

Spotify Project

API Manual



## 1. GET current user information

#### **Spotify API Reference:**

## Get Current User's Profile GET

#### Response Example

```
"country": "string",
      "display_name": "string",
      "email": "string",
      "explicit_content": {
        "filter_enabled": true,
       "filter_locked": true
      "external_urls": {
        "spotify": "string"
10
      "followers": {
       "href": "string",
        "total": 0
14
15
      "href": "string",
      "id": "string",
      "images": [
19
         "url": "https://i.scdn.co/image/ab67616d00001e02ff9ca10
20
         "height": 300,
21
          "width": 300
22
23
24
      "product": "string",
      "type": "string",
      "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
3    "items": [
4         {}
5        ],
6         "limit": 20,
7         "next": "https://api.spotify.com/v1/me/shows?offset=1&limit
8         "offset": 0,
9         "previous": "https://api.spotify.com/v1/me/shows?offset=1&
10         "total": 4
11 }
```

```
// 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:**

#### **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

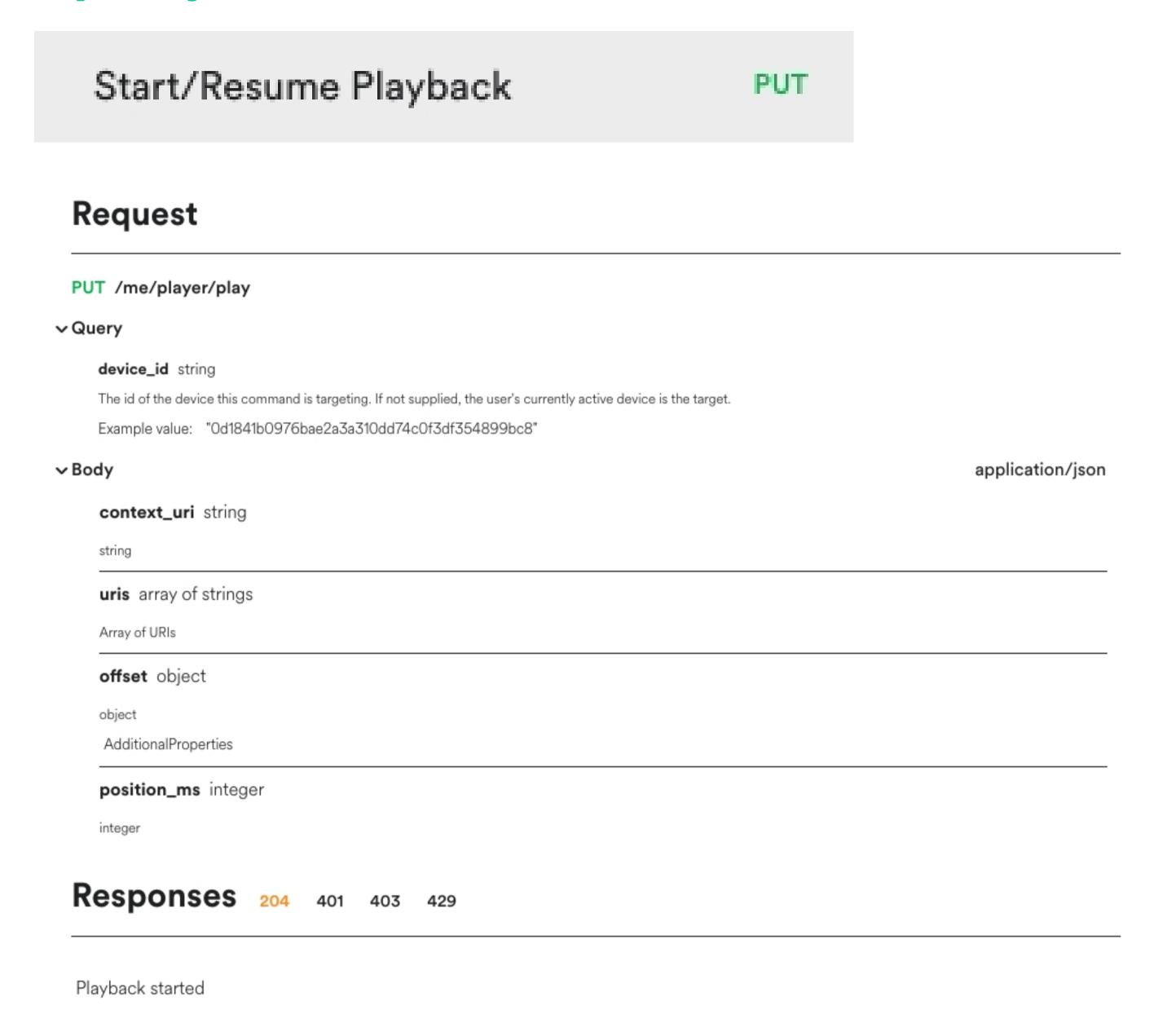
                                                                                                                         application/json
     device_ids array of strings
                                                                                                                                   required
     A JSON array containing the ID of the device on which playback should be started/transferred.
     For example: {device_ids:["74ASZWbe4IXaubB36ztrGX"]}
     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.
 Responses 204 401 403 429
  Playback transferred
```

