

Setting Up Roles and Permissions in Keycloak

Roles and permissions in Keycloak define what users and applications are allowed to do. Roles can be assigned to users, groups, or clients, and are embedded into access tokens to enforce authorization. This guide explains how to define and manage roles via the Admin Console, REST API, and CLI, with best practices and common issues.

Creating Roles via Keycloak Admin Console

This is the easiest way to create and manage roles visually.

Access the Admin Console

Log in to:

Choose the appropriate realm.

Create Realm Roles

1. Go to **Roles > Add Role**
2. Enter:
 - **Role Name:** e.g., admin, viewer, editor
 - **Description:** Optional but recommended
3. Click **Save**

[Realm roles](#) > [Create role](#)

Create role

Role name *

Description

[Save](#) [Cancel](#)

Create Client Roles

1. Go to **Clients** > **[client-name]** > **Roles** > **Create Role**
2. Fill in the **Role Name** and optional **Description**
3. Save the role

[Clients](#) > [Client details](#) > [Create role](#)

Create role

Role name *

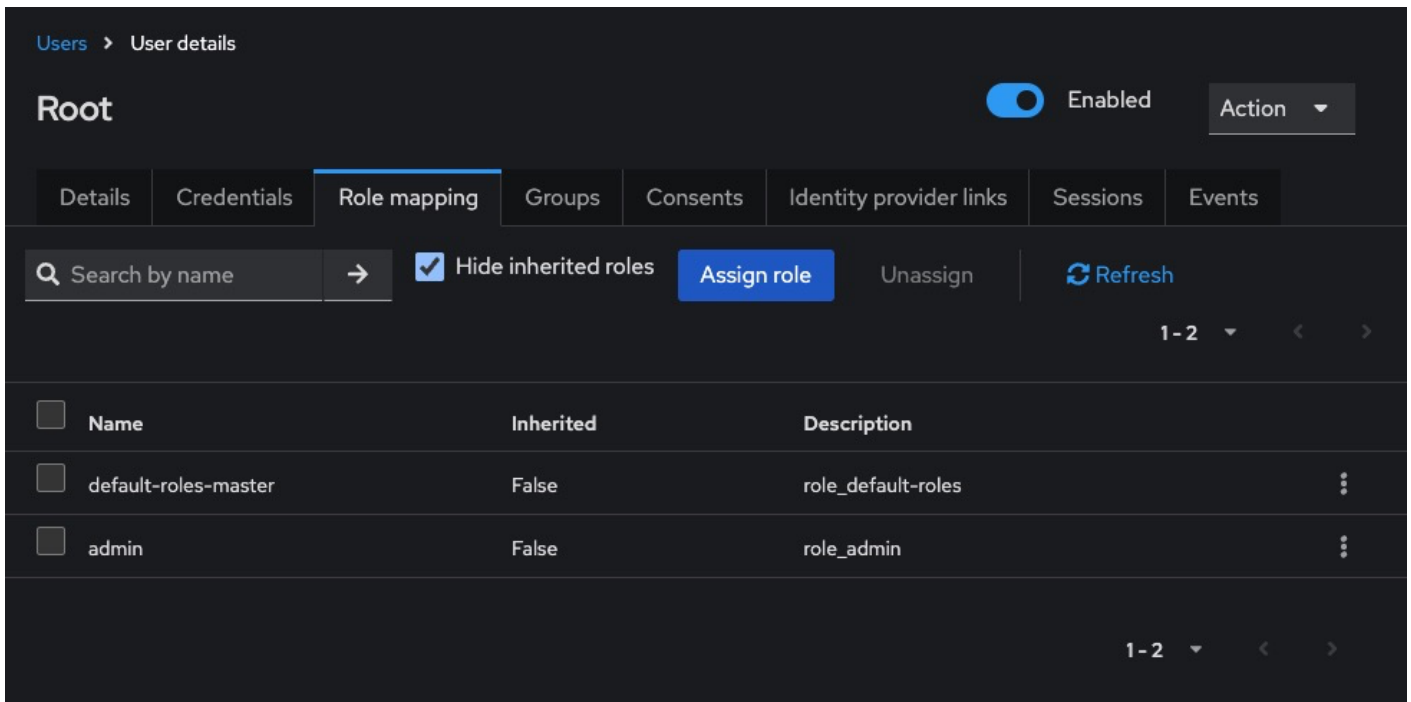
Description

[Save](#) [Cancel](#)

