docker run -p 8080:8080 -e KEYCLOAK\_ADMIN=admin -e KEYCLOAK\_ADMIN\_PASSWORD=admin quay.io/keycloak/keycloak:latest start-dev

```

Then access: http://localhost:8080

Login with `admin` / `admin`

---

### Step 2: Initial Configuration (Your Use Case)

Once logged in, follow these steps:

#### \*\*A. Create a Realm\*\*

- Click "Master" dropdown (top left) → "Create Realm"

- Name it: `chatbot-realm`

- Click "Create"

#### \*\*B. Disable Self-Registration\*\*

- Go to: \*\*Realm Settings\*\* → \*\*Login\*\* tab

- Turn OFF: "User registration"

- Save

#### \*\*C. Create a Client (Your Chatbot App)\*\*

- Go to: \*\*Clients\*\* → "Create client"

- Settings:

- \*\*Client ID\*\*: `chatbot-client`

- \*\*Client authentication\*\*: ON (for confidential clients)

- \*\*Valid redirect URIs\*\*: `http://localhost:3000/\*` (adjust for your app)

- \*\*Web origins\*\*: `http://localhost:3000`

- Save and note the \*\*Client Secret\*\* (under Credentials tab)

#### \*\*D. Create a Single Role\*\*

- Go to: \*\*Realm roles\*\* → "Create role"

- Name: `chatbot-user`

- Save

#### \*\*E. Configure Default Role\*\*

- Go to: \*\*Realm settings\*\* → \*\*User registration\*\* tab

- Under "Default roles", add `chatbot-user`

---

### Step 3: Admin Operations You'll Need

#### \*\*Create a User (Admin Task)\*\*

1. Go to: \*\*Users\*\* → "Create new user"

2. Fill in:

- Username

- Email (optional)

- Email verified: ON

- Enabled: ON

3. Click "Create"

4. Go to \*\*Credentials\*\* tab → "Set password"

5. Go to \*\*Role mapping\*\* → Assign `chatbot-user` role

#### \*\*Reset Password\*\*

- Users → Select user → Credentials tab → "Reset password"

#### \*\*Delete User\*\*

- Users → Select user → "Delete" button

---

### Step 4: Understanding Keycloak for Your Chatbot Integration

\*\*Key Concepts:\*\*

1. \*\*Authentication Flow (What you'll implement):\*\*

```

User Login → Keycloak validates → Returns JWT Access Token →

Chatbot validates token on each request

----------------------------------------------------------------------**Keycloak Core Concepts (ELI5 Version)**

**1. What is Keycloak?**

Think of Keycloak as a **bouncer + ID card system** for your application:

* It checks who users are (authentication)
* It decides what they can do (authorization)
* Your app doesn't need to handle passwords or user databases

**2. Realm 🏰**

**Think of it like:** A completely separate "universe" or "building"

**Real-world analogy:**

* Company A has their employees
* Company B has their employees
* They don't mix - each company is a separate "realm"

**In Keycloak:**

* Each realm has its own:
  + Users
  + Roles
  + Clients (apps)
  + Settings
* Realms are **completely isolated** from each other

**Example:**

Master Realm (Keycloak's admin realm - don't use for your app)

├── Admin users who manage Keycloak itself

Chatbot Realm (your app's realm)

├── Your chatbot users

├── Your chatbot app configuration

└── Your roles and permissions

**Why separate realms?**

* You might have: production-realm, testing-realm, dev-realm
* Or: company-internal-realm, customer-realm

**3. Client 📱**

**Think of it like:** An app that wants to use Keycloak for login

**Real-world analogy:**

* Your office building (realm) allows different apps:
  + Mobile app
  + Web app
  + Desktop app
* Each is a different "client"

**In Keycloak:** A client represents your chatbot application

**Client Types:**

* **Public clients**: Mobile/browser apps (can't keep secrets secure)
* **Confidential clients**: Backend servers (can keep secrets secure)
  + Your chatbot will likely be this type

**What a client has:**

* **Client ID**: chatbot-client (like a username for your app)
* **Client Secret**: abc123xyz (like a password for your app)
* **Redirect URIs**: Where Keycloak sends users after login

**4. Users 👤**

**Think of it like:** People who will use your chatbot

**What they have:**

* Username
* Email
* Password (stored securely by Keycloak)
* Roles (what they can do)
* Attributes (extra info like phone number, department, etc.)