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



## 5. Play a playlist / song

#### **Spotify API Reference:**

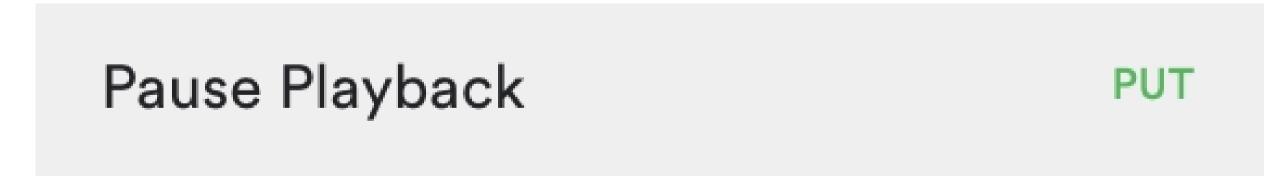


#### **Spotify Web API NPM package:**

```
// 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:**



### 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"

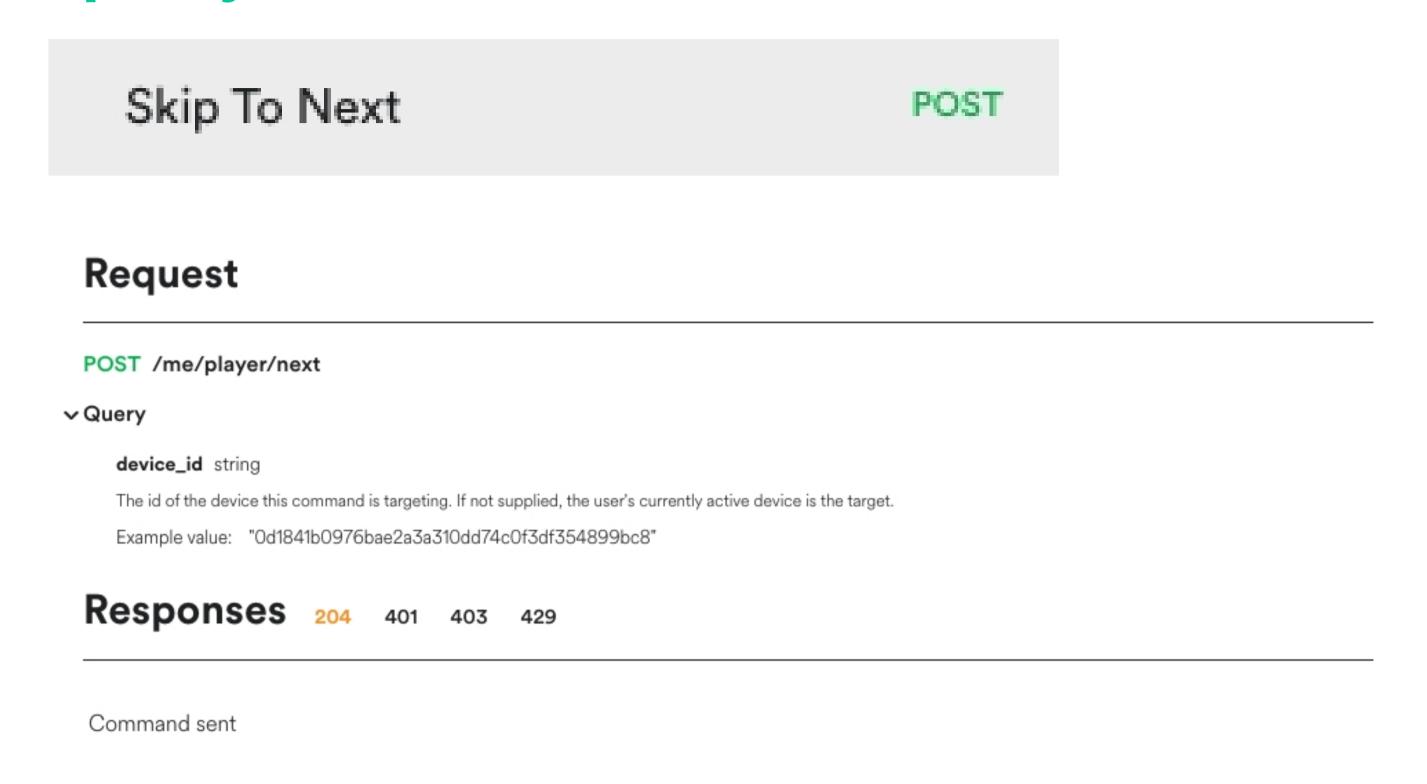
Responses 204 401 403 429

