

KEYCLOAK INTEGRATION WITH SPRING BOOT

Install Keycloak Using Docker

To install Keycloak using Docker, you can use the following command:

```
docker run -p 8080:8080 -e KEYCLOAK_ADMIN=admin -e  
KEYCLOAK_ADMIN_PASSWORD=admin quay.io/keycloak/keycloak:24.0.4 start-dev
```

This command will run Keycloak on the default port 8080.

Changing the Running Port

```
docker run -p 8095:8080 -e KEYCLOAK_ADMIN=admin -e  
KEYCLOAK_ADMIN_PASSWORD=admin quay.io/keycloak/keycloak:24.0.4 start-dev
```

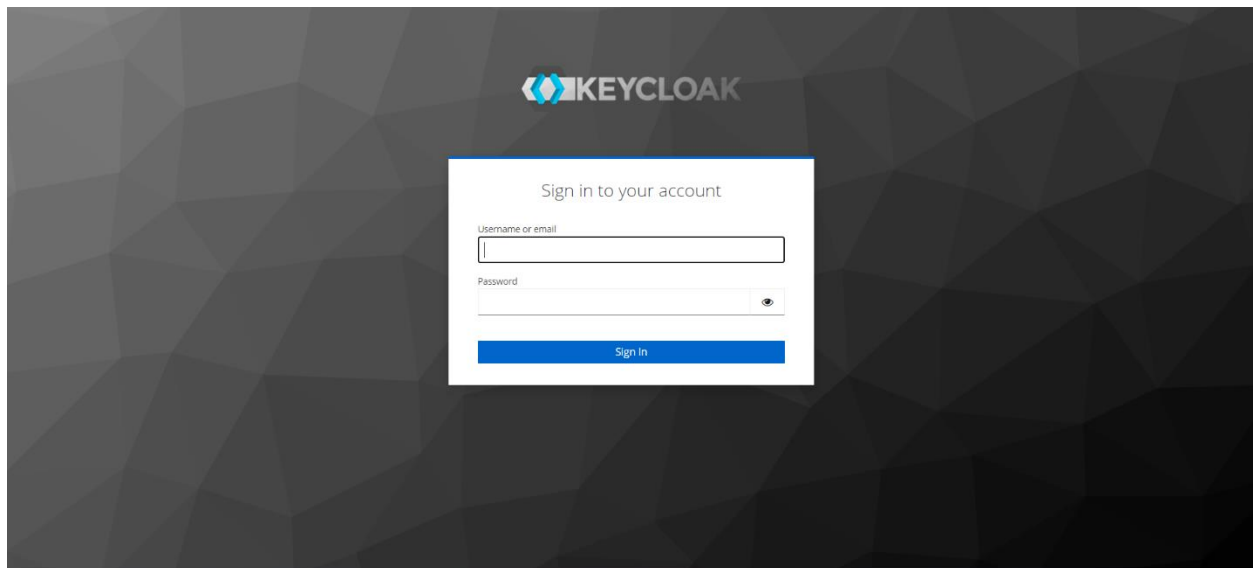
In this command, 8095:8080 maps port 8095 on your host machine to port 8080 in the Keycloak container.

