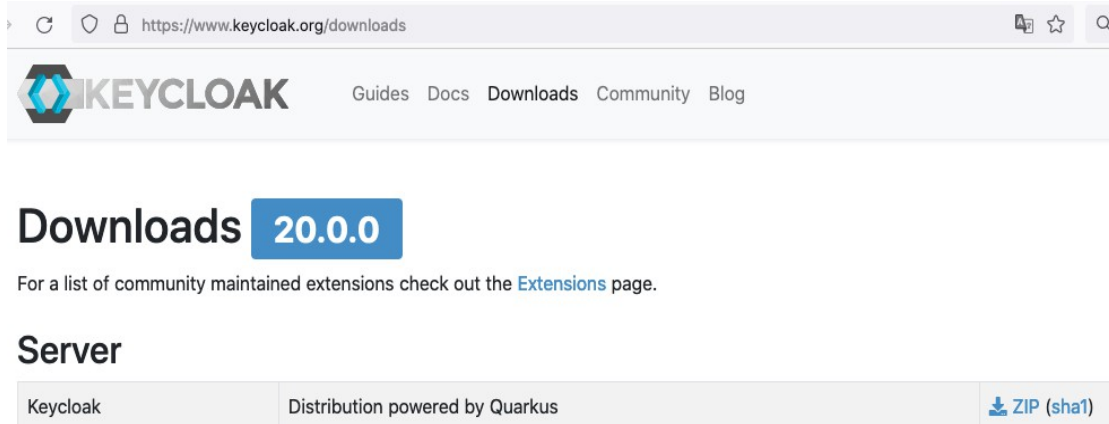


Inhaltsverzeichnis

1Setup Keycloak Server.....2

1 Setup Keycloak Server

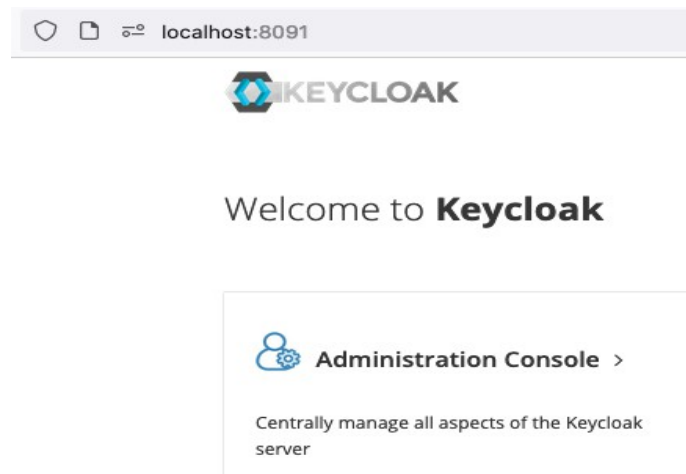
1. Download the latest keycloak server from <https://www.keycloak.org/downloads>.



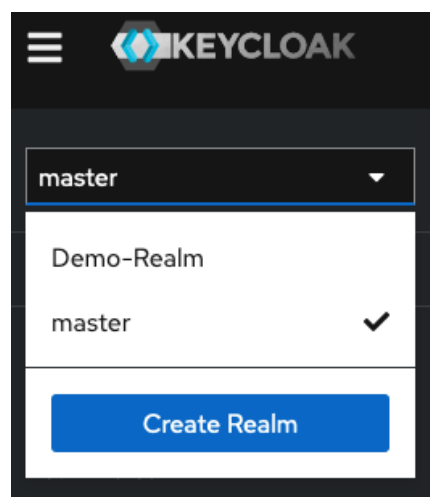
2. Unzip, open a terminal and change to directory keycloak-20.0.0 and run: `bin/kc.sh start-dev --http-port 8091`

```
keycloak-20.0.0 % bin/kc.sh start-dev --http-port 8091
```