```
// 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:**



#### **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 responsole.log('Something went wrong!', err);
});
```

# 8. Skip to previous track

#### **Spotify API Reference:**

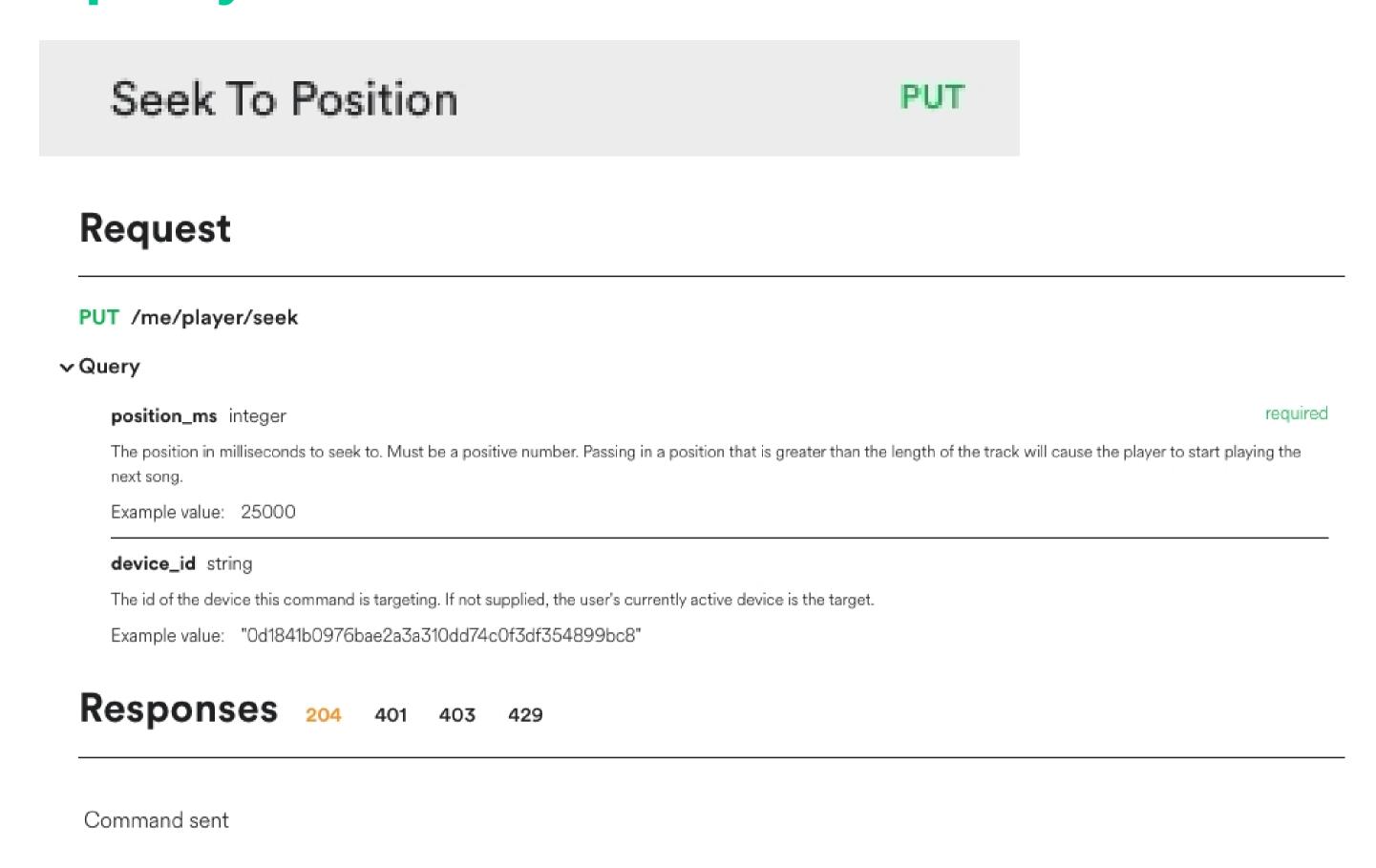


