
CCH Tagetik

OpenID Connect 1.0 Authentication Configuration

Document control

Document details	
Title	OpenID Connect 1.0 Authentication Configuration
Issue date	30/05/2024
Document status	Confidential
Version	V1

Version history

Version	Date	Author	Comment
V1	30/05/2024	Team BrusCKetta	Update section CCH Tagetik Server Configuration, with new authentication configuration option

Content

1. Introduction	4
2. Overview of OpenID Connect 1.0	5
2.1 Authorization Code Flow	5
3. OpenID Connect 1.0 implementation details for CCH Tagetik	7
3.1 ID Token validation	7
3.2 Claims validation	7
3.3 User Claims	7
3.4 OAuth2 Scopes	7
3.5 Logout	7
4. CCH Tagetik Server Configuration	8
5. CCH Tagetik Authentication Configuration	9
6. User management	10
6.1 Alternate Login Access	10
6.2 External Authentication Configuration – OpenID	11
7. Glossary	12
8. Notes and Copyright	13

1. Introduction

As of version 5.3 SP5 CCH Tagetik provides integration via OpenID Connect 1.0 (OIDC), a single sign-on (SSO) solution which allows computing clients to verify the identity of an end-user based on the authentication performed by an authorization server, as well as to obtain basic profile information about the end-user in an interoperable and REST-like manner.

This guide will describe how to configure the application server and CCH Tagetik to enable users authenticated by a provider, to be transparently authenticated in CCH Tagetik using OpenID Connect, based on the same credentials and without the need to retype passwords.

Due to its technical nature, this manual is intended for IT administrators who have experience with their installed JEE application server, CCH Tagetik authentication and its security model and OpenID Connect. The reader will require a CCH Tagetik administrative user with access to the Repository and the configuration of authentication in CCH Tagetik. For customers of CCH Tagetik Cloud Services, some or most of the following steps may be carried out by CCH Tagetik personnel. Please contact CCH Tagetik concerning any questions or comments related to the contents of this document.

2. Overview of OpenID Connect 1.0

This section is intended to provide an overview of the aspects of OpenID Connect which are relevant to the integration with CCH Tagetik.

OpenID Connect 1.0 is a simple identity layer on top of the [OAuth 2.0 protocol](https://oauth.net/2/) (<https://oauth.net/2/>). It enables Clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User in an interoperable and REST-like manner. It uses simple JSON Web Tokens (JWT), which you can obtain using flows conforming to the OAuth 2.0 specifications.

OpenID Connect implements authentication as an extension to the OAuth 2.0 authorization process. Use of this extension is requested by Clients by including the *openid* scope value in the Authorization Request. Information about the authentication performed is returned in a JSON Web Token (JWT) called an ID Token. OAuth 2.0 Authentication Servers implementing OpenID Connect are also referred to as OpenID Providers (OPs). OAuth 2.0 Clients using OpenID Connect are also referred to as Relying Parties (RPs).

The OpenID Connect protocol, in abstract, follows the following steps.

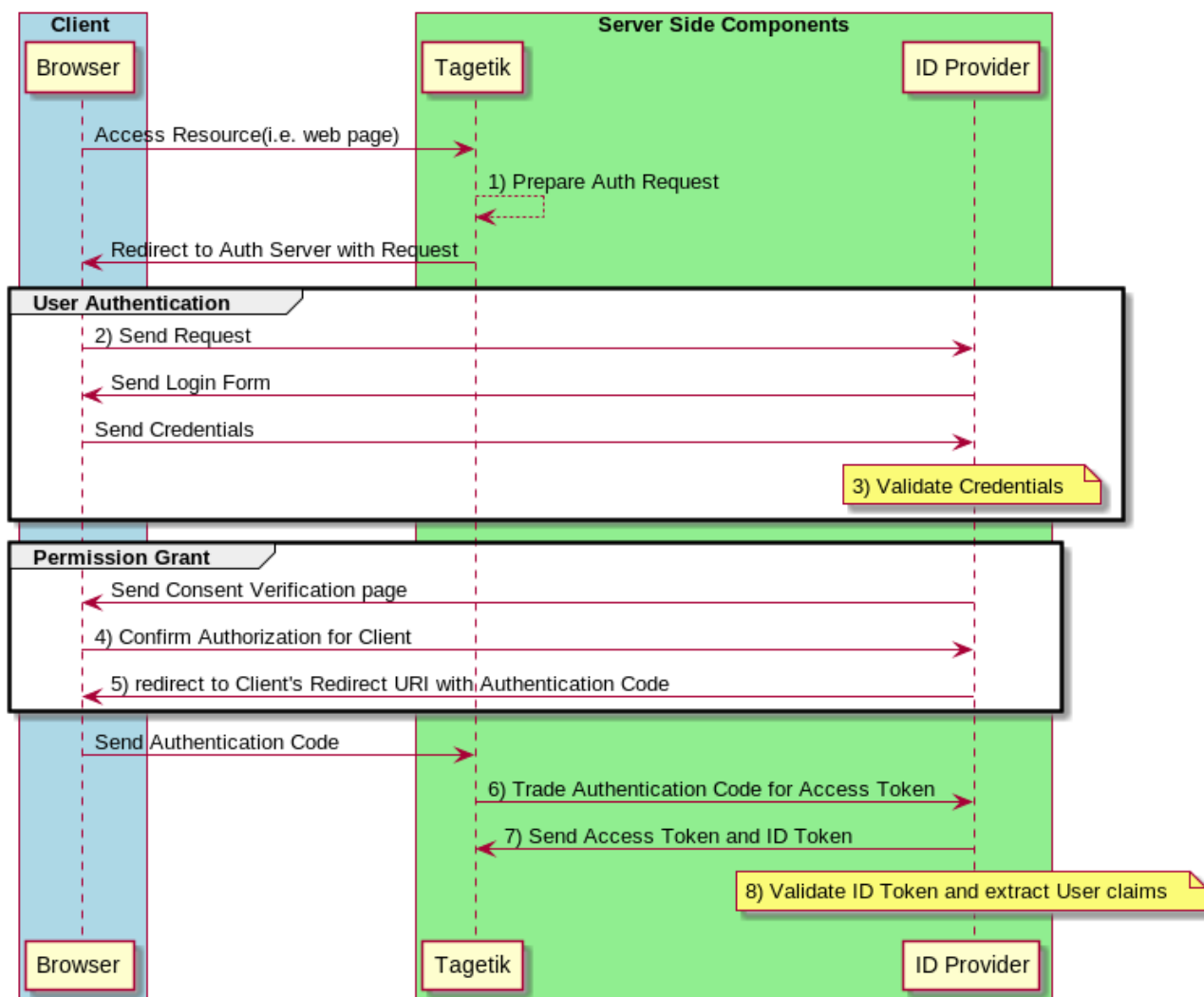
1. The RP (Client) sends a request to the OpenID Provider (OP).
2. The OP authenticates the End-User and obtains authorization.
3. The OP responds with an ID Token and usually an Access Token.
4. The RP can send a request with the Access Token to the UserInfo Endpoint.
5. The UserInfo Endpoint returns Claims about the End-User.

The OpenID Connect standard defines three different authentication flows: the Authorization Code Flow, the Implicit Flow and the Hybrid Flow. These flows determine how the ID Token and Access Token are returned to the Client.

CCH Tagetik implements the Authorization Code Flow which is the one expected for Web Applications.

2.1 Authorization Code Flow

The Authorization Code Flow returns an Authorization Code to the Client, which can then exchange it for an ID Token and an Access Token directly. This provides the benefit of not exposing any tokens to the User Agent and possibly other malicious applications with access to the User Agent. The Authorization Server can also authenticate the Client before exchanging the Authorization Code for an Access Token. The Authorization Code flow is suitable for Clients that can securely maintain a Client Secret between themselves and the Authorization Server.



The Authorization Code Flow goes through the following steps:

1. Client prepares an Authentication Request containing the desired request parameters.
2. Client sends the request to the Authorization Server.
3. Authorization Server Authenticates the End-User.
4. Authorization Server obtains End-User Consent/Authorization.
5. Authorization Server sends the End-User back to the Client with an Authorization Code.
6. Client requests a response using the Authorization Code at the Token Endpoint.
7. Client receives a response that contains an ID Token and Access Token in the response body.
8. Client validates the ID token and retrieves the End-User's Subject Identifier.

(Source: [OpenID Connect 1.0 Core](#))

3. OpenID Connect 1.0 implementation details for CCH Tagetik

3.1 ID Token validation

ID Token validation is performed using the JSON Web Signature standard. This relies on an endpoint to retrieve the algorithm and the encryption keys used for signing.

CCH Tagetik performs validation if and only if the *jwt.endpoint* key has been specified in the .properties configuration file (see [below](#)).

3.2 Claims validation

The following are the CCH Tagetik claims validation rules (please refer to http://openid.net/specs/openid-connect-core-1_0.html#IDTokenValidation):

- “exp” must be before the current date.
- “iss” must be present and must match the expected issuer.
- “aud” must match the client identifier.
- “azp”, if present, must match the client identifier.

3.3 User Claims

User claims are read from the ID Token. If all the expected claims are present or the *userinfo* endpoint has not been provided in the configuration, then these claims are used to populate user information. Otherwise a call to the *userinfo* endpoint is performed and the returned claims are used.

Two extra claims are added to the ID Token:

- “tgkUser” contains the role of the logged user, it can be user or admin.
- “referenceUsers” contains the reference users to assign to the logged user in Tagetik.

3.4 OAuth2 Scopes

CCH Tagetik OpenID Implementation is requesting the *profile*, *email*, *phone* and *openid* scopes and this is not configurable at the time of writing.

3.5 Logout

If the *revoke.endpoint* property is set then CCH Tagetik is using the endpoint to invalidate access token and refresh token (if present) upon logout.

4.CCH Tagetik Server Configuration

In order to enable and provide parameters for the OpenID Connect authentication, there are two options.

The first one consists in setting the following system property and the specified .properties file needs to be created with the appropriate values.

- *it.grupposervizi.easy.openid.configuration.properties.path*: path to a .properties file on the file systems that contains OpenID Connect authentication parameters.

The following is an example of an OpenID Connect .properties file for Google Authentication.

```
client.id=myClientID
client.secret=myClientSecret
authorization.endpoint=https://accounts.google.com/o/oauth2/v2/auth
token.endpoint=https://www.googleapis.com/oauth2/v4/token
redirect.uri=http://[host]:[port]/tagetikcpm
jwk.endpoint=https://www.googleapis.com/oauth2/v3/certs
revoke.endpoint=https://accounts.google.com/o/oauth2/revoke
userinfo.endpoint=https://www.googleapis.com/oauth2/v3/userinfo
acr.values=value1,value2
username.claim=myusernameclaim
issuer=https://accounts.google.com
```

The second option, instead, consists in defining the entire set of the above OpenID Connect authentication parameters, with prefix "it.grupposervizi.easy", inside the system properties file as separated system properties.

With this option the property *it.grupposervizi.easy.openid.configuration.properties.path* must not be configured.

The above keys must match the customer OpenID Provider configuration. All of them are mandatory except for:

- *jwk.endpoint*: if this key is not set the CCH Tagetik application will not perform JWT validation using the JSON Web Key standard.
- *revoke.endpoint*: if this key is not set the CCH Tagetik application won't revoke the access and refresh tokens upon logout.
- *userinfo.endpoint*: if this key is not set the CCH Tagetik application won't call the *userinfo* endpoint for fetching user information but it will use the claims returned in the ID Token.
- *acr.values*: if set, CCH Tagetik will add the *acr_values* parameter to the authentication request.
- *username.claim*: if set, CCH Tagetik will use this claim to identify the username. You can use this property when you want to use a custom username different from "sub" claim. **When you set this property please ensure that must be unique, case insensitive and does not contain any special characters. If at least one of these conditions is not respected, the username will be "sub" or email claim.**

The above configuration is cached and invalidated every 5 minutes.

5.CCH Tagetik Authentication Configuration

This section describes how to configure the CCH Tagetik application to use OpenID Connect authentication.

Steps

1. Access the web interface with a CCH Tagetik user which has administrative privileges and access to the Repository.
2. Access the Administrative web interface. From the standard CCH Tagetik 5 web interface, it is necessary to access the Administrative interface by clicking on the “three dots” icon. See the following screenshot.

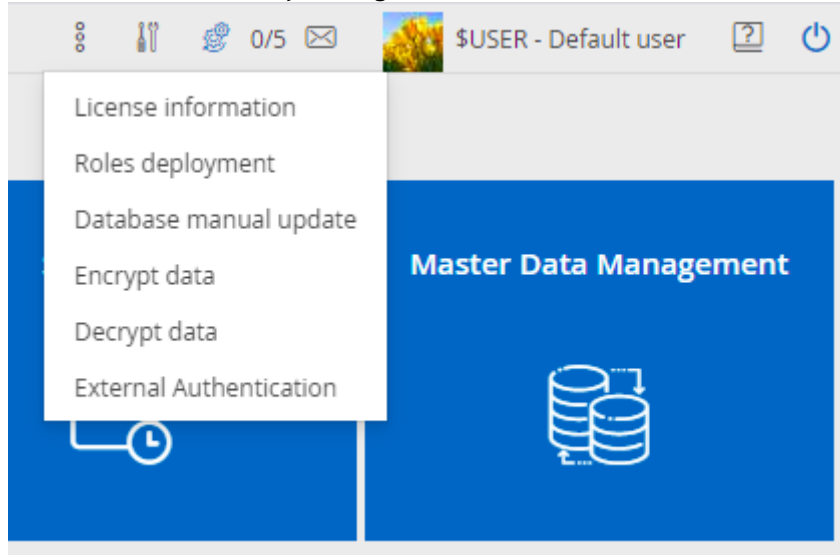


Figure 4 CCH Tagetik 5 Administrative Interface.

3. Then click **External Authentication**.
4. If toggle is disabled, enable toggle “Use Third Party Authentication”
5. Select “OpenID Connect Authentication” from the “Available authentication types” field.

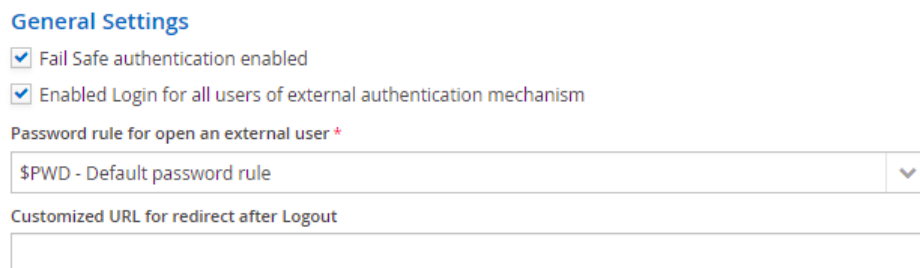
Figure 5 Configuration of OpenID Connect authentication in CCH Tagetik, authentication types.

6. Fill the General Settings fields and save the Authentication configuration by clicking on the “save” icon. After the above steps are completed, this instance of CCH Tagetik will be enabled for OpenID Connect authentication as long as the configuration of the properties file described above is completed correctly.

6. User management

In CCH Tagetik, external users, i.e. those users authenticated via external authorities such as Active Directory, initially, must be imported and assigned the appropriate roles and rights with respect to the data. For more information regarding the management of users, their roles and profiles, please refer to the User Management manual available from the CCH Tagetik web interface.

Otherwise, CCH Tagetik provides a parameter, “Enabled Login for all user of external authentication mechanism”, which when set permits authenticated users who are not already in the CCH Tagetik repository to be added automatically without roles or rights.



General Settings

☒ Fail Safe authentication enabled

☒ Enabled Login for all users of external authentication mechanism

Password rule for open an external user *

\$PWD - Default password rule

Customized URL for redirect after Logout

If "Enabled Login for all users of external authentication mechanism" is set to "true", then the user is automatically created in the Repository if the user is not already defined there.

A few notes:

- Important to remember that any user who is not authenticated via the OpenID Connect Provider, will not be permitted to access the application, even if the user has been configured in CCH Tagetik.
- Regarding the removal of inactive users, OpenID Connect does not permit the removal of a user from the CCH Tagetik Repository. Inactive user configurations must be removed from CCH Tagetik manually via the CCH Tagetik administrative web interface

6.1 Alternate Login Access

When OpenID Connect Authentication is enabled, the automatic mechanism that redirects unauthenticated requests to login form page is not available. To authenticate users with username and password with the configured Authentication modes, including the default authentication, it is necessary to connect directly to the URL of the CCH Tagetik application, i.e. `http://<server>:<port>/tagetikcpm/authenticate`, instead of the normal CCH Tagetik URL.

N.B. The usage of an alternative URL for the web interface is **not** available for the CCH Tagetik .NET Add-ins.

Therefore, users who are not configured to login via OpenID Connect or cannot do so, will not be able to login to CCH Tagetik using the CCH Tagetik .Net Add-ins.

6.2 External Authentication Configuration – OpenID

When OpenID Connect Authentication is correctly configured, it's possible to enable OpenID as external authentication provider via "External Authentication Configuration Interface". From Tagetik homepage, an admin, logged to Repository, can click on action menu, then "External Authentication" and land on External Authentication Configuration Interface, from this view he/she can click on top left toggle button to enable external authentication, then select OpenID as external provider from "Available authentication type" Combo Box. After OpenID is selected as external provider, all of its properties are showed in the right part of the view, all properties are in read only mode. Admin can now click on "Save" button to save configuration.

7. Glossary

The following table provides a short glossary for the terminology which are used in this manual.

AS	Application Server. This middleware software should be understood to mean JEE compliant application servers.
JEE	Java Platform Enterprise Edition
SSO	Single Sign On
OIDC	OpenID Connect
JWT	JSON Web Token
OP	OpenID Provider
RP	Relying Party
JWK	JSON Web Key
ID Token	OpenID specific token
Access Token	OAuth2 access token

8. Notes and Copyright

Tagetik Software S.r.l. with a sole quotaholder reserves the right to change the contents of this document without prior notice. All trademarks are the property of the trademark owners.

This document and its contents are the sole property of CCH Tagetik; and are protected by Italian Copyright Law no. 633/1941 and by any applicable international agreements and treaties concerning intellectual property rights. All rights, including the intellectual property rights, are reserved to Tagetik. The user of this document is authorized by Tagetik to make copies of this document for private purposes only in paper and/or electronic formats for the sole purpose of training in relation to the use of Tagetik's software for the benefit of the staff/employees of the above mentioned company. Any other use is hereby expressly forbidden.

CCH Tagetik

Tagetik Software srl.
Via Roosevelt, 103
55100 Lucca
Italy
tgk-info@wolterskluwer.com

wolterskluwer.com