If that port is currently used for other task change the port and try again

Sign in to Keycloak

After running this command, open any browser and go to: <http://localhost:8095>

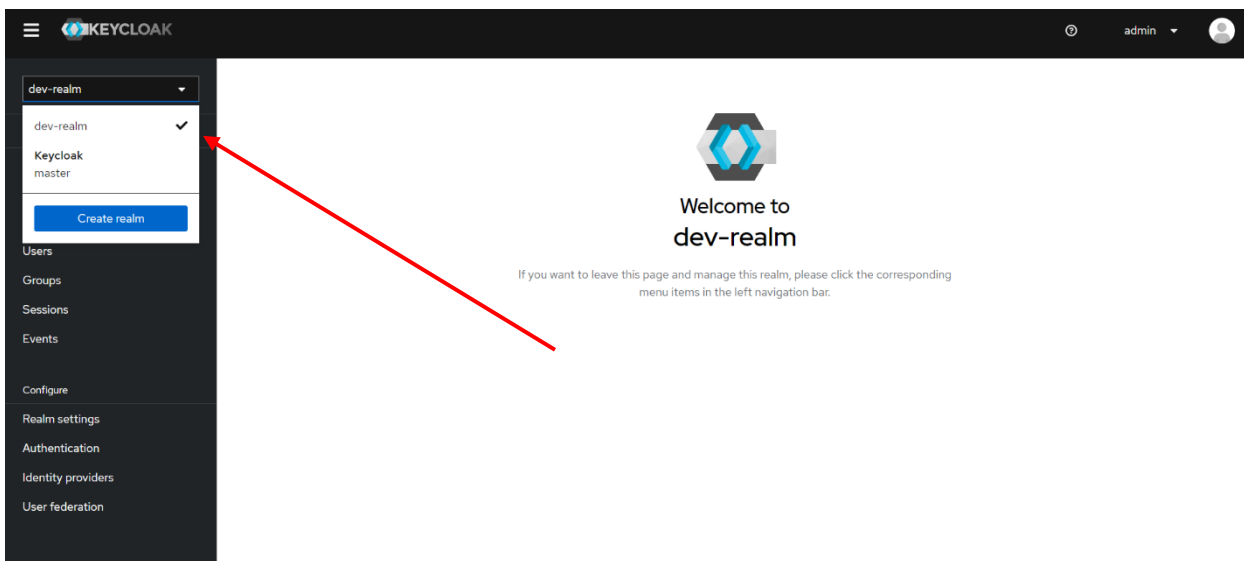
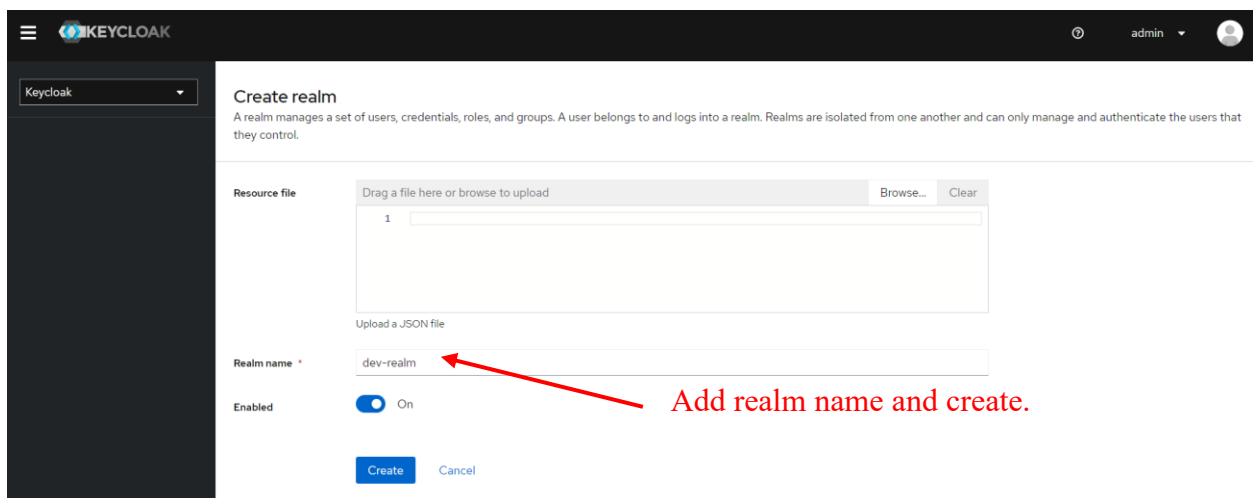
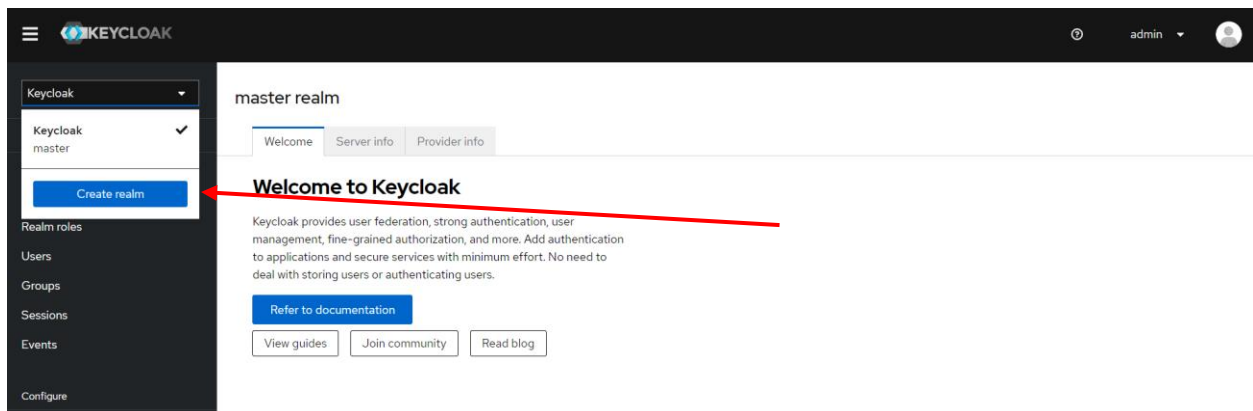
It will load below ui



