

Tarea Individual 27 - Implementación de Autenticación OAuth 2.0 en Node.js con Express

Abro <http://localhost:3000> y me muestra el access token y el refresh token:

```
localhost:3000/callback?code=AQAOqywQWagtTVHYCsGlfFNZyCzJlu7MWVAKVgPMrKQXgEgMWUGv-VsDN5nN2Ugtu3fLkark1b3mNQGXMPKTI2MFnlZQlInvO-illPEmrTADn2m6Gp5MqaiylRkC4...

Dar formato al texto

{"access_token": "BQC6NFHXzUQX-gv1F3b2NTBX3IsV2c1YXc7dSeC5R0mOGM63V_Oc3Mk7GUg8eeziNCdIXiB5phD1fRYEwV29b8Z8FEaoyaoUmmMgJ66BavyXJSVqPczoG0xgKEL7EwXSZVipzdTPIMJw47AkiIwrM1Acqv-VrUCoA6kydKhgc4Gxz0gdF7W7MSYTFfVvkIPsI03k4P7_DeW6XGuowA-0Q91THAUefXzALXyfnLk_Xjd62u8215q5sAQ", "refresh_token": "AQBOOrVpY6oOd6aAKqSwBLKA6YGm1Z5eJ3GjBF308j9k7euuQ8QB_1SPj80qzn8Jp4015QKQBWr-tIKMM6wnMagYwtSFdZBFL7E8IBDB6pA-12n9y1090N3vz82uv8IHQFE"}
```

Compruebo los datos de mi usuario entrando en /me:

```
localhost:3000/me

Dar formato al texto

{"error":"Falta el token de acceso"}
```

Le pongo el token de acceso con curl y me da los datos de mi usuario:

```
PS C:\Users\... Desktop\Programación de Servicios y Procesos\Ejercicios\spotify-app> curl -X GET "http://localhost:3000/me" -H "Authorization: Bearer BQC6jW6w4t2h70eN1Wghu1Bu-1YsennKuzkbnMURcx0Wada1SBFC3jCkQym62wnUPSKauuf1dgg3Uml1Mpxdl-Bpyuhr61M4HilEx42ZL6u025nsaPZZVH_hBgtspqZ7UKEfW6ijrWVpX2B0LH1cIPo7L3mB9eIwNEU#kyYooGn-Iu25tya#QZY5B-HuVw_sn87W6B4JTseuBrsPh-huaga8MUAJ05B3HjprOA"

{"country": "ES", "display_name": "...", "email": "...@gmail.com", "explicit_content": {"filter_enabled": false, "filter_locked": false}, "external_urls": {"spotify": "https://open.spotify.com/user/jq8g6v1lcxSe5fjb79ngz6x"}, "followers": {"href": null, "total": 2}, "href": "...", "id": "...", "images": [{"height": 300, "url": "...", "width": 300}], "product": "premium", "type": "user", "uri": "spotify:user:..."}
```

Código:

```

JS server.js > ...
1  require('dotenv').config();
2  console.log(process.env.SPOTIFY_CLIENT_ID); // Debería mostrar tu CLIENT_ID
3  console.log(process.env.SPOTIFY_CLIENT_SECRET); // Debería mostrar tu CLIENT_SECRET
4  console.log(process.env.SPOTIFY_REDIRECT_URI); // Debería mostrar la URL de redirección
5  const express = require('express');
6  const cors = require('cors');
7  const axios = require('axios');
8  const querystring = require('querystring');
9
10 const app = express();
11 const PORT = 3000;
12
13 // Habilitar CORS para permitir peticiones desde otros dominios
14 app.use(cors());
15
16 // Ruta para iniciar el flujo de autenticación
17 app.get('/login', (req, res) => {
18   const authUrl = 'https://accounts.spotify.com/authorize?' +
19     querystring.stringify({
20       client_id: process.env.SPOTIFY_CLIENT_ID,
21       response_type: 'code',
22       redirect_uri: process.env.SPOTIFY_REDIRECT_URI,
23       scope: 'user-read-private user-read-email',
24     });
25   res.redirect(authUrl);
26 });
27
28 // Ruta de callback para manejar la respuesta del servidor de autorización
29 app.get('/callback', async (req, res) => {
30   const code = req.query.code || null;
31
32   try {
33     const tokenResponse = await axios.post('https://accounts.spotify.com/api/token',
34       querystring.stringify({
35         code: code,
36         redirect_uri: process.env.SPOTIFY_REDIRECT_URI,

```

```

37         grant_type: 'authorization_code',
38     }},
39     {
40         headers: {
41             'Authorization': 'Basic ' + Buffer.from(
42                 process.env.SPOTIFY_CLIENT_ID + ':' + process.env.SPOTIFY_CLIENT_SECRET
43             ).toString('base64'),
44             'Content-Type': 'application/x-www-form-urlencoded',
45         },
46     },
47 );
48
49 const { access_token, refresh_token } = tokenResponse.data;
50
51 res.json({ access_token, refresh_token });
52 } catch (error) {
53     res.status(500).json({ error: 'Error obteniendo el token' });
54 }
55 });
56
57 // Ruta para obtener datos del perfil del usuario
58 app.get('/me', async (req, res) => {
59     const accessToken = req.headers['authorization'] || req.headers['Authorization'];
60
61     if (!accessToken) {
62         return res.status(401).json({ error: 'Falta el token de acceso' });
63     }
64
65     console.log('Token recibido:', accessToken); // Verificar si el token está llegando
66
67     try {
68         const userResponse = await axios.get('https://api.spotify.com/v1/me', {
69             headers: { Authorization: accessToken },
70         });
71
72         res.json(userResponse.data);
73     } catch (error) {
74         console.error(error); // Ver detalles del error
75         res.status(400).json({ error: 'Error al obtener datos del usuario' });
76     }
77 });
78
79
80
81 // Iniciar el servidor
82 app.listen(PORT, () => {
83     console.log(`Servidor corriendo en http://localhost:${PORT}`);
84 });
85

```