

# Cacert OpenID Connect Registration Examples

## Nextcloud


I use the “OpenID Connect” module in my Nextcloud server.

Click on the “plus sign” next to Registered Providers to add the connection to CAcert OpenID.

### OpenID Connect

Allows users to authenticate via OpenID Connect providers.

☐ Enable ID4me

☐ Store login tokens 

Registered Providers +

Just under the title, “Register a new provider,” you will see the redirection URL. Copy that.

**Register a new provider**


Configure your provider to redirect back to [https://baideal.buadh-brath.com/apps/user\\_oidc/code](https://baideal.buadh-brath.com/apps/user_oidc/code)

### Client configuration

Identifier (max 128 characters)

Client ID

Client secret

 Warning, if the protocol of the URLs in the discovery content is HTTP, the ID token will be delivered through an insecure connection.

Discovery endpoint

Custom end session endpoint

Scope

Extra claims

### Attribute mapping

☐ Enable nested and fallback claim mappings (like "custom.nickname | profile.name | name")

User ID mapping

Quota mapping

Groups mapping

Scrolling down that page, you will see more fields, primarily more check boxes.

## Register a new provider

User ID mapping

Quota mapping

Groups mapping

> Extra attributes mapping

### Authentication and Access Control Settings

☒ Use unique user ID

By default every user will get a unique user ID that is a hashed value of the provider and user ID. This can be turned off but uniqueness of users accross multiple user backends and providers is no longer preserved then.

☐ Use provider identifier as prefix for IDs

To keep IDs in plain text, but also preserve uniqueness of them across multiple providers, a prefix with the providers name is added.

☐ Use group provisioning.

This will create and update the users groups depending on the groups claim in the ID token. The Format of the groups claim value should be [{gid: "1", displayName: "group1"}, ...], ["group1", "group2", ...] or "group1,group2"

Group whitelist regex

Only groups matching the whitelist regex will be created, updated and deleted by the group claim. For example: /^blue/ allows all groups which ID starts with blue

☐ Restrict login for users that are not in any whitelisted group

Users that are not part of any whitelisted group are not created and can not login

☐ Check Bearer token on API and WebDAV requests

Do you want to allow API calls and WebDAV requests that are authenticated with an OIDC ID token or access token?

☐ Auto provision user when accessing API and WebDAV with Bearer token

Cancel

✓ Submit

## Register a new provider

### Authentication and Access Control Settings

☒ Use unique user ID

By default every user will get a unique user ID that is a hashed value of the provider and user ID. This can be turned off but uniqueness of users accross multiple user backends and providers is no longer preserved then.

☐ Use provider identifier as prefix for IDs

To keep IDs in plain text, but also preserve uniqueness of them across multiple providers, a prefix with the providers name is added.

☐ Use group provisioning.

This will create and update the users groups depending on the groups claim in the ID token. The Format of the groups claim value should be `[[{"gid": "1", displayName: "group1"}, ...], [{"group1", "group2", ...}]` or `"group1,group2"`

Group whitelist regex

Only groups matching the whitelist regex will be created, updated and deleted by the group claim. For example: `/^blue/` allows all groups which ID starts with blue

☐ Restrict login for users that are not in any whitelisted group

Users that are not part of any whitelisted group are not created and can not login

☐ Check Bearer token on API and WebDAV requests

Do you want to allow API calls and WebDAV requests that are authenticated with an OIDC ID token or access token?

☐ Auto provision user when accessing API and WebDAV with Bearer token

This automatically provisions the user, when sending API and WebDAV requests with a Bearer token. Auto provisioning and Bearer token check have to be activated for this to work.

☒ Send ID token hint on logout

Should the ID token be included as the `id_token_hint` GET parameter in the OpenID logout URL? Users are redirected to this URL after logging out of Nextcloud. Enabling this setting exposes the OIDC ID token to the user agent, which may not be necessary depending on the OIDC provider.

Cancel

✓ Submit

We will come back to those later.

Go to the CAcert OpenID Registration Site and start the “Register A New Site” process. Choose a recognizable name for your Nextcloud server, and put the Redirection URL in to the appropriate field.

Add any notes that you wish.

Click **Register Site**.

The screenshot shows the CAcert OpenID Registration Site. The top header features the CAcert logo and a 'Home' button. The main content area is titled 'Here are your Client ID and Client Secret' and includes a warning: 'Make sure that you copy these two values somewhere safe, because the Client Secret can not be retrieved again.' Below this, a table displays the Client ID and Client Secret. Further down, another table lists the URLs for Authorize, Token, and User Info. A link to download a list of relevant URLs is provided. The bottom section contains a form for registering a new site, with fields for Site Name, Redirect URL, Auth Method (Post or Basic), and Notes. A 'Register Site' button is at the bottom of the form. The footer of the page reads 'Copyright © CAcert, Inc 2025'.

**Here are your Client ID and Client Secret**

Make sure that you copy these two values somewhere safe, because the Client Secret can not be retrieved again.

