



TECHOVERTM

Self Made

Spotify Project

API Manual

1. GET current user information

Spotify API Reference:

Get Current User's Profile**GET**

Response Example

```
1  {
2    "country": "string",
3    "display_name": "string",
4    "email": "string",
5    "explicit_content": {
6      "filter_enabled": true,
7      "filter_locked": true
8    },
9    "external_urls": {
10     "spotify": "string"
11   },
12   "followers": {
13     "href": "string",
14     "total": 0
15   },
16   "href": "string",
17   "id": "string",
18   "images": [
19     {
20       "url": "https://i.scdn.co/image/ab67616d00001e02ff9ca1d
21       "height": 300,
22       "width": 300
23     }
24   ],
25   "product": "string",
26   "type": "string",
27   "uri": "string"
28 }
```

Spotify Web API NPM package:

```
// Get the authenticated user
spotifyApi.getMe()
  .then(function(data) {
    console.log('Some information about the authenticated user', data.body)
  }, function(err) {
    console.log('Something went wrong!', err);
  });
```

2. Get all Playlists

Spotify API Reference:

Get User's Playlists**GET**

Response Example

```
1  {
2    "href": "https://api.spotify.com/v1/me/shows?offset=0&limit=20",
3    "items": [
4      {}
5    ],
6    "limit": 20,
7    "next": "https://api.spotify.com/v1/me/shows?offset=1&limit=20",
8    "offset": 0,
9    "previous": "https://api.spotify.com/v1/me/shows?offset=1&limit=20",
10   "total": 4
11 }
```

Spotify Web API NPM package:

```
// Get a user's playlists
spotifyApi.getUserPlaylists('thelinmichael')
  .then(function(data) {
    console.log('Retrieved playlists', data.body);
  }, function(err) {
    console.log('Something went wrong!', err);
  });
```

3. GET devices

Spotify API Reference:

Get Available Devices

GET

Response Example

```
1  {
2    "devices": [
3      {
4        "id": "string",
5        "is_active": true,
6        "is_private_session": true,
7        "is_restricted": true,
8        "name": "Loudest speaker",
9        "type": "computer",
10       "volume_percent": 59
11     }
12   ]
13 }
```

Spotify Web API NPM package:

```
// Get a User's Available Devices
spotifyApi.getMyDevices()
  .then(function(data) {
    let availableDevices = data.body.devices;
    console.log(availableDevices);
  }, function(err) {
    console.log('Something went wrong!', err);
  });
```

4. Transfer a Playback

Spotify API Reference:

Transfer Playback

PUT

Request

PUT /me/player

▼ Body

device_ids

array of strings

A JSON array containing the ID of the device on which playback should be started/transferred.
For example: {device_ids:["74ASZWbe4IXaub836trGX"]}

Note: Although an array is accepted, only a single device_id is currently supported. Supplying more than one will return **400 Bad Request**

play

boolean

true:

ensure playback happens on new device.

false

or not provided: keep the current playback state.

application/json

required

Responses

204 401 403 429

Playback transferred

Spotify Web API NPM package:

```
// Transfer a User's Playback
spotifyApi.transferMyPlayback(deviceIds)
  .then(function() {
    console.log('Transferring playback to ' + deviceIds);
  }, function(err) {
    //if the user making the request is non-premium, a 403 FORBIDDEN response will be returned
    console.log('Something went wrong!', err);
  });
```


5. Play a playlist / song

Spotify API Reference:

Start/Resume Playback

PUT

Request

PUT /me/player/play

Query

device_id

string

The id of the device this command is targeting. If not supplied, the user's currently active device is the target.
Example value: "0d1841b0976bae2a3a310dd74c0f3df354899bc8"

Body

application/json

context_uri

string

uris

array of strings

Array of URIs

offset

object

object
AdditionalProperties

position_ms

integer

integer

Responses

204401403429

Playback started

Spotify Web API NPM package:

```
// Start/Resume a User's Playback
spotifyApi.play()
  .then(function() {
    console.log('Playback started');
  }, function(err) {
    //if the user making the request is non-premium, a 403 FORBIDDEN response
    console.log('Something went wrong!', err);
  });
```

6. Pause Playback

Spotify API Reference:

Pause Playback

PUT

Request

PUT /me/player/pause

Query

device_id

string

The id of the device this command is targeting. If not supplied, the user's currently active device is the target.
Example value: "0d1841b0976bae2a3a310dd74c0f3df354899bc8"

Responses

204401403429

Playback paused

Spotify Web API NPM package:

```
// Pause a User's Playback
spotifyApi.pause()
  .then(function() {
    console.log('Playback paused');
  }, function(err) {
    //if the user making the request is non-premium, a 403 FORBIDDEN response
    console.log('Something went wrong!', err);
  });
```

7. Skip to next track

Spotify API Reference:

Skip To Next

POST

Request

POST /me/player/next

▼ Query

device_id

string

The id of the device this command is targeting. If not supplied, the user's currently active device is the target.
Example value: "0d1841b0976bae2a3a310dd74c0f3df354899bc8"

Responses

204 401 403 429

Command sent

Spotify Web API NPM package:

```
// Skip User's Playback To Next Track
spotifyApi.skipToNext()
  .then(function() {
    console.log('Skip to next');
  }, function(err) {
    //if the user making the request is non-premium, a 403 FORBIDDEN response
    console.log('Something went wrong!', err);
  });
```

8. Skip to previous track

Spotify API Reference:

Skip To Previous

POST

Request

POST /me/player/previous

▼ Query

device_id

string

The id of the device this command is targeting. If not supplied, the user's currently active device is the target.
Example value: "0d1841b0976bae2a3a310dd74c0f3df354899bc8"

Responses

204 401 403 429

Command sent

Spotify Web API NPM package:

```
// Skip User's Playback To Previous Track
spotifyApi.skipToPrevious()
  .then(function() {
    console.log('Skip to previous');
  }, function(err) {
    //if the user making the request is non-premium, a 403 FORBIDDEN response
    console.log('Something went wrong!', err);
  });
```


9. Seek to song position

Spotify API Reference:

Seek To Position

PUT

Request

PUT /me/player/seek

▼ Query

position_ms

integer

required

The position in milliseconds to seek to. Must be a positive number. Passing in a position that is greater than the length of the track will cause the player to start playing the next song.
Example value: 25000

device_id

string

The id of the device this command is targeting. If not supplied, the user's currently active device is the target.
Example value: "0d1841b0976bae2a3a310dd74c0f3df354899bc8"

Responses

204401403429

Command sent

Spotify Web API NPM package:

```
// Seek To Position In Currently Playing Track
spotifyApi.seek(positionMs)
  .then(function() {
    console.log('Seek to ' + positionMs);
  }, function(err) {
    //if the user making the request is non-premium, a 403 FORBIDDEN response
    console.log('Something went wrong!', err);
  });
```

10. Set Volume

Spotify API Reference:

Set Playback Volume

PUT

Request

PUT /me/player/volume

▼ Query

volume_percent

integer

required

The volume to set. Must be a value from 0 to 100 inclusive.
Example value: 50

device_id

string

The id of the device this command is targeting. If not supplied, the user's currently active device is the target.
Example value: "0d1841b0976bae2a3a310dd74c0f3df354899bc8"

Responses

204401403429

Command sent

Spotify Web API NPM package:

```
// Set Volume For User's Playback
spotifyApi.setVolume(50)
  .then(function () {
    console.log('Setting volume to 50.');
```

Klar med Projektet?

Posta en video av er Spotify Klon på era sociala medier och tagga @techoveracademy !

Vi är sjukt stolta över dig.
Lets keep crushing it!