**5. Roles 🎭**

**Think of it like:** Job titles that define what you can do

**Real-world analogy:**

* "Manager" can approve expenses
* "Employee" can submit expenses
* "Admin" can do everything

**In your case:**

* You have ONE role: chatbot-user
* Everyone gets this role = everyone has same permissions
* (Simple! But you could add chatbot-admin later if needed)

**Two types:**

* **Realm roles**: Apply to entire realm (your case)
* **Client roles**: Apply only to specific clients

**6. Token 🎫**

**Think of it like:** A temporary pass that proves who you are

**Real-world analogy:**

* You show your ID at hotel check-in
* They give you a key card (token)
* You use the key card to access your room, gym, pool
* The key card expires after checkout

**In Keycloak:** After login, Keycloak gives your chatbot a **JWT token** that contains:

json

{

"username": "john\_doe",

"roles": ["chatbot-user"],

"expires": "2025-10-15T14:30:00Z"

}

```

Your chatbot checks this token instead of asking for password every time!

---

## 🔄 How It All Works Together

Let's trace a login:

```

1. User opens chatbot

↓

2. Chatbot redirects to Keycloak login page

↓

3. User enters username/password

↓

4. Keycloak checks credentials in the REALM

↓

5. If valid, Keycloak checks user's ROLES

↓

6. Keycloak creates a TOKEN with user info + roles

↓

7. Keycloak sends token back to CHATBOT (the client)

↓

8. Chatbot stores token

↓

9. For every chatbot message, chatbot checks:

- Is token still valid?

- Does user have "chatbot-user" role?

↓

10. If yes → allow chat

If no → reject

```

---

## 📊 Visual Structure

```

KEYCLOAK INSTANCE

│

├── MASTER REALM (built-in, for Keycloak admins)

│ └── Admin users

│

└── CHATBOT-REALM (your custom realm)

│

├── CLIENTS

│ └── chatbot-client (your app)

│ ├── Client ID: chatbot-client

│ ├── Client Secret: xyz123

│ └── Redirect URIs

│

├── ROLES

│ └── chatbot-user (everyone gets this)

│

└── USERS

├── user1@example.com

│ └── Has role: chatbot-user

├── user2@example.com

│ └── Has role: chatbot-user

└── admin@example.com

└── Has role: chatbot-user

**🎯 Quick Glossary for Your Use Case**

| **Term** | **What It Means for You** |
| --- | --- |
| **Realm** | Your chatbot's isolated environment |
| **Client** | Your chatbot application itself |
| **User** | People who will chat with your bot |
| **Role** | "chatbot-user" - everyone gets this |
| **Token** | Proof of login, contains username + role |
| **Admin** | You, managing users in Keycloak Console |

**🤔 Common Questions**

**Q: Why not just use a database with usernames/passwords?** A: You could, but then YOU have to:

* Hash passwords securely
* Handle password resets
* Implement "forgot password" emails
* Manage sessions
* Handle token expiration
* Implement role-based access Keycloak does all this for you!

**Q: Do users see Keycloak?** A: Usually yes (the login page), but it can be customized to look like your brand

**Q: Where are passwords stored?** A: In Keycloak's database (not your chatbot's database) - encrypted

**Q: What if Keycloak goes down?** A: Users can't login, but existing sessions (with valid tokens) can still work for a while

**✅ Ready for Next Steps?**

Now that you understand the concepts, you can:

1. **Install Keycloak** and see these concepts in the UI
2. **Create your realm** and understand why it's separate
3. **Create a client** and understand it represents your chatbot
4. **Create users/roles** and see how they fit together

Want me to walk you through the actual Keycloak UI now, or do you have questions about these concepts first?