

Walkthrough: Wordpress-Seite via OAuth an die MiData anbinden

Was ist OAuth?

Hitobito ist ein OAuth 2.0-Anbieter, was bedeutet, dass eine externe Anwendung Benutzer*innen über hitobito authentifizieren kann (normalerweise in Form einer "Login via hitobito"-Funktion, ähnlich wie bei Google und Facebook usw.). In dieser Anleitung lernst du, wie du das MiData-Login für deine Wordpress-Seite verwenden kannst.

Testsystem verwenden

Immer wenn du an einer Schnittstelle zur MiData arbeitest, solltest du dein Vorhaben zuerst auf dem Testsystem der MiData ausprobieren. So merkst du beispielsweise, wenn deine Applikation einen Fehler auf dem System verursacht oder wenn du ein ganz anderes Plugin einsetzen musst.

Mehr Infos zum Testsystem findest du im [F.A.Q.](#)

Konfiguration der OAuth App im MiData

Eine neue OAuth-Applikation in MiData erstellen: <https://pbs.puzzle.ch/de/oauth/applications>

<p>Name* <input type="text" value="Pfadi Laupen Test Webseite"/></p> <p>Redirect URIs <input type="text" value="https://pfadilaupen.ch/wp-admin/admin-ajax.php?action=openid-connect-authorize"/></p> <p><small>Ein Eintrag pro Zeile. Für lokale Tests urn:iETF:wg:oauth:2.0:oob verwenden.</small></p> <p>Scopes <input checked="" type="checkbox"/> Lesen deiner E-Mail Adresse (email) <input checked="" type="checkbox"/> Lesen deiner E-Mail Adresse und Name (name) <input type="checkbox"/> Lesen deiner Stammdaten und Rollen (with_roles) <input checked="" type="checkbox"/> Lesen deines OIDC Identity Tokens (openid) <input type="checkbox"/> Lesen aller Personen, Gruppen, Events und Rechnungen auf die du Zugriff hast, via die JSON-Schnittstellen (api)</p> <p>Logo <input type="button" value="Choose File"/> No file chosen</p> <p>Hosts mit API-Zugriff <input type="button" value="Eintrag hinzufügen"/></p> <p><small>Falls der Scope "api" aktiviert ist, dürfen Webseiten auf diesen Hosts die Daten aus der JSON-Schnittstelle vom Browser aus abrufen (CORS).</small></p>	<p>Folgende Informationen müssen dafür angegeben werden</p> <ul style="list-style-type: none">- Name der Applikation- Redirect URIs- Scopes
--	---

Installation und Konfiguration des Wordpress Plugins

Sobald die OAuth Applikation erstellt wurde, kann Wordpress als OAuth Client angebunden werden. Dafür ist die Installation und Konfiguration eines zusätzlichen Plugins notwendig.

	https://wordpress.org/plugins/daggerhart-openid-connect-generic/
	Das Plugin «OpenID Connect Generic Client» via Wordpress Plugin Manager installieren und aktivieren
	Folgende Plugin Konfiguration vornehmen <ul style="list-style-type: none">- Login Type: OpenID Connect button on login form- Client ID: <Wert aus MiData nehmen>- Client Secret Key: <Wert aus MiData nehmen>- ClientID Scope: openid email name- Login Endpoint URL:<ul style="list-style-type: none">o Test: https://pbs.puzzle.ch/oauth/authorizeo Produktion: https://db.scout.ch/oauth/authorize- Userinfo Endpoint URL:<ul style="list-style-type: none">o Test: https://pbs.puzzle.ch/oauth/userinfoo Produktion: https://db.scout.ch/oauth/userinfo- Token Validation Endpoint URL:<ul style="list-style-type: none">o Test: https://pbs.puzzle.ch/oauth/tokeno Produktion: https://db.scout.ch/oauth/token- End Session Endpoint URL:<ul style="list-style-type: none">o Test: https://pbs.puzzle.ch/oauth/logouto Produktion: https://db.scout.ch/oauth/logout

Nickname Key
 Where in the user claim array to find the user's nickname. Possible standard values: preferred_username, name, or sub.
 Example: preferred_username

Email Formatting
 String from which the user's email address is built. Specify "[email]" as long as the user claim contains an email claim.
 Example: {email}

Display Name Formatting
 String from which the user's display name is built.
 Example: {given_name} {family_name}

Identify with User Name
 If checked, the user's identity will be determined by the user name instead of the email address.

