RBAC Deep Dive

EPISODE 5 -- KEYCLOAK SERIES -- 14 SCENES

SCENE 1 Login

Welcome back to the HelixNet Keycloak series.

Episode 5 -- RBAC Deep Dive.

Last episode we toured the admin console and saw all six realms.

Today we go inside the POS realm and look at how **role-based access control** actually works.

SCENE 2 Realm Roles List **MONEY SHOT**

Here are the **five custom roles** we built for the POS system.

Each role has an emoji icon and a clear description of what it does.

pos-admin -- full system control, the crown.

pos-auditor -- read-only access for compliance.

pos-cashier -- front-line, can process transactions but limited to **10% discounts**.

pos-developer -- testing role, can create products but no access to production data.

pos-manager -- the senior role, **unlimited discounts** and full reporting.

SCENE 3 Role Detail -- pos-admin

Let's look inside each role. The admin role -- full control over the POS realm and configuration.

Notice the tabs: Details, Attributes, Users in role.

SCENE 4 Role Detail -- pos-cashier

The cashier role. Key detail: **"Limited to 10% discount threshold."**

This is how you prevent staff from giving away the house.

SCENE 5 Role Detail -- pos-manager

The manager role. "Unlimited discounts and reporting."

When a customer needs more than 10% off, the manager steps in.

SCENE 6 Users in Role -- pos-cashier

Now click "Users in role" to see who has cashier access.

This is the other side of the coin -- **role to users mapping**.

SCENE 7 User Pam -- Cashier Only

Pam is a front-line cashier. One role: pos-cashier.

She can ring up sales and process checkouts. That's it.

Principle of least privilege in action.

SCENE 8 User Ralph -- Cashier + Manager

Ralph has two roles: cashier AND manager.

He works the register but can also approve discounts and run reports.

This is **role accumulation** -- each role adds capabilities.

SCENE 9 User Michael -- Developer Only

Michael is our developer. One role: pos-developer.

He can create test products but has no access to live transactions.

Complete separation of concerns.

SCENE 10 User Felix -- Super User **MONEY SHOT #2**

And Felix. Look at this -- **admin, cashier, AND manager**.

Three roles. Full system control.

In a real deployment, you'd have exactly one or two Felix-level users.

Everyone else gets the minimum they need.

SCENE 11 Client Scopes

Client scopes control what information goes into the JWT token.

Ten scopes here -- email, profile, roles, web-origins.

When a user logs in, these scopes determine what claims appear in their access token.

SCENE 12 Authentication Flows

Seven built-in authentication flows.

Browser flow for interactive logins, direct grant for API authentication.

Docker auth for container registries, first broker login for federated identity.

SCENE 13 Realm Settings

The realm settings page. This is **mission control**.

Realm ID: kc-pos-realm-dev. Frontend URL points to our Keycloak instance.

SSL required for external requests. OpenID and SAML endpoints at the bottom.

SCENE 14 Final Shot -- Roles List

Back to where we started. **Five roles, clean RBAC, production-ready.**

This is what separates a real platform from a toy project.

Next episode: we'll look at the client architecture -- how the three POS clients connect to Keycloak.