Assignment: API Testing Using Postman - Spotify API

✅ **Objective:**

You are required to perform API testing on the Spotify Web API using Postman. The assignment includes setting up authentication, sending API requests, and validating responses based on real-world scenarios.

🛠️ **[10 %] Part 1: Authentication Setup Steps**

1. **Create a Spotify Developer Account**
    - Go to https://developer.spotify.com.
    - Sign up or log in with your Spotify account.
2. **Register Your App**
    - Go to the **Dashboard**, create a new app.
    - Note down your Client ID and Client Secret.
3. **Obtain Access Token (Authorization)**
    - Use **Spotify's Web Console** for quick testing:
        - https://developer.spotify.com/console
    - Alternatively, set up OAuth 2.0 in Postman:
        - **Authorization Type:** OAuth 2.0
        - **Add Token URL:** https://accounts.spotify.com/api/token
        - Provide Client ID and Client Secret
        - Use Client Credentials
4. **Set Environment Variables in Postman**
    - Create variables:
        - access_token → Your generated token
        - base_url → https://api.spotify.com/v1

5. **Test Authentication**
    - Send a GET request to:

{{base_url}} /albums/4aawyAB9vmqN3uQ7FjRGTy

    - Ensure the status code is 200 OK.
    - You are now authenticated and ready to proceed.

Request -> url

Body ->

Request Type ->

Expected Response ->

🧩 Part 2: High-Level API Testing Scenarios

✅ 1. Search for Tracks by Keyword [5 %]

Endpoint:
GET https://api.spotify.com/v1/search?q=love&type=track

Purpose:
Verify that searching for tracks returns relevant results.

Test Points:

- Status code = 200 OK
- Response contains tracks.items[]
- Validate that track names or album titles contain the search keyword

---

✅ 2. Search for Albums by Artist Name [5 %]

Endpoint:
GET https://api.spotify.com/v1/search?q=Eminem&type=album

Purpose:
Confirm album search returns valid results for a given artist.

Test Points:

- Status code = 200 OK
- Response includes album details (name, id, release_date)
- Validate album list is not empty

✅ 3. Get Album Details by Album ID [10 %]

Endpoint:
GET https://api.spotify.com/v1/albums/{album_id}

Purpose:
Retrieve detailed information for a specific album.

Test Points:

- Status code = 200 OK
- Validate presence of:
    - album_type
    - artists[]
    - tracks.items[]
    - release_date

✅ 4. Get Track Details by Track ID [10 %]

Endpoint:
GET https://api.spotify.com/v1/tracks/{track_id}

Purpose:
Check if track details are retrieved successfully.

Test Points:

- Status code = 200 OK
- Validate:
    - name
    - duration_ms
    - popularity
    - album

✅ 5. Get Audio Features for a Track [10 %]

Endpoint:
GET https://api.spotify.com/v1/audio-features/{track_id}

Purpose:
Ensure audio analysis features (tempo, key, energy, etc.) are returned.

Test Points:

- Status code = 200 OK
- Response includes:
    - danceability
    - energy
    - tempo
    - valence
    - liveness

✅ 6. Get Audio Analysis for a Track [10 %]

Endpoint:
GET https://api.spotify.com/v1/audio-analysis/{track_id}

Purpose:
Retrieve detailed audio structure analysis of a track.

Test Points:

- Status code = 200 OK
- Validate presence of:
    - bars[]
    - beats[]
    - sections[]
    - segments[]

✅ 7. Browse Categories [10 %]

Endpoint:
GET https://api.spotify.com/v1/browse/categories

Purpose:
Test browsing available content categories (e.g., genres, moods).

Test Points:

- Status code = 200 OK
- Response includes categories.items[]
- Validate categories like "Pop", "Workout", "Chill" are present

---

✅ 8. Get Playlists for a Category [10 %]

Endpoint:
GET https://api.spotify.com/v1/browse/categories/{category_id}/playlists

Purpose:
Retrieve playlists under a specific category (e.g., "Pop").

Test Points:

- Status code = 200 OK
- Response contains playlists with:
    - name
    - id
    - tracks.total

✅ 9. Get Featured Playlists [10 %]

Endpoint:
GET https://api.spotify.com/v1/browse/featured-playlists

Purpose:
Validate featured playlists are displayed correctly.

Test Points:

- Status code = 200 OK
- Response contains:
    - message (e.g., "Today's Top Picks")
    - playlists.items[]

✅ 10. Get New Releases [10 %]

Endpoint:
GET https://api.spotify.com/v1/browse/new-releases

Purpose:
Verify latest album releases are returned by the API.

Test Points:

- Status code = 200 OK
- Response includes:
    - albums.items[]
    - name, release_date, total_tracks

Delivery & Deadline:

Upload your postman collection on GitHub, then send your repo link via email aabdelhafeez.route@gmail.com:

- [*+10 Points*] **Early Birds**: Saturday 28th June 2025 @ 12 AM
- **Final Date**: Thursday 4th July 2025 @ 12 AM
- **Follow the following schema for sending email:**

  To: aabdelhafeez.route@gmail.com
  Subject: API Assignment 1
  Body:

Full Name	(e.g. aaaaa mmmmm aaaaa mmm kkkk)
Group Time (Sat/Mon&Thur)	(e.g. Sat)
Group Type (Online/Offline)	(e.g. Online)
Repo Link	(e.g. https://github.com/repo)