```
// Skip User's Playback To Previous Track
spotifyApi.skipToPrevious()
   .then(function() {
      console.log('Skip to previous');
   }, function(err) {
      //if the user making the request is non-premium, a 403 FORBIDDEN respondent console.log('Something went wrong!', err);
   });
```



## 9. Seek to song position

#### **Spotify API Reference:**

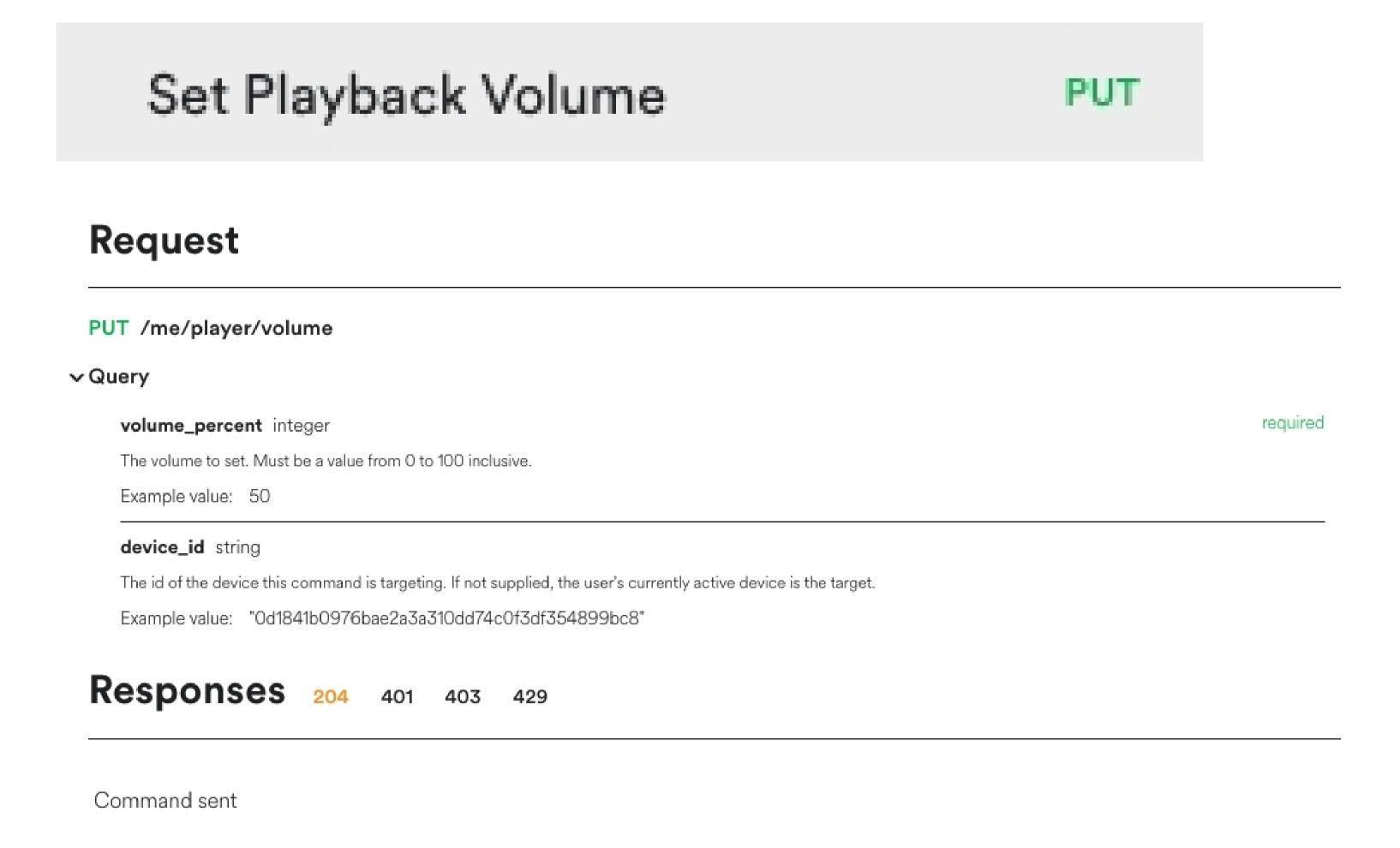


#### **Spotify Web API NPM package:**

```
// 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 respor console.log('Something went wrong!', err);
});
```

## 10. Set Volume

#### **Spotify API Reference:**



```
// Set Volume For User's Playback
spotifyApi.setVolume(50)
  .then(function () {
    console.log('Setting volume to 50.');
    }, function(err) {
        //if the user making the request is non-premium, a 403 FORBIDDEN respondence on sole.log('Something went wrong!', err);
    });
```

# 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!



www.techover.nu