Assign Roles to Users

1. Go to **Users** > **[username]** > **Role Mappings**
2. In **Available Roles**, choose from:
 - Realm roles (top-left dropdown)
 - Client roles (select client under “Client Roles”)

3. Click **Add selected**



Users > User details

Root Enabled Action

Details Credentials **Role mapping** Groups Consents Identity provider links Sessions Events

Search by name → Hide inherited roles **Assign role** Unassign Refresh

<input type="checkbox"/>	Name	Inherited	Description	
<input type="checkbox"/>	default-roles-master	False	role_default-roles	⋮
<input type="checkbox"/>	admin	False	role_admin	⋮

1-2 < >

1-2 < >

Creating Roles via Keycloak REST API

Get Access Token

```
curl -X POST "https://<keycloak-domain>/realms/master/protocol/openid-connect/token" \  
-H "Content-Type: application/x-www-form-urlencoded" \  
-d "username=admin" \  
-d "password=admin-password" \  
-d "grant_type=password" \  
-d "client_id=admin-cli"
```

Save the access_token.

Create Realm Role

```
curl -X POST "https://<keycloak-domain>/admin/realms/<realm>/roles" \  
-H "Authorization: Bearer <access_token>" \  
-H "Content-Type: application/json" \  
-d '{
```

```
"name": "viewer",  
"description": "Read-only access"  
'
```

Create Client Role

```
curl -X POST "https://<keycloak-domain>/admin/realms/<realm>/clients/<client-id>/roles" \  
-H "Authorization: Bearer <access_token>" \  
-H "Content-Type: application/json" \  
-d '{  
  "name": "api-user",  
  "description": "API access for clients"  
'
```

To get the client ID:

```
curl -H "Authorization: Bearer <access_token>" \  
"https://<keycloak-domain>/admin/realms/<realm>/clients"
```

Creating Roles via Docker CLI

Access the Container

```
docker exec -it keycloak bash
```

Create Roles via CLI

```
/opt/keycloak/bin/kcadm.sh config credentials \  
--server http://localhost:8080 --realm master \  
--user admin --password admin  
  
/opt/keycloak/bin/kcadm.sh create roles -r <realm> \  
-s name=auditor -s description="Can view reports"
```

To create client roles:

```
/opt/keycloak/bin/kcadm.sh create clients/<client-id>/roles -r <realm> \  
-s name=external-api -s description="Role for external apps"
```

Required Permissions for Managing Roles

To manage roles, users need:

- manage-realm role for realm roles
- manage-clients role for client-specific roles

To assign via Admin Console:

Users > [admin-user] > Role Mappings > Realm Roles > Add 'manage-realm' or 'manage-clients'

Best Practices for Roles and Permissions

- **Use Fine-Grained Role Names:** Use names like `invoice_viewer`, `invoice_editor`, or `admin_dashboard` for clarity.
- **Use Groups to Assign Roles in Bulk:** Create groups such as `managers`, `sales`, or `auditors`, then assign roles to groups.
- **Map Roles to Access Tokens:** Use **Client > Mappers** to include role names in the `access_token` or `id_token`.
- **Prefer Client Roles for Application Permissions:** Client roles are scoped to individual apps and help separate responsibilities.
- **Use Composite Roles Sparingly:** Composite roles combine multiple roles into one but may add complexity if overused.

Common Issues and Troubleshooting

Issue	Possible Cause	Solution
Role doesn't appear in token	Missing protocol mapper	Add a role mapper in Client > Mappers

Issue	Possible Cause	Solution
User not authorized despite role assignment	Role not assigned to the correct client/realm	Verify if the role is client-scoped or realm-wide
403 Forbidden despite valid login	Role not embedded in access token	Ensure token includes required roles via protocol mappers
REST API: 409 Conflict when creating role	Role with same name already exists	Use a unique name or update existing role
Cannot assign role to user	User lacks manage-users privilege	Ensure admin has role assignment rights

Revision #1

Created 2025-06-17 12:02:48 UTC

Updated 2025-06-17 12:09:57 UTC