



Tetiana Prykhodko 48978

Łukasz Gordon 41110

Jacek Ziółkowski 40978

# Spotify Dashboard

# What is Spotify Dashboard?

A Spotify dashboard is a streamlit application that provides users with various insights and functionalities related to Spotify data. It involves integrating data from the Spotify API and other web page to create a visual and interactive representation of music-related information. Spotify is a leading global music streaming platform that offers users access to an extensive library of songs, albums, and playlists from various genres and artists. With its user-friendly interface and personalized recommendations, Spotify revolutionizes the way people discover, enjoy, and share music. The platform supports both free and premium subscription models, providing users with a seamless and immersive audio streaming experience across multiple devices. The dashboard consists of 6 tabs: Spotify Dashboard, General, Artist, Country, Matrix and info.

The first tab is a welcome page briefly summarizing what you can expect from the dashboard and what it's about. The last tab a thank you page with dashboard creators information.

The General tab provides a variety of general information including most streamed songs and albums of all time, artists with most followers in selected country, total streams, most streamed artists, artists who gained the most streams. It also provides insightful charts: a bar chart describing most streamed songs globally, a stacked bar chart with streams by each genre over time, a pie chart with top 10 most streamed artists and top 10 artists with most followers. The results can be freely adjusted by selecting given artist, country or day using a selection box on the left hand side. Preferred artist to be selected is The Weekend and United States as a country.

The Country tab focuses solely on numbers of streams per country. It showcases 2 bar charts of biggest streams gainers and losers in 24 hours and 7 day time period. There is as well a tree map chart to see how much each country contributes to the total number of streams on its continent which can be selected using a selection sidebar. Finally a chart showing the change of streams over time in each country illustrated on a world map.

The Artist tab allows to deep dive into particular artist's performance. From the drop down selection bar on the top you can select an artist name and you will see his image, details and statistic like best performing songs and streams by each song.

The Matrix tab proposes a complex matrix chart which can visualize the data based on user's preferences and selection. Here an example of how it can be configured.

The image shows a configuration interface for a Spotify dashboard, divided into two main sections on a green background.

**Left Section: Please filter here:**

- Select Date Range:** A range slider showing dates from 2023-12-10 to 2023-12-14.
- Select Country:** A dropdown menu currently set to "United Kingdom".
- Select Streams Range:** A range slider showing values from 1012.00 to 6051677.00.
- Select Followers Range:** A range slider showing values from 5562.00 to 96270337.00.

**Right Section: Dimensions**

- X-Axis Dimension:** A dropdown menu set to "7Day".
- Y-Axis Dimension:** A dropdown menu set to "Streams+".
- Color Dimension:** A dropdown menu set to "Artist&Title".
- Size Dimension:** A dropdown menu set to "Followers".

In summary, a Spotify dashboard is a tool that leverages the Spotify API and other data sources to create an informative and visually appealing interface for users to explore and analyze music-related data.

## Who can use it and what for?

The Dashboard is primarily tailored for music industry professionals, such as record label executives and artist managers, it provides invaluable insights into emerging artists, regional preferences, and streaming trends. Meanwhile, business analysts leverage its capabilities for market analysis, enabling them to refine marketing strategies, optimize release schedules, and evaluate campaign effectiveness.

Beyond the business realm, public administration entities find value in the dashboard for cultural initiatives, tourism planning, and informed resource allocation. Researchers can delve into sociological and cultural studies, analyzing streaming data to understand the global impact and evolution of music culture.

Additionally, the dashboard facilitates behavioral studies, offering insights into the intricate relationship between music consumption and broader societal trends. From a business perspective, the Spotify Dashboard holds the key to market segmentation, enabling tailored strategies based on regional streaming preferences. Decision-makers can use its data-driven insights for signing decisions, concert planning, and revenue optimization. In the scientific arena, the dashboard provides a rich dataset for cultural studies and behavioral analysis, fostering a deeper understanding of the dynamic interplay between music and society. With an intuitive design that simplifies the exploration of artists, countries, and streaming popularity, the dashboard empowers regular music enthusiasts to unearth hidden gems and broaden their musical horizons.. In essence, the Spotify Dashboard not only serves the needs of businesses, science, and public administration but also democratizes music discovery

# Data

The provided code demonstrates the process of obtaining information from the Spotify API using OAuth 2.0 authentication. Here's a breakdown of the API request steps:

## 1. Authorization Header:

In order to authenticate with the Spotify API, the script constructs a Base64-encoded string from the client ID and client secret. This string is included in the Authorization header of the HTTP POST request to the Spotify Accounts service.

```
def get_token():
    auth_string = client_id+":"+client_secret
    auth_bytes = auth_string.encode("utf-8")
    auth_base64 = str(base64.b64encode(auth_bytes), "utf-8")

    url="https://accounts.spotify.com/api/token"
    headers={
        "Authorization":"Basic "+auth_base64,
        "Content-Type":"application/x-www-form-urlencoded"
    }
```

5

## 2. Token Request:

The script makes a POST request to the Spotify Accounts service at the URL "<https://accounts.spotify.com/api/token>" to obtain an access token. The payload of the request includes the grant type, which is set to "client\_credentials" in this case.

```
url="https://accounts.spotify.com/api/token"
headers={
    "Authorization":"Basic "+auth_base64,
    "Content-Type":"application/x-www-form-urlencoded"
}
data={"grant_type":"client_credentials"}
result=post(url, headers=headers, data=data)
json_result=json.loads(result.content)
token=json_result["access_token"]
```

## 3. API Request with Access Token:

Once the access token is obtained, subsequent requests to the Spotify API include the token in the Authorization header. In the example code, the `get_auth_header()` function is used to construct the Authorization header with the access token

```
def get_auth_header(token):  
    return {"Authorization": "Bearer "+token}
```

#### 4. Artist Search Request:

The `search_for_artist(art_name)` function demonstrates how to use the obtained access token to search for a specific artist on the Spotify API. It constructs the appropriate URL, makes a GET request, and parses the JSON response.

```
def search_for_artist(art_name):  
    token = get_token()  
    url="https://api.spotify.com/v1/search"  
    headers=get_auth_header(token)  
    query=f"?q={art_name}&type=artist&limit=1"  
    query_url=url+query  
    res=get(query_url, headers=headers)  
    json_result = json.loads(res.content)  
    return json_result
```

The Python script provided exemplifies an efficient web scraping approach to gather Spotify-related statistics from the Kwordb website. It encompasses multiple aspects, including country-specific daily charts, artist information, and top lists for different years and all-time top songs.

#### 1. Country-Specific Daily Charts:

- Dynamically retrieves today's date and formats it for folder creation.
- Disables SSL warnings to ensure seamless data retrieval.
- Parses HTML with BeautifulSoup, extracting country names and daily chart links.
- Stores the fetched data in Excel files within the "Country" directory.

#### 2. Artist Page:

- Expands functionality to scrape data from Kwordb's artist page.
- Creates a new folder for today's date in the "Artist" directory.
- Captures a general artist overview and saves it as an Excel file.

#### 3. Top Lists for Different Years:

- Fetches data from Kword's top lists page, generating a DataFrame.
- Saves the resulting DataFrame as "Top\_years.xlsx" in the "xlsx\_results" directory

#### 4. All-Time Top List of Songs:

- Extracts data from Kword's all-time top list of songs.
- Structures the data with separate 'Artist' and 'Title' columns.
- Saves the DataFrame, appending the current date, in the "top\_2500" directory.

This Python script demonstrates an effective web scraping strategy for Spotify data analysis, utilizing BeautifulSoup and functions from the "Functions" module. It systematically collects and organizes information on country-specific charts, artist statistics, and top song lists from the Kword website, facilitating subsequent data analysis and providing insights into Spotify's dynamic music landscape. Keeping loading times of the dashboard in mind we have reduced our data source to 3831 rows. In order to run the dashboard its necessary to update paths in each .py file and then use the streamlit run “path to .py file” command.

## Contribution

- Łukasz Gordon - Responsible for structuring project data and building models based on available data - Created and organized a comprehensive dashboard, including overview, general, artist, and half of the country, matrix, and info tabs - Participated in the creation of project report
- Tetiana Prykhodko utilized a combination of web scraping and API requests to contribute to the project, enabling the extraction of Spotify-related data from the Kword website and the Spotify API.
- Jacek Ziółkowski - Responsible for creation of the dashboard and report.

## Resources

Extensive internet research for charts, solutions, and visualization ideas including streamlit and plotly documentation ie. <https://docs.streamlit.io/library/api-reference/charts>. Implementing visualization ideas later leveraged by ChatGPT to refine and correct code ensuring accuracy in the dashboard implementation.