How do I embed static authentication links on my website?

Tolleiv Nietsch - 2026-08-28 - How To ...

In addition to the registration link, you can embed further static authentication links on your website that let users reset their password or manage their account themselves.

Creating a forgot-password link

You can embed a link for resetting the password on your own website, e.g. in case the user has forgotten their password.

- Assemble the URL from the following elements:
  - Keycloak base URL including the name of the Bare.ID instance, as shown in the dashboard: e.g. `http://login.example.com/auth/realms/Realmname`
  - Endpoint for resetting the credentials: `login-actions/reset-credentials`
  - Client ID: e.g. `?client_id=BeispielApplikation`
  - Additional parameters: e.g. `&response_type=code&scope=email&kc_locale=de`

In the example, this results in the following URL for resetting the password:

```
http://login.example.com/auth/realms/Realmname/login-actions/reset-credentials?client_id=BeispielApplikation&response_type=code&scope=email&kc_locale=de
```

Further actions after login

For the following actions (changing the password, editing the profile, deleting the account), a normal OpenID Connect/OAuth authentication request is sent to the auth endpoint – exactly as for the login – extended by the Keycloak-specific parameter `&kc_action=…`. The following applies:

- The user must be logged in. If no session exists yet, the normal login is shown first and the action is performed afterwards.
- The `redirect_uri` used must be registered in the client's *Valid Redirect URIs*, otherwise Keycloak aborts with an error.
- After the action is completed, a `code` is returned to the `redirect_uri` – just like for a normal login – which the application can exchange (or ignore) as usual.
- If the user cancels the action, the corresponding OIDC success parameter (`code`) is still passed on the redirect.
- The availability of the actions varies depending on the instance configuration. In principle, all actions from Triggering login actions are also available for such a call.
- All requirements that apply to normal authentication also apply here: requirements regarding the signature, request type (PAR, request object, PKCE), etc.

The action is determined by the value of the `kc_action` parameter:

Action	Value for `kc_action`
Change password	`UPDATE_PASSWORD`
Edit profile	`UPDATE_PROFILE`
Delete account	`delete_account`

The URL matches the auth endpoint request (as for the login), extended by the desired `kc_action` value. Example for changing the password:

```
http://login.example.com/auth/realms/Realmname/protocol/openid-connect/auth?response_type=code&client_id=BeispielApplikation&scope=openid&redirect_uri=http://login.example.com/authentication/login&kc_action=UPDATE_PASSWORD
```

> **Note:** The value of the `kc_action` parameter is case-sensitive (e.g. `UPDATE_PASSWORD` in uppercase, `delete_account` in lowercase).

For “Delete account”, the “Delete Account” feature must additionally be enabled in the Bare.ID instance and the user must have the `delete-account` role (a client role of the `account` client). Otherwise, Keycloak displays an error page.

Optional parameters

- **Language (`kc_locale`):** All four URLs (resetting the password, changing the password, editing the profile and deleting the account) can be extended with the optional parameter `&kc_locale=[two-letter language code]`. This lets you specify the language in which the displayed frontend is rendered – for example `de` or `en`. This requires internationalization and the respective language to be enabled in the Bare.ID instance.
- **CSRF protection (`state`):** As for the normal login, it is recommended to pass a random `&state=…` value to the auth endpoint URLs and to verify it after the redirect back.