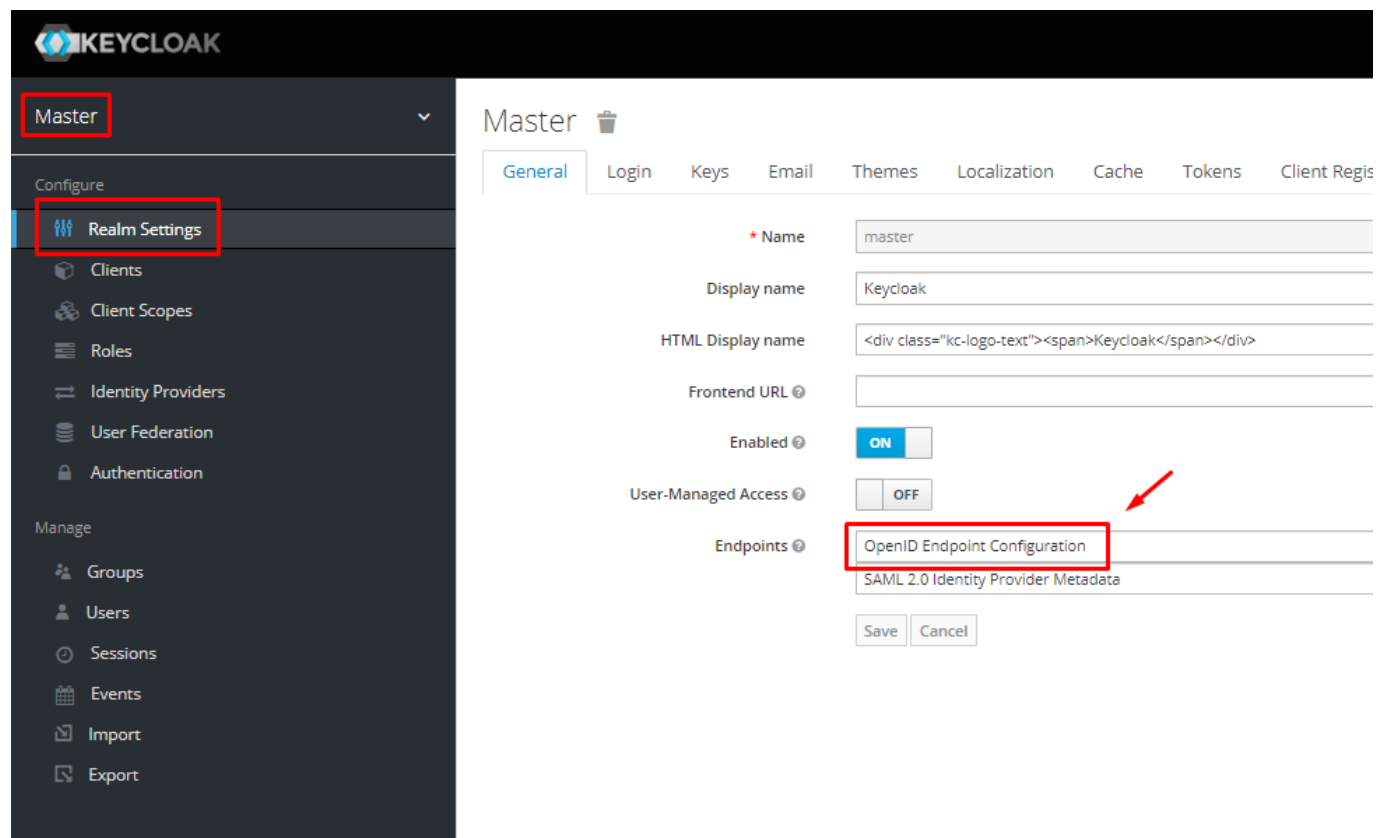


\_1. Conocer los endpoint de **OpenID OIDC** en **Keycloak**, para esto seleccionamos el realm «**master**», Configure «**Realm Settings**» y en la pestaña «**General**» damos clic donde dice «**OpenID Endpoint Configuration**»



The screenshot shows the Keycloak Admin Console interface. On the left sidebar, the 'Master' realm is selected, and 'Realm Settings' is highlighted under the 'Configure' section. The main content area shows the configuration for the 'master' realm in the 'General' tab. The 'Endpoints' section is visible, with 'OpenID Endpoint Configuration' highlighted by a red box and a red arrow pointing to it. Other fields include 'Name' (master), 'Display name' (Keycloak), 'HTML Display name' (HTML code), 'Frontend URL', 'Enabled' (ON), and 'User-Managed Access' (OFF).

