

Keycloak SPI

@Osama Harbe @German Shein @Mohammad Saqlaine @Ilya Bykov @Muhammad Nasar

SPI Functionality:

- Phone support like e-mail
- One Time Password (OTP) by phone | email
- Login by phone | email
- Register with phone | email
- Authentication by phone | email

Phone registration support

Two user attributes are going to be used by this provider: `phoneNumberVerified` (bool) and `phoneNumber|email` (str). Multiple users can have the same `phoneNumber|email`, but only one of them will have `phoneNumberVerified = true` at the end of a verification process. This accommodates the use case of pre-paid numbers that get recycled if inactive for too much time.

Under **Authentication > Flows**:

1. Enable the **Internationalization** option with the support of **Arabic** language under **Realm Settings > Localization**

OCC

Realm settings are settings that control the options for users, applications, roles, and groups

General

Login

Email

Themes

Keys

Events

Localization

Se

Internationalization ⓘ

☒ Enabled

Supported locales

العربية ✕

English ✕

Select locales

Default locale

English

Save

Revert

2. Enable the **User registration** option under **Realm Settings > Login**

OCC

Realm settings are settings that control the opti

General

Login

Email

Themes

Login screen customization

User registration ⓘ

☒ On

Forgot password ⓘ

☐ Off

Remember me ⓘ

☐ Off

3. Under **Realm Settings > Themes** Set **Login Theme** to **phone**

OCC

Realm settings are settings that control the options for users, applic

General

Login

Email

Themes

Keys

Events

Login theme ⓘ

phone

Account theme ⓘ

Select a theme

Admin theme ⓘ

Select a theme

Email theme ⓘ

Select a theme

Save

Revert

4. Duplicate the `Registration` flow to `Registration with phone | email` flow through the menu button on the right of the `registration` flow
5. Bind the newly created `Registration with phone | email` flow by clicking on actions on the top right
6. Replace `Registration User Profile Creation` by deleting it and adding a step named `Registration Phone User Creation`
7. Once `Registration Phone User Creation` step is added, move it to the top
8. Set `Registration Phone User Creation` step requirement to `Required`
9. Click on settings for `Registration Phone User Creation` to disable input email and input name

Registration Phone User Creation config x

Alias ⓘ

Phone number as username ⓘ

☒ On

Input name ⓘ

☐ Off

Input Email ⓘ

☒ Off

Save

Cancel

10. Enable phone verification, click on `Registration with phone registration Form` > Add `Phone validation` if you want to verify phone.
11. (Optional) Read query parameter add to user attribute:
Click on `Registration with phone registration Form` > Actions > Add execution on the `Query Parameter Reader` line
Click on `Registration with phone registration Form` > Actions > `configure` add accept param name in to
12. (Optional) Hidden password field:
Delete or disable `Password Validation`.
13. (Optional) if not any user profile:
Delete or disable `Profile Validation`

Tip: If Realm parameter `Email as username` is true, then config `Phone number as username` and `hide email` is invalid!
If parameter `duplicate-phone` is true then `Phone number as username` is invalid!

Steps	Requirement
Registration with phone registration form registration form	Required
Registration Phone User Creation	Required
Profile Validation	Required
Password Validation	Required
Phone validation	Required
Recaptcha	Disabled

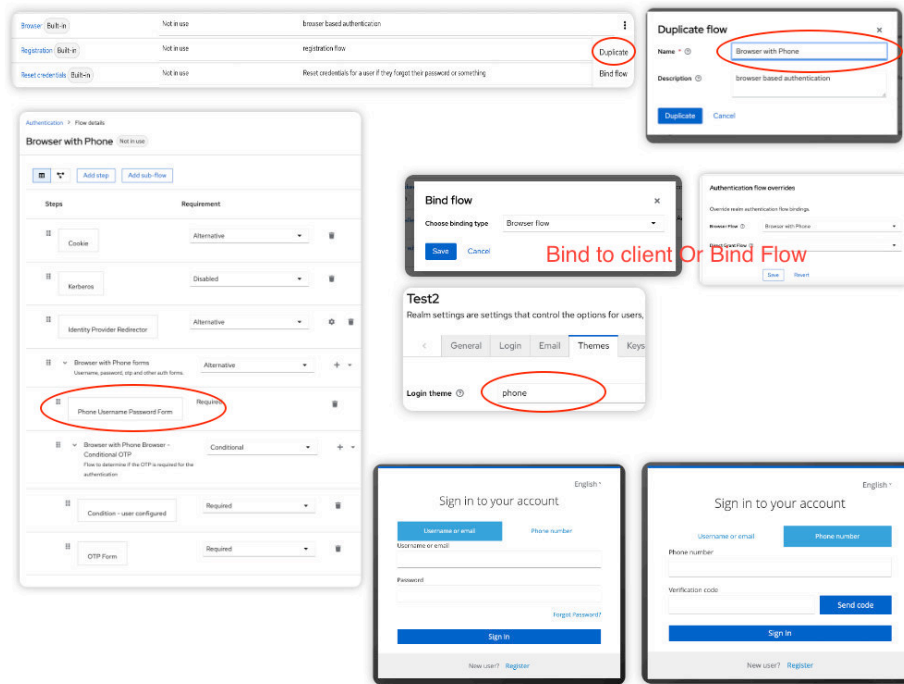
Registration URL:

`<http://<domain>>/realms/<realm name>/protocol/openid-connect/registrations?client_id=<client id>&response_type=code&scope=openid%20email&redirect_uri=<redirect uri>`

Login by phone

Under `Authentication` > `Flows`:

1. Duplicate the Browser flow to Browser with phone flow
2. Set Bind Browser with phone to Browser flow On the Authentication page, bind Browser with phone to Browser flow
3. Replace Username Password Form with Phone Username Password Form
4. Click on the settings icon next to Phone Username Password Form to configure.
5. If theme not updated, under Realm Settings > Themes Set Login Theme as phone

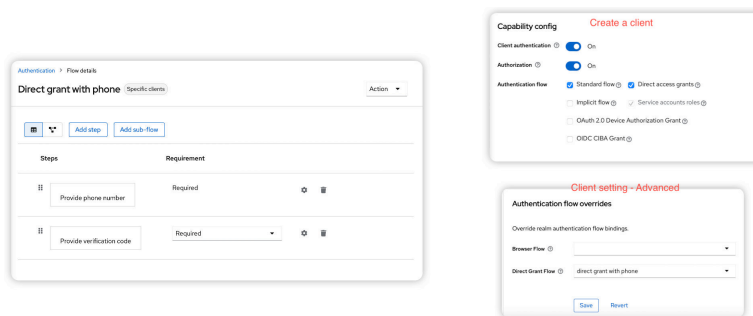


Only use phone login or get Access token use endpoints:

Under Authentication > Flows:

- Copy the Direct Grant flow to Direct grant with phone flow
- Click on Add step on the Provide Phone Number line
- Click on Add step on the Provide Verification Code line
- Delete or disable other
- Set both of Provide Phone Number and Provide Verification Code to REQUIRED

Under Clients > \$YOUR_CLIENT > Advanced > Authentication Flow Overrides Bind Direct Grant Flow to Direct grant with phone



Either Phone/OTP or Username/Password :

Authentication > Flow details

phone or username direct grant Specific clients Action

Flow successfully updated

Steps Requirement

Steps	Requirement
username verify	Conditional
Condition - phone provided	Required
Username Validation	Required
Password	Required
phone verify	Conditional
Condition - phone provided	Required
Provide phone number	Required
Provide verification code	Required

[Android client example](#)

Everybody phone number (if not exists create user by phone number) get Access token use endpoints:

Under Authentication > Flows :

- Copy the Direct Grant flow to Direct grant everybody with phone flow
- Click on Actions > Add step on the Authentication Everybody By Phone line and move to first
- Delete or disable other
- Set Authentication Everybody By Phone to REQUIRED

Under Clients > \$YOUR_CLIENT > Advanced > Authentication Flow Overrides Set Direct Grant Flow to Direct grant everybody with phone

About the API endpoints:

You'll get 2 extra endpoints that are useful to do the verification from a custom application.

- GET /realms/{realmName}/sms/verification-code?phoneNumber=+5534990001234 (To request a number verification. No auth required.)
- POST /realms/{realmName}/sms/verification-code?phoneNumber=+5534990001234&code=123456 (To verify the process. User must be authenticated.)

You'll get 2 extra endpoints that are useful to do the access token from a custom application.

- GET /realms/{realmName}/sms/authentication-code?phoneNumber=+5534990001234 (To request a number verification. No auth required.)
- POST /realms/{realmName}/protocol/openid-connect/token Content-Type: application/x-www-form-urlencoded
grant_type=password&phone_number=\$PHONE_NUMBER&code=\$VERIFICATION_CODE&client_id=\$CLIENT_ID&client_secret=\$CLIENT_SECRET

And then use Verification Code authentication flow with the code to obtain an access code.

To test the Token we can use the following API:

POST /realms/{realmName}/protocol/openid-connect/token

Params:

- 1- Authorization: Bearer token.
- 2- content-type: application/json