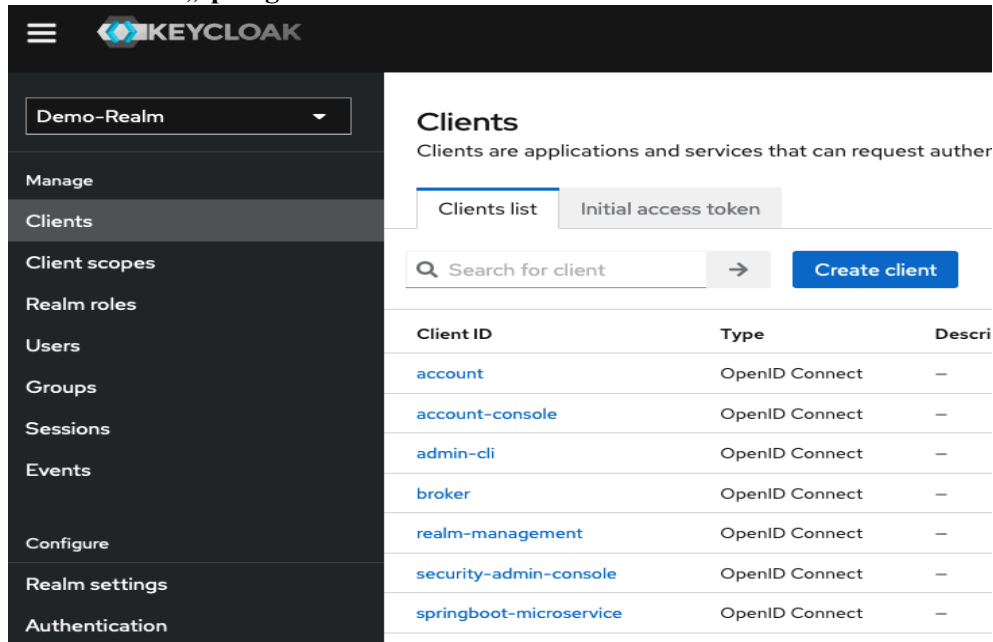
3. Open keycloak admin console



4. Have a look at <https://www.djamware.com/post/6225b66ba88c55c95abca0b6/spring-boot-security-postgresql-and-keycloak-rest-api-oauth2>
5. Create an admin account
6. Create a new Realm **Demo-Realm** and select it



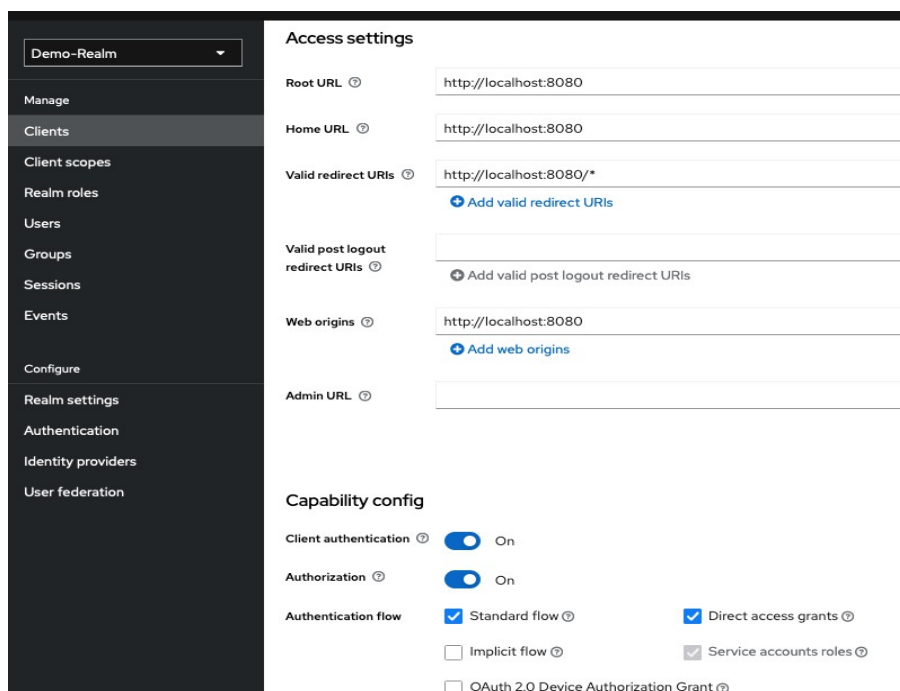
7. Create a new client „springboot-microservice“



The screenshot shows the Keycloak 'Clients' page for the 'Demo-Realm'. The left sidebar contains navigation links: Manage, Clients (selected), Client scopes, Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, and Authentication. The main content area is titled 'Clients' and includes a description: 'Clients are applications and services that can request authn'. There are two tabs: 'Clients list' (active) and 'Initial access token'. A search bar with a magnifying glass icon and a 'Create client' button are present. Below is a table of existing clients.

Client ID	Type	Descri
account	OpenID Connect	–
account-console	OpenID Connect	–
admin-cli	OpenID Connect	–
broker	OpenID Connect	–
realm-management	OpenID Connect	–
security-admin-console	OpenID Connect	–
springboot-microservice	OpenID Connect	–

8. Fill in Root-Url, Home-Url, Valid-Redirect-Urls and Web-Origins and switch on „Client authentication“ and „Authorization“



The screenshot shows the configuration page for the 'springboot-microservice' client in the 'Demo-Realm'. The left sidebar is the same as in the previous screenshot. The main content area is titled 'Access settings' and contains several input fields for configuration. Below this is the 'Capability config' section with various checkboxes and toggles.