username: admin

password: admin

Create Realm



Create Client

The top screenshot shows the Keycloak admin console with the 'Clients' page. The 'Clients list' tab is active, showing a table of existing clients. A red arrow points to the 'Create client' button. The bottom screenshot shows the same page with a red arrow pointing to the 'Create client' button.

Client ID	Name	Type	Description	Home URL
account	\${client_account}	OpenID Connect	—	http://localhost:8095/realms/dev-realm/account/
account-console	\${client_account-console}	OpenID Connect	—	http://localhost:8095/realms/dev-realm/account/

here you can add any client id and name and click next.

The screenshot shows the 'Create client' form in the Keycloak admin console. The 'Client type' is set to 'OpenID Connect'. The 'Client ID' field contains 'authenticationClients' and the 'Name' field contains 'spring boot application'. Red arrows point to these fields from the text above.

1 General settings
2 Capability config
3 Login settings

Client type: OpenID Connect
Client ID: authenticationClients
Name: spring boot application
Description:
Always display in UI: Off

KEYCLOAK

admin

dev-realm

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

Identity providers

User federation

Clients > Create client

Create client

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

1 General settings

2 Capability config

3 Login settings

Client authentication

Off

Authorization

Off

Authentication flow

☒ Standard flow

☐ Implicit flow

☐ OAuth 2.0 Device Authorization Grant

☐ OIDC CIBA Grant

☒ Direct access grants

☐ Service accounts roles

dev-realm

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

Identity providers

User federation

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

1 General settings

2 Capability config

3 Login 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

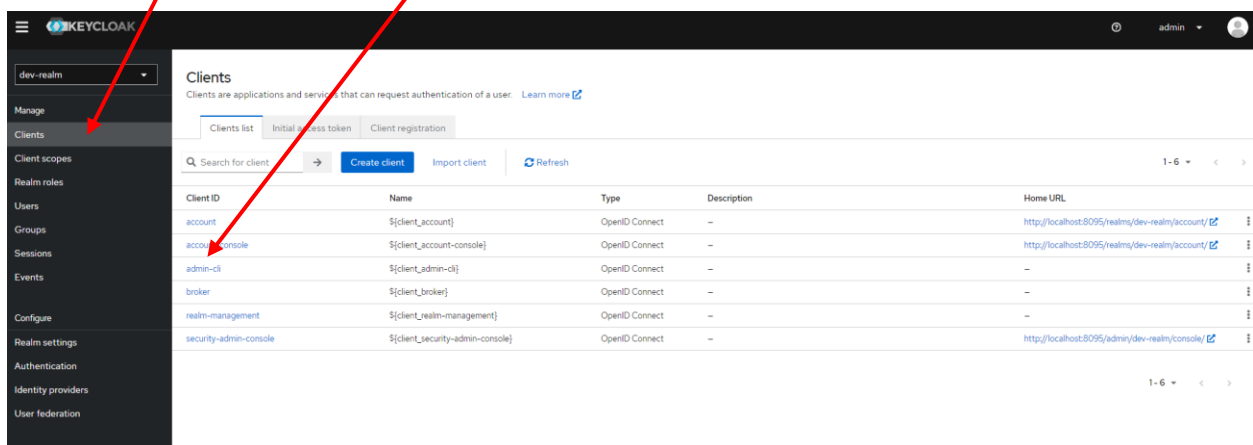
Save

Back

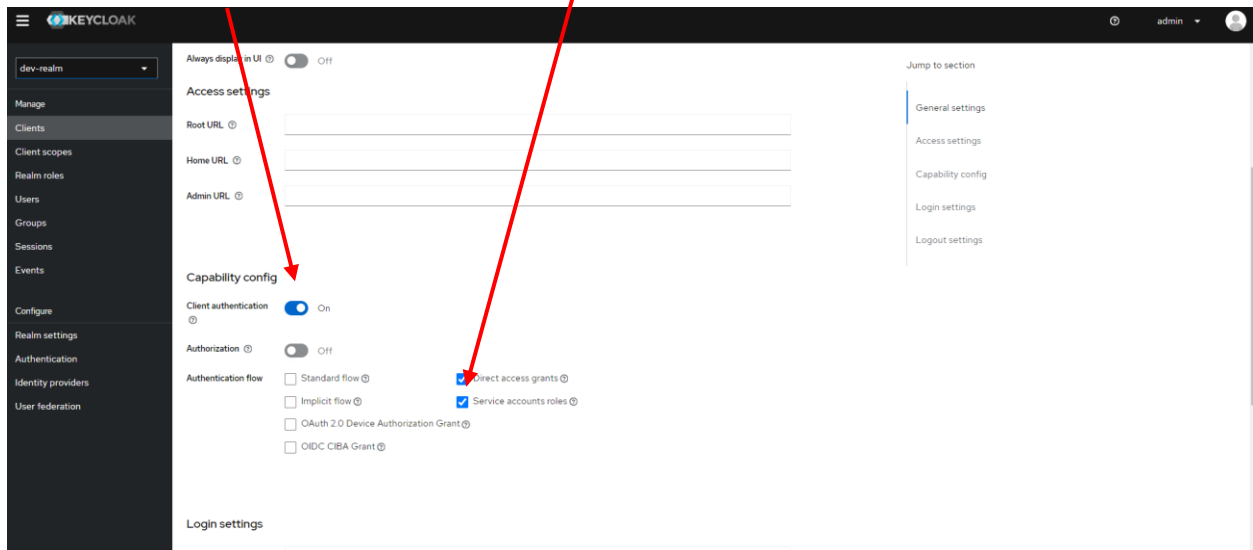
Cancel

Give permission to the admin-cli

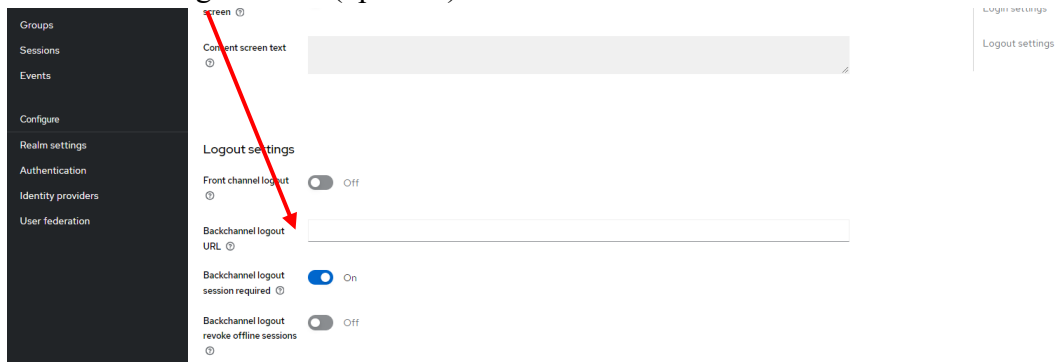
Go to the clients and select admin-cli



Enable client-authentication and service account roles



And add the logout URL (optional). Then click the Save button at the bottom of the page.