Se abrirá una URL <http://localhost:8080/realms/master/.well-known/openid-configuration>

```
1 // 20220708233925
2 // http://localhost:8080/realms/master/.well-known/openid-configuration
3
4 {
5   "issuer": "http://localhost:8080/realms/master",
6   "authorization_endpoint": "http://localhost:8080/realms/master/protocol/openid-connect/auth",
7   "token_endpoint": "http://localhost:8080/realms/master/protocol/openid-connect/token",
8   "introspection_endpoint": "http://localhost:8080/realms/master/protocol/openid-connect/token/introspect",
9   "userinfo_endpoint": "http://localhost:8080/realms/master/protocol/openid-connect/userinfo",
10  "end_session_endpoint": "http://localhost:8080/realms/master/protocol/openid-connect/logout",
11  "frontchannel_logout_session_supported": true,
12  "frontchannel_logout_supported": true,
13  "jwks_uri": "http://localhost:8080/realms/master/protocol/openid-connect/certs",
14  "check_session_iframe": "http://localhost:8080/realms/master/protocol/openid-connect/login-status-iframe.html",
15  "grant_types_supported": [
16    "authorization_code",
17    "implicit",
18    "refresh_token",
19    "password",
20    "client_credentials",
21    "urn:ietf:params:oauth:grant-type:device_code",
22    "urn:openid:params:grant-type:ciba"
23  ],
24  "acr_values_supported": [
25    "0",
26    "1"
27  ],
28  "response_types_supported": [
29    "code",
30    "none",
31    "id_token",
32    ". . . "
```

2. Nos vamos a **Postman** y consultamos el token, donde:

- **Token method:** POST
- **Token endpoint:** http://localhost:8080/realms/master/protocol/openid-connect/token
- **x-www-form-urlencoded**
  - **grant\_type:** password
  - **client\_id:** admin-cli (es la aplicación cliente registrada en keycloak, se puede crear otra aplicación)
  - **username:** juliopari (previamente creado)
  - **password:** 123456 (previamente creado)

The screenshot shows a REST client interface for a POST request to `http://localhost:8080/realms/master/protocol/openid-connect/token`. The request body is a form-urlencoded object with the following parameters:

KEY	VALUE	DESCRIPTION
Content-Type	application/x-www-form-urlencoded	
grant_type	password	
client_id	admin-cli	
username	juliopari	
password	123456	

The response status is 200 OK. The response body is a JSON object containing an `access_token` field, which is a long alphanumeric string representing a JWT token. The token is displayed in a 'Raw' view.

**\_3. Copiamos el valor de «access\_token» que es un JSON Web Token y nos vamos a [jwt.io](https://jwt.io) para abrirlo**

The image shows a JWT token decoder interface. On the left, under 'Encoded', there is a long string of base64url characters. On the right, under 'Decoded', the token is shown in its structured form. The header contains: `{ "alg": "RS256", "typ": "JWT", "kid": "9uLgSmZMXBuAASft2CvJuhIM1eLQ8Fywa_Ung5mTXB4" }`. The payload contains: `{ "exp": 1657341900, "iat": 1657341840, "jti": "d7799475-1e0f-4efc-bce6-28bae26f600c", "iss": "http://localhost:8080/realms/master", "sub": "b1f68700-ef31-4163-8df4-8fae9ec1944b", "typ": "Bearer", "azp": "admin-cli", "session_state": "4fee38be-3438-492e-93f6-584f078de48f", "acr": "1", "scope": "email profile", "sid": "4fee38be-3438-492e-93f6-584f078de48f", "email_verified": false, "name": "Julio Pari", "preferred_username": "juliopari", "given_name": "Julio", "family_name": "Pari", "email": "info@juliopari.com" }`. The payload section is highlighted with a red border.

## Julio Pari (IT Architect IBM)



Si te ha interesado este artículo y deseas un apoyo o asesoría en algún requerimiento, envíame un mensaje a: (info@juliopari.com) o sino a través de LinkedIn: <https://www.linkedin.com/in/juliopari/>