Access settings

- Root URL:
- Home URL:
- Valid redirect URIs: [Add valid redirect URIs](#)
- Valid post logout redirect URIs: [Add valid post logout redirect URIs](#)
- Web origins: [Add web origins](#)
- Admin URL:

Capability config

- Client authentication: ☒ On
- Authorization: ☒ On
- Authentication flow: ☒ Standard flow ☒ Direct access grants ☐ Implicit flow ☒ Service accounts roles ☐ OAuth 2.0 Device Authorization Grant

9. Create two Realm-Roles **app-admin** and **app-user**

The screenshot shows the 'Realm roles' page in Keycloak. On the left is a dark sidebar with navigation links: Manage, Clients, Client scopes, Realm roles (selected), Users, Groups, Sessions, and Events. The main content area is titled 'Realm roles' with a subtitle 'Realm roles are the roles that you define for use in the current realm.' and a 'Learn more' link. Below this is a search bar 'Search role by name' with a right arrow and a blue 'Create role' button. A table lists the existing roles:

Role name	Composite
app-admin	True
app-user	True
default-roles-demo-realm ⓘ	True
offline_access	False
uma_authorization	False

10. Switch to client **springboot-microservice** and create two roles **admin** and **user**

The screenshot shows the 'Client details' page for the 'springboot-microservice' client. The left sidebar is the same as in the previous screenshot, with 'Clients' selected. The main content area is titled 'springboot-microservice' with an 'OpenID Connect' button. Below the title is a subtitle 'Clients are applications and services that can request authentication of a user.' and a set of tabs: Settings, Keys, Credentials, Roles (selected), Client scopes, Authorization, and Service accounts. There is a search bar 'Search role by name' with a right arrow and a blue 'Create role' button. A table lists the roles for this client:

Role name	Composite
admin	False
uma_protection	False
user	False

11. Switch to Realm-Roles and select Role **app-user**, click on Action and select „Add associated roles“

The screenshot shows the 'Role details' page for the 'app-user' role. The left sidebar is the same as in the previous screenshots, with 'Realm roles' selected. The main content area is titled 'app-user' with a 'Composite' button. Below the title are tabs: Details (selected), Associated roles, Attributes, Users in role, and Permissions. An 'Action' dropdown menu is open, showing 'Add associated roles' and 'Delete this role'. The 'Role name' field is filled with 'app-user'. The 'Description' field is empty. At the bottom are 'Save' and 'Revert' buttons.

12. Select „filter by clients“ and enter „springboot-microservices“ and select role **user**

Assign roles to app-user account

Filter by clients

Q spring

×

→

1-3

<

>

<input type="checkbox"/>	Name	Description
<input type="checkbox"/>	springboot-microservice admin	
<input type="checkbox"/>	springboot-microservice uma_protection	
<input checked="" type="checkbox"/>	springboot-microservice user	

1-3

<

>

Assign Cancel

13. Realm-Role app-user is now a composite role

[Realm roles](#) > Role details

app-user Composite

Details

Associated roles

Attributes

Users in role

Permissions

Q Search by name

→

☒ Hide inherited roles

Assign role

Unassign

<input type="checkbox"/>	Name	Inherited	Description
<input type="checkbox"/>	springboot-microservice user	False	—

14. Do the same with Realm-Role app-admin and associate role admin

[Realm roles](#) > Role details

app-admin Composite

Details

Associated roles

Attributes

Users in role

Permissions

Q Search by name

→

☒ Hide inherited roles

Assign role

Unassign

<input type="checkbox"/>	Name	Inherited
<input type="checkbox"/>	springboot-microservice admin	False

15. Create 3 Users named employee1 – employee3, set „Email verified“ to „On“

[Users](#) > Create user

Create user

Username *	<input type="text" value="employee1"/>
Email	<input type="text" value="m0.m0@company.com"/>
Email verified ?	<input checked="" type="checkbox"/> On
First name	<input type="text" value="Max0"/>
Last name	<input type="text" value="Mustermann0"/>
Required user actions ?	<input type="text" value="Select action"/>
Groups ?	Join Groups
<div>Create Cancel</div>	

16. Set user credentials

[Users](#) > User details

employee1

Details	Attributes	Credentials	Role mapping	Groups
?	Type	User label		
	Password	My password		

17. Assign Realm-Role **app-user** to employee1

employee1

Details	Attributes	Credentials	Role mapping	Groups	Consents	Identity provide
<input type="text" value="Search by name"/>		→	<input checked="" type="checkbox"/> Hide inherited roles	Assign role	Unassign	
<input type="checkbox"/>	Name		Inherited	Description		
<input type="checkbox"/>	app-user		False	–		
<input type="checkbox"/>	default-roles-demo-realm		False	\${role_default-roles}		