Before save

Clients > Client details

admin-cli (OpenID Connect) Enabled Action

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

Settings Roles Client scopes Sessions Advanced

General settings

Client ID *

Name

Description

Always display in UI ☐ Off

Jump to section

- General settings
- Access settings
- Capability config
- Login settings

After save

Clients > Client details

admin-cli (OpenID Connect) Enabled Action

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

Settings **Keys** Credentials Roles Client scopes Service accounts roles Sessions Advanced

General settings

Client ID *

Name

Description

Jump to section

- General settings
- Access settings
- Capability config

To create/update user using rest api we should assign new role realm-management manage - users.

Steps:

Clients > Client details

admin-cli (OpenID Connect) Enabled Action

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

Settings Keys Credentials Roles Client scopes **Service accounts roles** Sessions Advanced

To manage detail and group mappings, click on the username service-account-administrator

Search by name ☒ Hide inherited roles Assign role Unassign Refresh 1-1 < >

<input type="checkbox"/> Name	Inherited	Description
<input type="checkbox"/> default-roles-dev-realm	False	`\${role_default-roles}`

Assign roles to admin-cli

Filter by realm roles Search by role name Refresh 1-2 < >

Filter by clients

	Description
<input type="checkbox"/> offline_access	`\${role_offline-access}`
<input type="checkbox"/> uma_authorization	`\${role_uma_authorization}`

Assign Cancel

Assign roles to admin-cli

Filter by clients

Search by role name

→

Refresh

11 - 20

<

>

<input type="checkbox"/>	Name	Description
<input type="checkbox"/>	realm-management impersonation	\${role_impersonation}
<input type="checkbox"/>	realm-management manage-authorization	\${role_manage-authorization}
<input type="checkbox"/>	realm-management manage-clients	\${role_manage-clients}
<input type="checkbox"/>	realm-management manage-events	\${role_manage-events}
<input type="checkbox"/>	realm-management manage-identity-providers	\${role_manage-identity-providers}
<input type="checkbox"/>	realm-management manage-realm	\${role_manage-realm}
<input checked="" type="checkbox"/>	realm-management manage-users	\${role_manage-users}
<input type="checkbox"/>	realm-management query-clients	\${role_query-clients}
<input type="checkbox"/>	realm-management query-groups	\${role_query-groups}
<input type="checkbox"/>	realm-management query-realms	\${role_query-realms}

11 - 20 < >

Assign

Cancel

Check this role and click assign button.

User-management Spring boot application

Git Repository: <https://github.com/savindu29/keycloak-user-management.git>

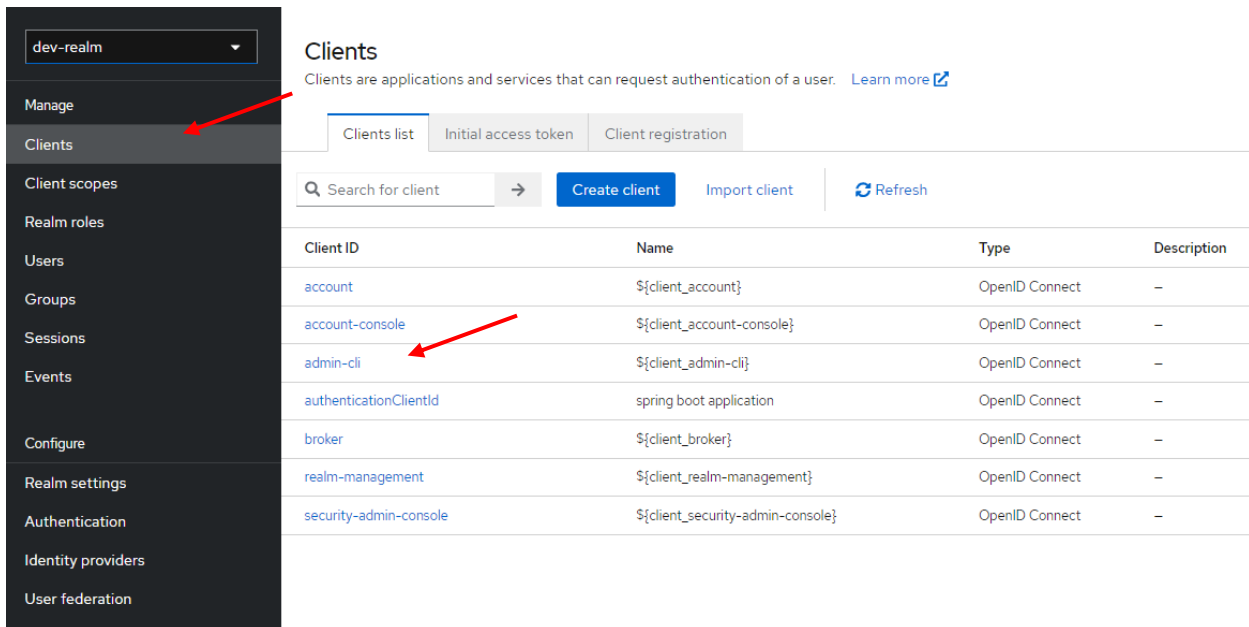
Clone the repository.