State time limit
 State valid time in seconds. Defaults to 180

Enable Refresh Token
 If checked, support refresh tokens used to obtain access tokens from supported IDPs.

WordPress User Settings
 Modify the interaction between OpenID Connect and WordPress users.

Link Existing Users
 If a WordPress account already exists with the same identity as a newly-authenticated user over OpenID Connect, login as that user instead of generating an error.

Create user if does not exist
 If the user identity is not linked to an existing WordPress user, it is created. If this setting is not enabled, and if the user authenticates with an account which is not linked to an existing WordPress user, then the authentication will fail.

Redirect Back to Origin Page
 After a successful OpenID Connect authentication, this will redirect the user back to the page on which they clicked the OpenID Connect login button. This will cause the login process to proceed in a traditional WordPress fashion. For example, users logging in through the default wp-login.php page would end up on the WordPress Dashboard and users logging in through the WooCommerce "My Account" page would end up on their account page.

Redirect to the login screen when session is expired
 When enabled, this will automatically redirect the user back to the WordPress login page if their access token has expired.

Authorization Settings
 Control the authorization mechanics of the site.

Enforce Privacy
 Require users be logged in to see the site.

Alternate Redirect URI
 Provide an alternative redirect route. Useful if your server is causing issues with the default admin-ajax method. You must flush rewrite rules after changing this setting. This can be done by saving the Permalinks settings page.

Log Settings
 Log information about login attempts through OpenID Connect Generic.

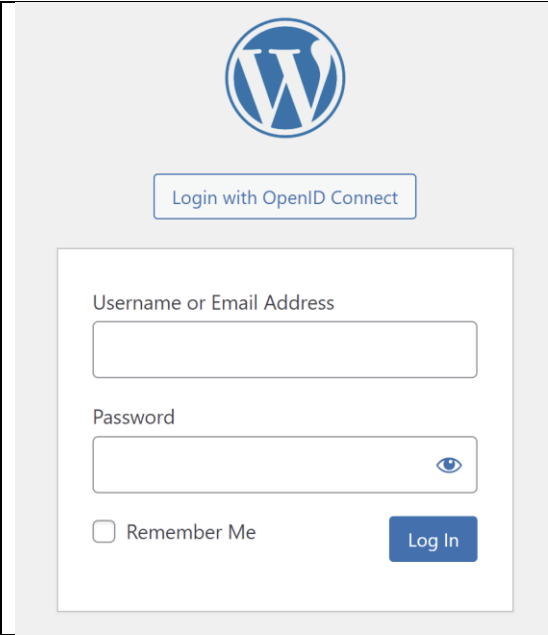

Enable Logging
 Very simple log messages for debugging purposes.

Log Limit
 Number of items to keep in the log. These logs are stored as an option in the database, so space is limited.

○ Produktion:
<https://db.scout.ch/oauth/logo>
[ut](#)

- Identity Key: email
- HTTP Request Timeout: 5
- Email Formatting: {email}
- Display Name Formatting: {nickname}
- Enable Refresh Token: true
- Link Existing Users: true
- Create user if does not exist: false, somit müssen die User, welche über OAuth authentifiziert werden, vorgängig innerhalb von Wordpress angelegt werden.
- Redirect back to Origin Page: false
- Redirect to the login screen when session expired: true
- Enforce Privacy: false
- Alternate Redirect URI: false
- Enable Logging: false

Login Experience

	<p>Aufruf der Wordpress Login Seite (z.B https://wordpresseite.ch/wp-admin/).</p> <p>Auf "Login with OpenID Connect" klicken</p>
	<p>Mithilfe eines MiData Accounts sich anmelden</p>

Wechsel aufs produktive System

Sobald du dir sicher bist, dass deine Applikation das macht, was sie soll, kannst du deinen Zugang für die «richtige» MiData beantragen.

OAuth API Key beantragen

Über das nachfolgende Formular kann auf der produktiven MiData (<https://db.scout.ch/>) eine OAuth Applikation beantragt werden:

<https://forms.office.com/Pages/ResponsePage.aspx?id=iq6Fcs2Xq0m9ordFTZ0Fa8gnQG-i3p9KkbcKGL9nFhtUMEpMQkYwMzQxNUVEWEIxRTNWTdhpMDVEMS4u>