18. Remove all actions in field „Required user action“ and save

19. Repeat everything from point 14-17 with employee2 and assign Realm-Role app-admin

employee2

Details	Attributes	Credentials	Role mapping	Groups	Cor
<input type="text" value="Search by name"/>		→	<input checked="" type="checkbox"/> Hide inherited roles	Assign role	
<input type="checkbox"/>	Name		Inherited		
<input type="checkbox"/>	default-roles-demo-realm		False		
<input type="checkbox"/>	app-admin		False		

20. Repeat everything from point 14-17 with employee3 and assign Realm-Role **app-admin** and **app-user**

[Users](#) > User details

employee3

Details	Attributes	Credentials	Role mapping	Groups	Conse
---------	------------	-------------	--------------	--------	-------

→ ☒ Hide inherited roles Assign role

<input type="checkbox"/> Name	Inherited
<input type="checkbox"/> app-user	False
<input type="checkbox"/> default-roles-demo-realm	False
<input type="checkbox"/> app-admin	False

21. Switch to client scopes and add new scope read

Client scopes

Client scopes are a common set of protocol mappers and roles that are shared between i

× → Create client scope

<input type="checkbox"/> Name	Assigned type	Protocol
<input type="checkbox"/> read	None ▼	OpenID Conn

22. Assign Realm-Roles app-admin, app-user and Roles user and admin to scope read

[Client scopes](#) > Client scope details

read openid-connect

Settings	Mappers	Scope
----------	---------	-------

→ ☒ Hide inherited roles Assign role

<input type="checkbox"/> Name	Inherited
<input type="checkbox"/> app-user	False
<input type="checkbox"/> app-admin	False
<input type="checkbox"/> springboot-microservice admin	False
<input type="checkbox"/> springboot-microservice user	False

23. Switch to client „springboot-microservices“ and add scope read as „Optional“

springboot-microservice OpenID Connect

Clients are applications and services that can request authentication of a user.

Settings	Keys	Credentials	Roles	Client scopes	Authorization	Service												
<div> <div>Setup</div> <div>Evaluate</div> </div>																		
<div> <div> <div>🔍 Name</div> <div>Q read</div> <div>×</div> <div>→</div> </div> <div>Add client scope</div> </div>																		
<table border="1"> <thead> <tr> <th><input type="checkbox"/></th> <th>Assigned client scope</th> <th>Assigned type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>springboot-microservice-dedicated</td> <td>none</td> <td>Dedicated</td> </tr> <tr> <td><input type="checkbox"/></td> <td>read</td> <td>Optional</td> <td></td> </tr> </tbody> </table>							<input type="checkbox"/>	Assigned client scope	Assigned type	Description	<input type="checkbox"/>	springboot-microservice-dedicated	none	Dedicated	<input type="checkbox"/>	read	Optional	
<input type="checkbox"/>	Assigned client scope	Assigned type	Description															
<input type="checkbox"/>	springboot-microservice-dedicated	none	Dedicated															
<input type="checkbox"/>	read	Optional																

24. Switch to Realm settings and open „OpenID Endpoint Configuration“

Configure

Realm settings

Authentication

Identity providers

User federation

Endpoints ?

OpenID Endpoint Configuration [↗](#)

SAML 2.0 Identity Provider Metadata [↗](#)

Save

Revert

25. <http://localhost:8091/realms/Demo-Realm/.well-known/openid-configuration>

← → ↺ 📄

localhost:8091/realms/Demo-Realm/.well-known/openid-configuration

JSON Rohdaten Kopfzeilen

Speichern Kopieren Alle einklappen Alle ausklappen 🔍 JSON durchsuchen

```

{
  "issuer": "http://localhost:8091/realms/Demo-Realm",
  "authorization_endpoint": "http://localhost:8091/realms/Demo-Realm/protocol/openid-connect/auth",
  "token_endpoint": "http://localhost:8091/realms/Demo-Realm/protocol/openid-connect/token",
  "introspection_endpoint": "http://localhost:8091/realms/Demo-Realm/protocol/openid-connect/token/introspect",
  "userinfo_endpoint": "http://localhost:8091/realms/Demo-Realm/protocol/openid-connect/userinfo",
  "end_session_endpoint": "http://localhost:8091/realms/Demo-Realm/protocol/openid-connect/logout",
  "frontchannel_logout_session_supported": true,
  "frontchannel_logout_supported": true,
  "jwks_uri": "http://localhost:8091/realms/Demo-Realm/protocol/openid-connect/certs",
  "check_session_iframe": "http://localhost:8091/realms/Demo-Realm/protocol/openid-connect/login-status-iframe.html",
  "grant_types_supported": [
    "authorization_code",
    "implicit",
    "refresh_token",
    "password",
    "client_credentials",
    "urn:ietf:params:oauth:grant-type:device_code",
    "urn:openid:params:grant-type:ciba"
  ],
  "acr_values_supported": [
    "0"
  ]
}

```

26.