In the application.properties file setup those properties matching to your details .change realm , domain, client as needed

```
keycloak.realm=dev-realm
keycloak.domain=localhost:8095
keycloak.adminClientId=admin-cli
keycloak.adminClientSecret=CLJKMU03G0sPi04V2Jh4PWyaCVAWQzga
keycloak.client=authenticationClientId
keycloak.urls.auth=http://${keycloak.domain}
```



To get this step in below



Clients
Clients are applications and services that can request authentication of a user. [Learn more](#)

dev-realm

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

Identity providers

User federation

Clients list Initial access token Client registration

Search for client → Create client Import client Refresh

Client ID	Name	Type	Description
account	<code>\${client_account}</code>	OpenID Connect	–
account-console	<code>\${client_account-console}</code>	OpenID Connect	–
admin-cli	<code>\${client_admin-cli}</code>	OpenID Connect	–
authenticationClientId	spring boot application	OpenID Connect	–
broker	<code>\${client_broker}</code>	OpenID Connect	–
realm-management	<code>\${client_realm-management}</code>	OpenID Connect	–
security-admin-console	<code>\${client_security-admin-console}</code>	OpenID Connect	–

Go to the clients then go the admin-cli

dev-realm

Manage

- Clients
- Client scopes
- Realm roles
- Users
- Groups
- Sessions
- Events
- Configure
- Realm settings
- Authentication
- Identity providers
- User federation

Clients > Client details

admin-cli OpenID Connect

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

Settings Keys **Credentials** Roles Client scopes Service accounts roles Sessions Advanced

Client Authenticator Client Id and Secret

Save

Client Secret CLJKMU03G0sPi04V2Jh4PWyaCVAWQzga Regenerate

Registration access token Regenerate

Go to the class containing the main method and edit that port and realm name

```

savindu.panagoda
@SpringBootApplication(scanBasePackages = {"com.savindu.*", "org.springframework"})
@SecurityScheme(
    name = "keycloak",
    openIdConnectUrl = "http://localhost:8095/realms/dev-realm/.well-known/openid-configuration",
    scheme = "bearer",
    type = SecuritySchemeType.HTTP,
    in = SecuritySchemeIn.HEADER,
    bearerFormat = "JWT"
)

public class UserManagementApplication {

    savindu.panagoda
    public static void main(String[] args) { SpringApplication.run(UserManagementApplication.class, args); }
  
```

Set up email verification.

Go to the realm settings and email tab.

dev-realm

Realm settings are settings that control the options for users, applications, roles, and groups in the current realm. [Learn more](#)

General Login **Email** Themes Keys Events Localization Security defenses Sessions Tokens Client policies User profile User registration

Template

From * Sender email address

From display name ⓘ Display name for Sender email address

Reply to Reply to email address

Reply to display name ⓘ Display name for "reply to" email address

Envelope from ⓘ Sender envelope email address

In the bottom of the page click on this

Connection & Authentication

Host * SMTP host

Port SMTP port (defaults to 25)

Encryption ☐ Enable SSL ☐ Enable StartTLS

Authentication ☒ Disabled

⚠ To test the connection you must first configure an e-mail address for the current user (admin). [Configure e-mail address](#)

It will navigate to the admin user details page

KEYCLOAK

Users > User details

Admin

Details Credentials Role mapping Groups Consents Identity provider links Sessions

ID * 2f1f78af-a78c-4922-a6c3-77094c0f4834

Created at * 5/28/2024, 12:34:45 PM

Required user actions ⓘ Select action

Email verified ⓘ ☒ No

General

Username * admin

Email

First name

Last name

Save Revert

Add your default email here and save

Fill the details here. I'm use gmail and

resource : <https://youtu.be/kTcmbZqNiGw?si=7eMBSvVf-StJ-3MS>

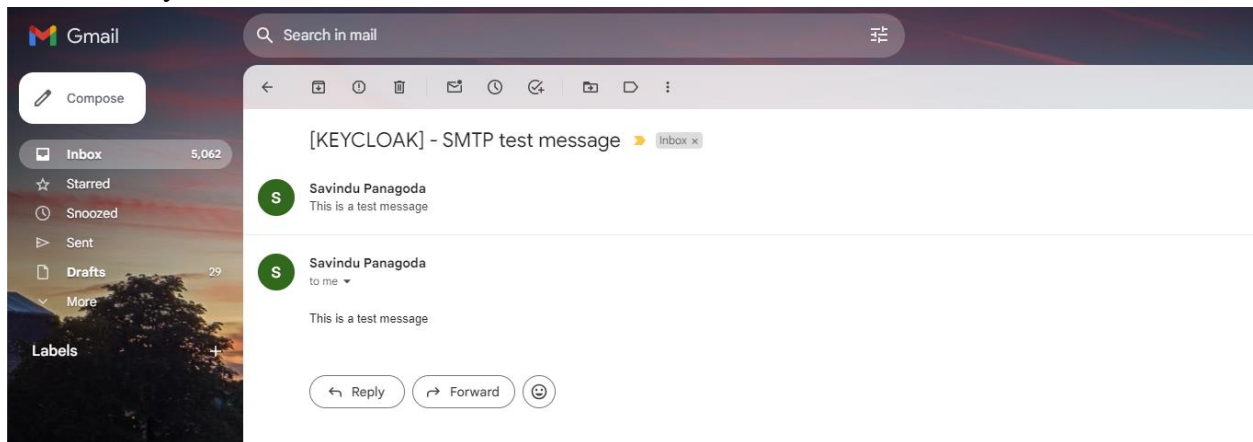
The fully qualified domain name of the SMTP service is smtp.gmail.com. Configuration options include: Port 25, 465, or 587. SSL and TLS protocols.

The screenshot shows the Keycloak administration console with the 'SMTP' configuration page selected under 'Realms' > 'dev-realm' > 'Configure' > 'SMTP'. The page is divided into two main sections: 'From' and 'Connection & Authentication'. The 'From' section includes fields for 'From' (savindu329@gmail.com), 'From display name' (Savindu Panagoda), 'Reply to' (Reply to email address), 'Reply to display name' (Display name for "reply to" email address), and 'Envelope from' (Sender envelope email address). The 'Connection & Authentication' section includes fields for 'Host' (smtp.gmail.com), 'Port' (465), 'Encryption' (checked for 'Enable SSL' and 'Enable StartTLS'), 'Authentication' (checked for 'Enabled'), 'Username' (savindu329@gmail.com), and 'Password' (masked with dots). Red arrows point to the 'From' field, the 'Host' field, the 'Username' field, and the 'Password' field. Text annotations are present: 'From which email address you want' points to the 'From' field; 'Same as the which email address you want to send.' points to the 'Username' field; and 'Not the email password, refer to the resources provided and then you can find the password should add here.' points to the 'Password' field.

