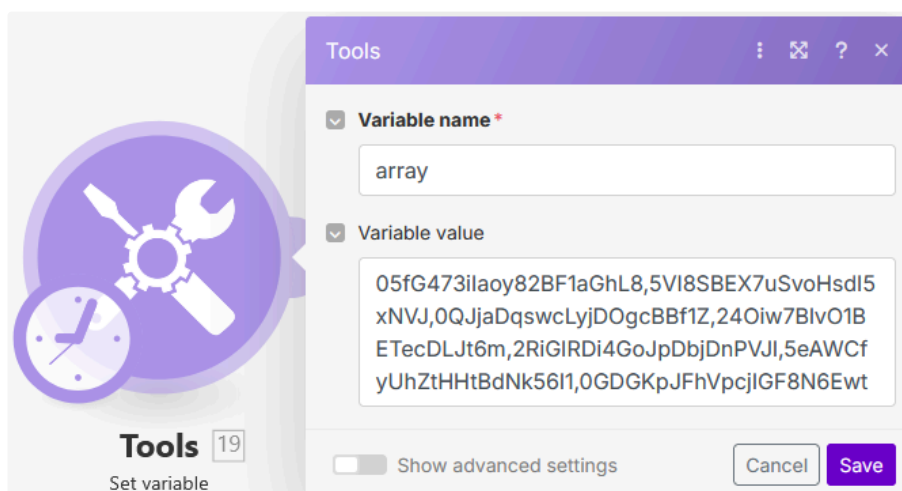
Es la hora de armar la automatización. Este es el organigrama final que nos tendría que quedar:



### Nodo 1 - Variable

Tools > Set variable

Aquí debemos darle el nombre que queramos, en este caso array. El valor de la variable debe ser la lista que hemos hecho anteriormente en el apartado **IDs de Artistas**.



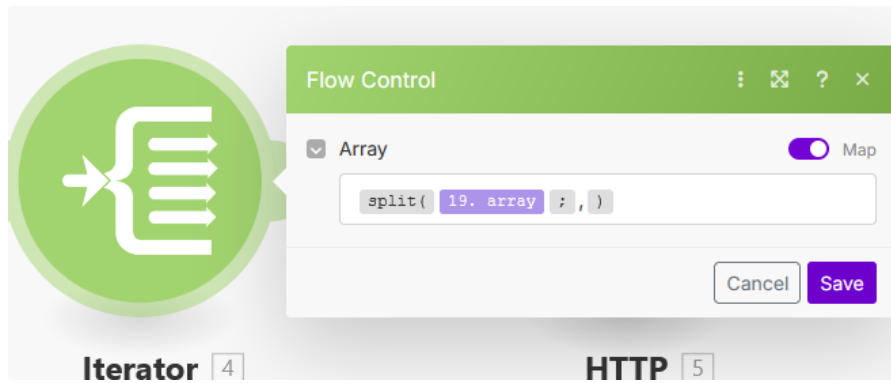
### Nodo 2 - Iterator (1)

Flow Control > Iterator

Aquí debemos buscar la función `split`. Despúes buscamos la variable que sale del apartado anterior (en mi caso `19.array`) y ponerla después de la apertura de paréntesis `(`. También hay que añadir el separador del array, que en este caso es una coma. va justo antes del cierre de paréntesis `)`.

```
1 {{split([VARIABLE]; ",")}}
```

⚠ Importante que la opción **Map** esté activada.





### Nodo 3 - HHTTP Request

HTTP > Make an OAuth 2.0 request

Esta parte es un poco más compleja. Primero debemos añadir la conexión a Spotify de la siguiente manera:

☒ **Connection \***

 Spotify Connection Add

 For more information on how to create a connection to HTTP, see the [online Help](#).

