

Spotify Web API

This Postman Collection contains a set of automated tests designed to verify the functionality of the Spotify Web API.

It focuses on CRUD operations related to playlists, including creating playlists, adding and removing tracks and reordering them.

The collection is divided into positive and negative test suites to ensure both valid and invalid behaviors are properly validated.

Main goals:

- Verify correct API responses for valid requests (functional testing).
- Ensure appropriate error handling and response messages for invalid inputs.
- Validate response codes, structure, messages according to Spotify API documentation.

AUTHORIZATION OAuth 2.0

Positive Endpoints

This folder contains tests that verify correct and expected behavior of the Spotify API when valid data is provided.

These tests ensure that the API successfully processes requests such as:

- creating a new playlist with valid input
- adding valid tracks to a playlist,
- successfully reordering or removing existing tracks.
- retrieving playlist details and verifying correct response status and structure.

Expected outcome:

All requests should return 2xx status codes and valid JSON responses according to the Spotify API specification.

AUTHORIZATION OAuth 2.0

This folder is using OAuth 2.0 from collection [Spotify Web API](#)

Playlist

AUTHORIZATION OAuth 2.0

This folder is using OAuth 2.0 from collection [Spotify Web API](#)

POST Create Playlist



/users//playlists

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

Body raw (json)

```
json

{
  "name": "No Name Playlist",
  "description": "New playlist description",
  "public": false
}
```

POST Add Items to Playlists



/playlists//tracks

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

Body raw (json)

```
json

{
  "uris": [
    "spotify:track:480j122Gpi2520Ify4SNzm"
  ],
  "position": 0
}
```

PUT Update playlist tracks order



/playlists//tracks

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

Body raw (json)

json

```
{
  "range_start": 1,
  "insert_before": 5,
  "range_length": 2
}
```

DELETE Remove Playlist Items



/playlists//tracks

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

Body raw (json)

json

```
{
  "tracks": [
    {
      "uri": "spotify:track:480j122Gpi2520Ify4SNzm"
    }
  ],
  "snapshot_id": ""
}
```

GET User's Playlists



/me/playlists

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

User Profile

AUTHORIZATION OAuth 2.0

This folder is using OAuth 2.0 from collection [Spotify Web API](#)

GET User Profile



/me

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

Tracks

AUTHORIZATION OAuth 2.0

This folder is using OAuth 2.0 from collection [Spotify Web API](#)

GET Track



/tracks/

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

GET Users Saved Track



/me/tracks/

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

PUT Save Tracks for Current User



/me/tracks

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

Body raw (json)

```
json

{
  "ids": [
    "0JP9xo3adEtGSdUEISisZL"
  ]
}
```

DELETE Remove User's Saved Tracks



/me/tracks?ids=0JP9xo3adEtGSdUEISisZL

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

PARAMS

ids	0JP9xo3adEtGSdUEISisZL
-----	------------------------

Body raw (json)

```
json
```

```
{
  "ids": [
    "0JP9xo3adEtGSdUEISisZL"
  ]
}
```

Negative Endpoints

This folder includes tests that verify how the Spotify API handles invalid, incomplete or unexpected data.

These tests help identify how well the API validates inputs and manages errors.

Example include:

- Creating a playlist with an empty or missing name
- Adding tracks using an invalid uri
- attempting reorder tracks that are out of bounds
- deleting tracks using an invalid trackId

Expected outcome:

The API should return appropriate error responses (4xx or 5xx) with meaningful messages that describe the issue.

AUTHORIZATION OAuth 2.0

This folder is using OAuth 2.0 from collection [Spotify Web API](#)

Playlist

AUTHORIZATION OAuth 2.0

This folder is using OAuth 2.0 from collection [Spotify Web API](#)

POST Add Items to Playlist - invalid uri



/playlists//tracks

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

Body raw (json)

json

```
{
  "uris": [
    "spotify:track:480j122Gpi"
  ],
  "position": 0
}
```

POST Add Items to Playlist - unauthorized attempt

/playlists//tracks

Body raw (json)

json

```
{
  "uris": [
    "spotify:track:480j122Gpi2520Ify4SNzm"
  ],
  "position": 0
}
```

POST Create Playlist - without name



/users//playlists

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

Body raw (json)

json

```
{
```

```
"name": "",
"description": "New playlist description",

"public": false
}
```

POST Create Playlist - unauthorized

/users//playlists

Body raw (json)

json

```
{
  "name": "No Name Playlist",
  "description": "New playlist description",
  "public": false
}
```

POST Create Playlist - invalid data type



/users//playlists

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

Body raw (json)

json

```
{
  "name": "No name",
  "description": "New playlist description",
  "public": "yees"
}
```

PUT Reorder Playlist - playlist contain only one track



/playlists//tracks

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

Body raw (json)

json

```
{
  "range_start": 1,
  "insert_before": 5,
  "range_length": 2
}
```

DELETE Remove Playlist Items - invalid uri



/playlists//tracks

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

Body raw (json)

json

```
{
  "tracks": [
    {
      "uri": "spotify:track:480j122G"
    }
  ],
  "snapshot_id": ""
}
```

DELETE Remove Playlist Items - invalid snapsgotld



/playlists//tracks

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

Body raw (json)

```
json

{
  "tracks": [
    {
      "uri": "spotify:track:480j122Gpi2520Ify4SNzm"
    }
  ],
  "snapshot_id": "234r5t6y7ui8o9poikujhygtrfeddsfghjk"
}
```

DELETE Remove Playlist Items - from non existent playlist



/playlists/4Jdy6m6sa8IHztbWadsfgf/tracks

AUTHORIZATION OAuth 2.0

This request is using OAuth 2.0 from collection [Spotify Web API](#)

Body raw (json)

```
json

{
  "tracks": [
    {
      "uri": "spotify:track:480j122Gpi2520Ify4SNzm"
    }
  ],
  "snapshot_id": "234r5t6y7ui8o9poikujhygtrfeddsfghjk"
}
```