Click on the test connection here. And if you setup it correctly it will show like this

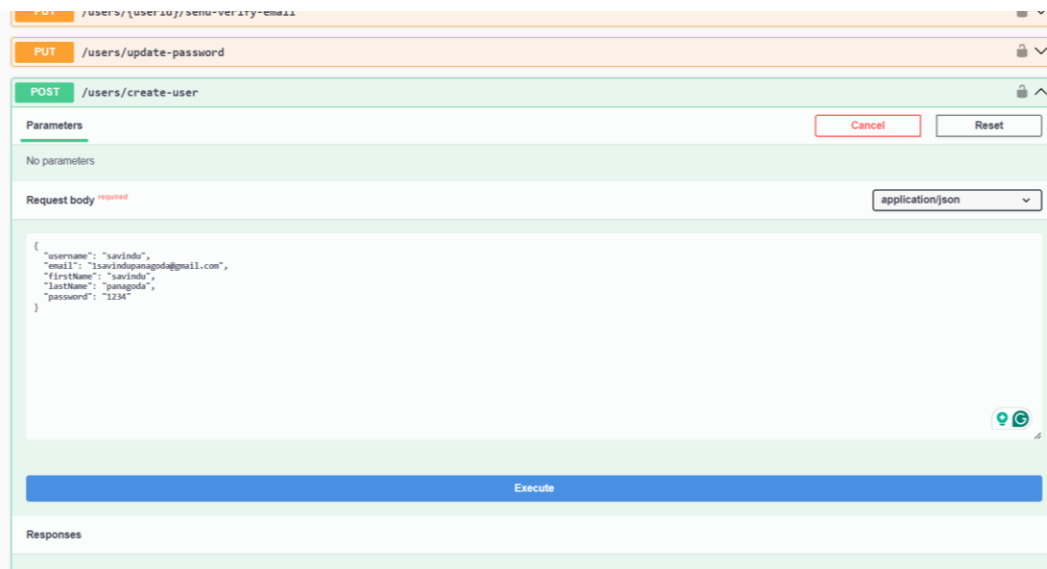


Then check your default email it will receive test email like below



Then run the spring boot app and open the swagger: <http://localhost:8070/swagger-ui/index.html>

Create user : fill the details and execute



After you execute this there will receive you a verification email. When click the received link you can verify the email

path : <http://localhost:8095/realms/dev-realm/protocol/openid-connect/token>

Body: add `client_secret`, `client_id`, and `grant_type` and use `username` and `password` which is you given when creating the user

POST

▼

http://localhost:8095/realms/dev-realm/protocol/openid-connect/token

Send

▼

Params

Authorization

Headers (8)

Body

Scripts

Tests

Settings

Cookies

☐ none
 ☐ form-data
 ☒ x-www-form-urlencoded
 ☐ raw
 ☐ binary
 ☐ GraphQL

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	client_secret	CLJKMU03G0sPi04V2Jh4PWyaCVAWQZga			
<input checked="" type="checkbox"/>	client_id	admin-cli			
<input checked="" type="checkbox"/>	username	savindu			
<input checked="" type="checkbox"/>	password	1234			
<input checked="" type="checkbox"/>	grant_type	password			
	Key	Value	Description		

Body

Cookies

Headers (9)

Test Results

ⓘ

Status: 200 OK

Time: 393 ms

Size: 2.27 KB

Save as example

...

[illegible]

13

It is public controller. Using your username you can reset your password. After execute you will receive reset password via email

PUT

/public/{username}/forgot-password

Parameters

Cancel

Name	Description
username * required	
string	savindu
(path)	

Execute

Clear

Responses

Curl

curl -X 'PUT' \
"http://localhost:8070/public/savindu/forgot-password" \
-H 'accept: */*'

Request URL

http://localhost:8070/public/savindu/forgot-password

Server response

Code	Details
200	<div>Response headers</div> <div>connection: keep-alive content-length: 0 date: Tue, 28 May 2024 11:37:59 GMT keep-alive: timeout=60</div>

Responses

A screenshot of an email interface. At the top, the subject is "Update Your Account" with a yellow envelope icon and a grey "Inbox x" button. The sender is "Savindu Panagoda" with a green circular profile picture containing a white 'S'. The recipient is "to me" with a dropdown arrow. The time is "5:07 PM (1 minute ago)" and there are icons for star, smiley face, and reply. The main text says: "Your administrator has just requested that you update your Dev-realm account by performing the following action(s): Update Password. Click on the link below to start this process." Below this is a blue hyperlink "Link to account update" which is pointed to by a red arrow. Underneath the link, it says "This link will expire within 12 hours." and "If you are unaware that your administrator has requested this, just ignore this message and nothing will be changed." At the bottom are buttons for "Reply", "Forward", and a smiley face icon.