☒ **Connection name \***

My HTTP OAuth 2.0 connection

☒ **Flow type \***

Authorization Code

☒ **Authorize URI \***

☒ **Token URI \***

☒ **Scope**

[+ Add item](#)

☒ **Client ID \***

☒ **Client Secret \***

- **Connection name** → Ponle un nombre cualquiera, “Spotify Connection” por ejemplo
- **Flow type** → Authorization Code
- **Authorize URI** → <https://accounts.spotify.com/authorize>

- **Token URI** → `https://accounts.spotify.com/api/token`
- **Scope** → Vacío
- **Client ID** → Ver en la app creada en **Spotify Developer**
- **Client Secret** → Ver en la app creada en **Spotify Developer**

Una vez hecha y verificada la conexión, pondremos lo siguiente, pero el `4.value` debemos seleccionar lo que ha salido del Iterator anterior. Donde dice `market=ES` debemos cambiar a nuestro código de país (ES = España, MX = México, US = Estados Unidos,...):

```
1 https://api.spotify.com/v1/artists/{{[VARIABLE]}}/albums?include_groups=album,single,ep&market
```

⚠ Importante que **Parse response** esté en **YES**.

The screenshot shows a configuration panel for an HTTP request. It includes the following fields and options:

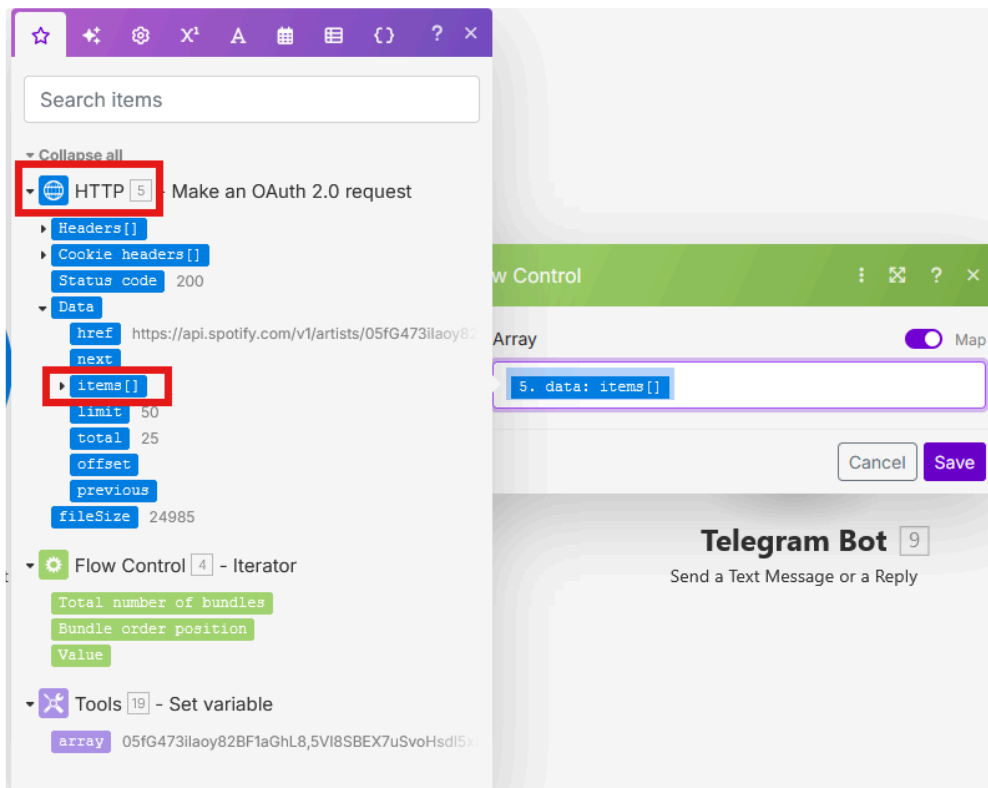
- URL \***: A text box containing the URL `https://api.spotify.com/v1/artists/4.Value/albums?include_groups=album,single,ep&market=ES&limit=50`. The `4.Value` part is highlighted in green.
- Method \***: A dropdown menu set to `GET`. A `Map` toggle is visible to the right.
- Headers**: A section with a `+ Add a header` button. A `Map` toggle is visible to the right.
- Query String**: A section with a `+ Add parameter` button. A `Map` toggle is visible to the right.
- Body type**: A dropdown menu.
- Parse response**: A checkbox that is checked, with radio buttons for `Yes` (selected) and `No`.

## Nodo 4 - Iterator (2)

### Flow Control > Iterator

Antes de esto, tendrás que ejecutar la automatización al menos una vez, para asegurarte de que la solicitud HTTP obtiene la información necesaria. Aquí simplemente tenemos que poner los datos que han salido de la operación anterior, en este caso `items[]`:

⚠ Importante que la opción **Map** esté activada.

