

Démarche : Connecter un site Wordpress à MiData via OAuth

Qu'est-ce que OAuth ?

Hitobito est un fournisseur OAuth 2.0, ce qui signifie qu'une application externe peut authentifier les utilisateurs via hitobito (généralement sous la forme d'une fonction " connexion via hitobito ", similaire à celle de Google et de Facebook, etc.) Dans ce guide, tu apprendras comment utiliser le login de MiData pour ton site Wordpress.

Utiliser un système de test

A chaque fois que tu travailles sur une interface de MiData, tu devrais d'abord tester ton projet sur le système de test de MiData. Tu pourras ainsi remarquer si ton application provoque une erreur sur le système ou si tu dois utiliser un plugin complètement différent.

Tu trouveras plus d'informations sur le système de test dans le [FAQ](#)

Configuration de l'application OAuth dans MiData

Créer une nouvelle application OAuth dans MiData : <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="text" value="Eintrag hinzufügen"/> <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>Les informations suivantes doivent être fournies pour cela</p> <ul style="list-style-type: none">- - Nom de l'application- - Redirect URIs- - Scopes
---	---

Installation et configuration du plugin du Wordpress

Dès que l'application OAuth a été créée, Wordpress peut être connecté en tant que client OAuth. Pour cela, il est nécessaire d'installer et de configurer un plugin supplémentaire.

	https://wordpress.org/plugins/daggerhart-openid-connect-generic/
	Installer et activer le plugin "OpenID Connect Generic Client" via le Wordpress Plugin Manager
<p>Client Settings</p> <p>Enter your OpenID Connect identity provider settings.</p> <p>Login Type OpenID Connect button on login form Select how the client (login form) should provide login options.</p> <p>Client ID W6pi3xDI-QjDOWpn293TJKCS5phKvsl1m5WLSUPTE The ID this client will be recognized as when connecting to the Identity provider server. Example: my-wordpress-client-id</p> <p>Client Secret Key 23CWPOFX3aJsDuAHHNrxWtpFt5_uthedHwdK9XL4dFo Arbitrary secret key the server expects from this client. Can be anything, but should be very unique.</p> <p>OpenID Scope openid email name Space separated list of scopes this client should access. Example: email profile openid offline_access</p> <p>Login Endpoint URL https://pbs.puzzle.ch/oauth/authorize Identify provider authorization endpoint. Example: https://example.com/oauth2/authorize</p> <p>Userinfo Endpoint URL https://pbs.puzzle.ch/oauth/userinfo Identify provider User information endpoint. Example: https://example.com/oauth2/userinfo</p> <p>Token Validation Endpoint URL https://pbs.puzzle.ch/oauth/token Identify provider token endpoint. Example: https://example.com/oauth2/token</p> <p>End Session Endpoint URL https://pbs.puzzle.ch/oauth/logout Identify provider logout endpoint. Example: https://example.com/oauth2/logout</p> <p>ACR values Use a specific defined authentication contract from the IDP - optional.</p> <p>Identity Key email Where in the user claim array to find the user's identification data. Possible standard values: preferred_username, name, or sub. If you're having trouble, use "sub". Example: preferred_username</p> <p>Disable SSL Verify <input type="checkbox"/> Do not require SSL verification during authorization. The OAuth extension uses curl to make the request. By default CURL will generally verify the SSL certificate to see if its valid an issued by an accepted CA. This setting disabled that verification. Not recommended for production sites.</p> <p>HTTP Request Timeout 5 Set the timeout for requests made to the IDP. Default value is 5. Example: 30</p>	Effectuer la configuration suivante du plugin <ul style="list-style-type: none">- Type de connexion : OpenID Connect button on login form- Client ID : < Prendre la valeur dans MiData >- Client Secret Key : < Prendre la valeur dans MiData >- ClientID Scope : openid email name- Login Endpoint URL :<ul style="list-style-type: none">o Test: https://pbs.puzzle.ch/oauth/authorizeo Production : https://db.scout.ch/oauth/authorize- Userinfo Endpoint URL :<ul style="list-style-type: none">o Test: https://pbs.puzzle.ch/oauth/userinfoo Production: https://db.scout.ch/oauth/userinfo- Token Validation Endpoint URL:<ul style="list-style-type: none">o Test: https://pbs.puzzle.ch/oauth/tokeno Production: https://db.scout.ch/oauth/token- End Session Endpoint URL:<ul style="list-style-type: none">o Test: https://pbs.puzzle.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.

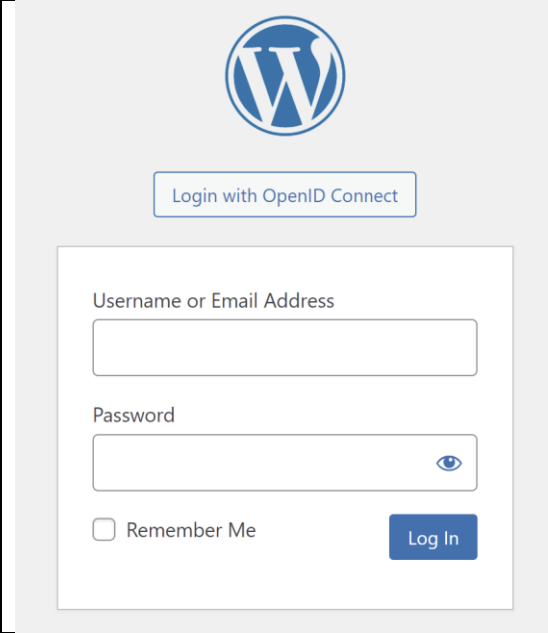

Enable Logging
 Very simple log messages for debugging purposes.

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

○ Production :
<https://db.scout.ch/oauth/logout>

- 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, les utilisateurs qui sont authentifiés via OAuth doivent donc être créés au préalable dans Wordpress.
- 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

Expérience de connexion

	<p>Appel de la page de connexion de Wordpress (par ex. https://wordpressite.ch/wp-admin/).</p> <p>Cliquer sur "Login with OpenID Connect"</p>
	<p>Se connecter à l'aide d'un compte MiData</p>

Passage au système productif

Une fois que tu es sûr que ton application fait ce qu'elle doit faire, tu peux demander ton accès pour la "vraie" MiData.

Demander l'OAuth API Key

Une application OAuth sur la MiData productive (<https://db.scout.ch/>) peut être demandée via le formulaire suivant:

<https://forms.office.com/Pages/ResponsePage.aspx?id=iq6Fcs2Xq0m9ordFTZ0Fa8gnQG-i3p9KkbcKGL9nFhtUMEpMQkYwMzQxNUVEWEixRTNWTdhpMDVEMS4u>