Client ID	3flb030a-930d-47bd-a40a-8237260267ee
Client Secret	0pnSk-8el_pQuaYhZlXhWLG6yp

You will also need the following URLs for your site plugin.

Authorize	https://authserver.cacert-phoenix.org:4444/oauth2/auth
Token	https://authserver.cacert-phoenix.org:4444/oauth2/token
User Info	https://authserver.cacert-phoenix.org:4444/userinfo

Download a list of relevant URLs [here](#)

Copyright © CAcert, Inc 2025

2. The "Redirect URL" that the OpenID plugin for the site that you are registering requires.

3. Does your plugin use Post or Basic authentication? The default is POST. If you have requirements other than those two, please contact the CAcert OpenID development team at [bmccullough@cacert.org](mailto:bmccullough@cacert.org)

4. Optional notes for yourself.

You may download the various URLs provided by the OpenID component [here](#)

Site Name	NextCloud Site
Redirect URL	https://<My Domain Name>/apps/user_oidc/code
Auth Method	<input checked="" type="radio"/> Post <input type="radio"/> Basic
Notes	

Copyright © CAcert, Inc 2025

Save the Client ID and Client Secret in a safe place. The Registration Site WILL NOT allow you to retrieve those values again.

Back on the Nextcloud form, enter those two values in the appropriate locations.

**Register a new provider**


Configure your provider to redirect back to [https://baideal.buadh-brath.com/apps/user\\_oidc/code](https://baideal.buadh-brath.com/apps/user_oidc/code)

### Client configuration

Identifier (max 128 characters)

Client ID

Client secret

 Warning, if the protocol of the URLs in the discovery content is HTTP, the ID token will be delivered through an insecure connection.

Discovery endpoint

Custom end session endpoint

Scope

Extra claims

### Attribute mapping

☐ Enable nested and fallback claim mappings (like "custom.nickname | profile.name | name")

User ID mapping

Quota mapping

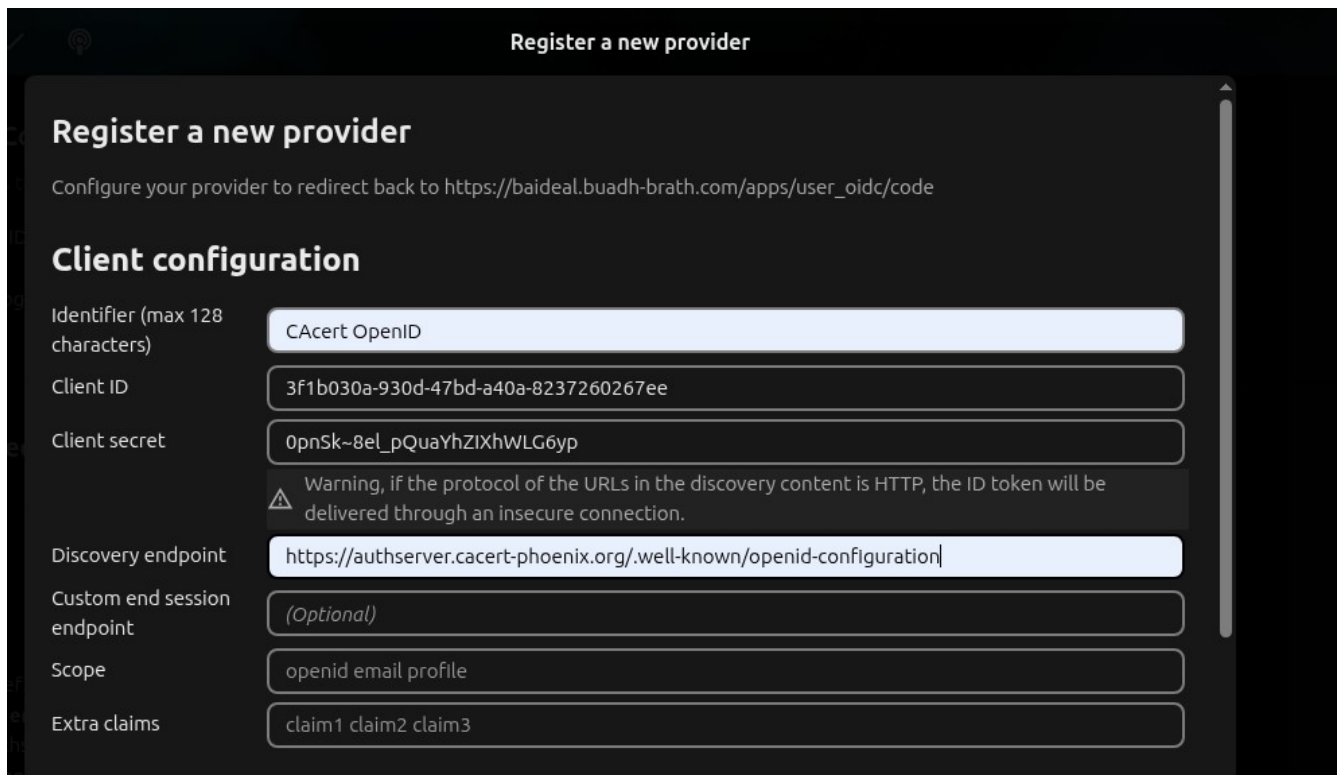
Groups mapping

[https://baideal.buadh-brath.com/apps/user\\_oidc/backchannel-logout/CAcert%20OpenID%20202](https://baideal.buadh-brath.com/apps/user_oidc/backchannel-logout/CAcert%20OpenID%20202)

In the CAcert OpenID Registration Site, copy the value of the link shown at the bottom of the form, labelled “Download a list of relevant URLs here”.

Download a list of relevant URLs [here](#)

Paste that value into the field in the Nextcloud form into the field labelled “Discovery endpoint.”



The screenshot shows the 'Register a new provider' form in Nextcloud. The form is titled 'Register a new provider' and includes a subtitle: 'Configure your provider to redirect back to https://baideal.buadh-brath.com/apps/user\_oidc/code'. The form is divided into a 'Client configuration' section. The fields and their values are as follows:

Field	Value
Identifier (max 128 characters)	CACert OpenID
Client ID	3f1b030a-930d-47bd-a40a-8237260267ee
Client secret	0pnSk~8el_pQuaYhZIXhWLG6yp
Discovery endpoint	https://authserver.cacert-phoenix.org/.well-known/openid-configuration
Custom end session endpoint	(Optional)
Scope	openid email profile
Extra claims	claim1 claim2 claim3

A warning message is displayed below the Client secret field: 'Warning, if the protocol of the URLs in the discovery content is HTTP, the ID token will be delivered through an insecure connection.'

**Note:** The default configuration of this Nextcloud module is to create new users based on the OpenID Provider and e-mail address. When a new user logs in using OpenID, a new account is created in your Nextcloud system. If you want existing users to be able to log in using their existing credentials, then Uncheck the box labelled “Use unique user ID”.

The screenshot shows the 'Register a new provider' configuration page. At the top, there's a title bar with a question mark icon and the text 'Register a new provider'. Below this, there are two input fields: 'Quota mapping' with the value 'quota' and 'Groups mapping' with the value 'groups'. A button labeled '> Extra attributes mapping' is visible. The main section is titled 'Authentication and Access Control Settings'. It contains two checkboxes, both of which are unchecked. The first checkbox is 'Use unique user ID', followed by a paragraph explaining that by default, every user gets a unique user ID (a hashed value of the provider and user ID), and this can be turned off, but uniqueness across multiple user backends and providers is no longer preserved. The second checkbox is 'Use provider identifier as prefix for IDs', followed by a paragraph explaining that to keep IDs in plain text while preserving uniqueness across multiple providers, a prefix with the provider's name is added.

Register a new provider

Quota mapping

Groups mapping

> Extra attributes mapping

### Authentication and Access Control Settings

☐ Use unique user ID

By default every user will get a unique user ID that is a hashed value of the provider and user ID. This can be turned off but uniqueness of users across multiple user backends and providers is no longer preserved then.

☐ Use provider identifier as prefix for IDs

To keep IDs in plain text, but also preserve uniqueness of them across multiple providers, a prefix with the provider's name is added.

Finally, click the Submit button at the bottom right of the page.

This screenshot shows a modal dialog box with a checkbox labeled 'Auto provision user when accessing API and WebDAV with Bearer token'. Below the checkbox is a paragraph explaining that this automatically provisions the user when sending API and WebDAV requests with a Bearer token, and that auto provisioning and Bearer token checks must be activated for this to work. At the bottom right of the dialog are two buttons: 'Cancel' and 'Submit'. The 'Submit' button is highlighted with a checkmark icon.

Do you want to allow API calls and WebDAV requests that are authenticated with an OIDC ID token or access token?

☐ Auto provision user when accessing API and WebDAV with Bearer token

This automatically provisions the user, when sending API and WebDAV requests with a Bearer token. Auto provisioning and Bearer token check have to be activated for this to work.


Cancel Submit

You will be returned to the OpenID Connect configuration page with your new entry showing.



## OpenID Connect

Allows users to authenticate via OpenID Connect providers.

- ☐ Enable ID4me
- ☐ Store login tokens 

## Registered Providers +

### CACert OpenID



#### Client ID

a439411f-ef3d-4f26-a298-e38755feb8e0

#### Discovery endpoint

<https://authserver.cacert-phoenix.org/.well-known/openid-configuration>

#### Backchannel Logout URL

[https://baideal.buadh-brath.com/apps/user\\_oidc/backchannel-logout/CACert%20OpenID](https://baideal.buadh-brath.com/apps/user_oidc/backchannel-logout/CACert%20OpenID)

#### Redirect URI (to be authorized in the provider client configuration)

[https://baideal.buadh-brath.com/apps/user\\_oidc/code](https://baideal.buadh-brath.com/apps/user_oidc/code)

My example, shown above, has a different name and Client ID, because I had already completed configuring my NextCloud before creating this example.