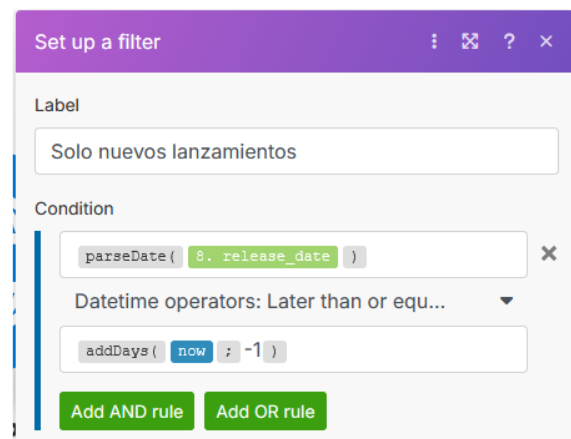
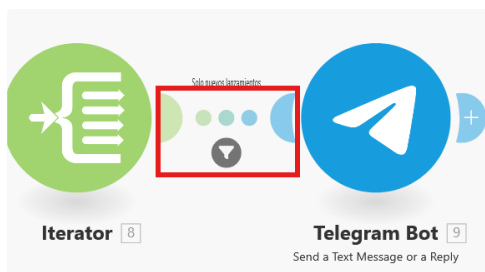


## Nodo 5 - Mensaje de Telegram

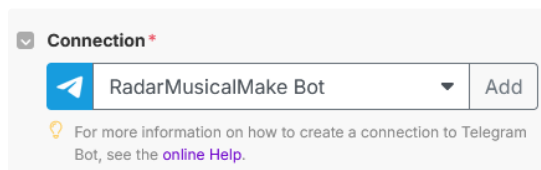
Telegram Bot > Send a Text Message or a Reply

Antes de meternos a configurar este módulo, haremos clic en la conexión entre el **Iterator** y el **Telegram Bot** para aplicar el siguiente filtro. Donde dice **8. release\_date** debemos poner los datos de release date del Iterator anterior. Este filtro es para que sólo pille lanzamientos de hace 24 horas, por lo que esta automatización habría que correrla cada día, pero se puede configurar para que lo haga semanalmente cambiando el **-1** por **-7**, o por los días que se quiera:

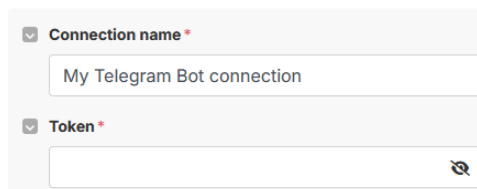
```
1 {{parseDate([VARIABLE])}}
2 {{addDays(now; -1)}}
```



Ahora sí, configuramos el mensaje de Telegram. Primero creamos la conexión con el bot. En token debemos poner el token que hemos sacado del bot de Telegram en la sección **Telegram Bot**:



Clicamos en **Add**



Una vez hecha la conexión, configuramos EL Chat ID que también lo hemos sacado anteriormente en la sección **Telegram Bot** y luego simplemente pondremos el mensaje a nuestro gusto, con los datos sacados del último Iterator: `artists[]:name` (nombre de artista) y `name` (nombre de canción/álbum):

```
1 🎵 Nuevo lanzamiento:  
2 {{8.artists[].name}} - {{8.name}}
```



Una vez hecho todo, podemos probarlo poniendo el filtro de este mismo apartado en `-365` en vez de `-1` para que pille los lanzamientos del último año, a ver si lanza el mensaje. Luego acordarse de volver a ponerlo en `-1` o lo que queráis.

Por último, hay que activar el escenario y poner que se ejecute cada cierto tiempo. Esto depende de vuestras preferencias.

✖ Acordaos de ajustar el filtro anteriormente mencionado dependiendo de cada cuánto lo ejecutéis, ya que si se ejecuta una vez a la semana pero el filtro está en `-1`, se ejecutará cada semana pero sólo pillará los lanzamientos del último día, por lo que en este caso habría que ponerlo en `-7`.



Schedule settings

**Run scenario:**  
Every day

**Time:**  
09:00

Time zone: Europe/Madrid  
Format: H:mm

☐ Show advanced settings

Cancel Save

Run once ☒ Daily at 9:00

Mensaje de ejemplo:

