Wie binde ich statische Authentifizierungs-Links auf meiner Webseite ein?

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

Neben dem Registrierungs-Link kannst Du weitere statische Authentifizierungs-Links auf Deiner Webseite einbinden, mit denen Nutzer ihr Passwort zurücksetzen oder ihr Konto selbst verwalten können.

Passwort-Vergessen-Link erstellen

Ein Link zum Zurücksetzen des Passworts kann in die eigene Webseite eingebunden werden, z. B. für den Fall, dass das Nutzerpasswort vergessen wurde.

- Setze die URL aus den folgenden Elementen zusammen:
  - Keycloak-Basis-URL inklusive Name der Bare.ID-Instanz, wie im Dashboard: z. B. `http://login.example.com/auth/realms/Realmname`
  - Endpunkt zum Zurücksetzen der Anmeldeinformationen: `login-actions/reset-credentials`
  - Client-ID: z. B. `?client_id=BeispielApplikation`
  - Weitere Parameter: z. B. `&response_type=code&scope=email&kc_locale=de`

Das ergibt im Beispiel die folgende URL zum Zurücksetzen des Passworts:

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

Weitere Aktionen nach dem Login

Für die folgenden Aktionen (Passwort ändern, Profil anpassen, Account löschen) wird jeweils ein normaler OpenID Connect-/OAuth-Authentifizierungsanfrage an den Auth-Endpunkt gestellt – genauso wie beim Login –, ergänzt um den Keycloak-spezifischen Parameter `&kc_action=…`. Dabei gilt:

- Der Nutzer muss angemeldet sein. Besteht noch keine Sitzung, wird zunächst der normale Login angezeigt und die Aktion anschließend ausgeführt.
- Die verwendete `redirect_uri` muss in den *Valid Redirect URIs* des Clients hinterlegt sein, sonst bricht Keycloak mit einem Fehler ab.
- Nach Abschluss der Aktion wird – wie beim normalen Login – ein `code` an die `redirect_uri` zurückgegeben, den die Applikation wie gewohnt einlösen (oder ignorieren) kann.
- Bricht der Nutzer die Aktion ab, wird beim Redirect dennoch der entsprechende OIDC-Erfolgsparameter (`code`) mitgeteilt.
- Die Verfügbarkeit der Aktionen weicht je nach Instanz-Konfiguration ab. Grundsätzlich stehen für einen solchen Aufruf auch alle Aktionen aus Login-Aktionen anstoßen zur Verfügung.
- Es gelten alle Anforderungen, die auch für die normale Authentifizierung gelten: Anforderungen an Signatur, Aufruftyp (PAR, Request Object, PKCE) etc.

Die Aktion wird über den Wert des Parameters `kc_action` bestimmt:

Aktion	Wert für `kc_action`
Passwort ändern	`UPDATE_PASSWORD`
Profil anpassen	`UPDATE_PROFILE`
Account löschen	`delete_account`

Die URL entspricht dem Auth-Endpoint-Request (wie beim Login), ergänzt um den gewünschten `kc_action`-Wert. Beispiel für das Ändern des Passworts:

```
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
```

> **Hinweis:** Beim Wert des Parameters `kc_action` ist die Groß- und Kleinschreibung wichtig (z. B. `UPDATE_PASSWORD` in Großbuchstaben, `delete_account` in Kleinbuchstaben).

Für „Account löschen" muss zusätzlich in der Bare.ID-Instanz die Funktion „Delete Account" aktiviert sein und der Nutzer über die Rolle `delete-account` (Client-Rolle des `account`-Clients) verfügen. Andernfalls zeigt Keycloak eine Fehlerseite an.

Optionale Parameter

- **Sprache (`kc_locale`):** Alle vier URLs (Passwort zurücksetzen, Passwort ändern, Profil anpassen und Account löschen) können mit dem optionalen Parameter `&kc_locale=[zweistelliger Sprachcode]` versehen werden. Damit gibst Du an, in welcher Sprache das angezeigte Frontend dargestellt wird – also z. B. `de` oder `en`. Voraussetzung ist, dass die Internationalisierung und die jeweilige Sprache in der Bare.ID-Instanz aktiviert sind.
- **CSRF-Schutz (`state`):** Wie beim normalen Login empfiehlt es sich, den Auth-Endpoint-URLs einen zufälligen `&state=…`-Wert mitzugeben und diesen nach der Rückleitung zu prüfen.