

Content Manager

Software Version 25.2

OpenID Connect authentication

opentext™

Document Release Date: April 2025
Software Release Date: April 2025

Legal notices

Copyright 2008-2025 Open Text

The only warranties for products and services of Open Text and its affiliates and licensors ("Open Text") are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Except as specifically indicated otherwise, this document contains confidential information and a valid license is required for possession, use or copying. If this work is provided to the U.S. Government, consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Documentation updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

Support

Visit the [MySupport portal](#) to access contact information and details about the products, services, and support that OpenText offers.

This portal also provides customer self-solve capabilities. It gives you a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the MySupport portal to:

- View information about all services that Support offers
- Submit and track service requests
- Contact customer support
- Search for knowledge documents of interest
- View software vulnerability alerts
- Enter into discussions with other software customers
- Download software patches
- Manage software licenses, downloads, and support contracts

Many areas of the portal require you to sign in. If you need an account, you can create one when prompted to sign in.

Contents

Introduction	4
ADFS for Native client	5
Configure ADFS	5
Configure OpenID settings in Content Manager	12
ADFS for Web Client and Service API	14
Create the ADFS application	14
Add the settings to the Web Client	16
Configure ADFS for the Office integration access	16
Add Office integration to the settings of the Web Client	18
Azure AD for WebClient, Mobile App and Service API	19
Create the Azure AD application	19
Configure for mobile	21
Add the mobile redirect URI	21
Add the mobile redirect URI to the Service API	21
Configure authentication in hptrim.config	21
Configuring refresh interval for authentication	22
Enable redirect	22
Allow users (Web Client only)	23
Logout	23
Allow anonymous access in IIS	23
Azure AD for Content Manager desktop	24
Create the Azure AD application	24
Configure authentication in Content Manager Enterprise Studio	25
Configure Azure AD for Office integration access	26
Google authentication	28
Create the Google credentials	28
Configure authentication in hptrim.config	28
Enable redirect	29
Logout	29
Allow anonymous access in the IIS	29

Introduction

This document provides high level information for you to configure OpenID Connect authentication with Content Manager.

OAuth authentication is managed via OpenID Connect authentication. The authentication is configured in your Identity Provider (e.g. Azure AD) and then the appropriate details are stored in Content Manager in the **hptrim.config** file for the Web Client and Service API, and in Content Manager Enterprise Studio for the desktop client.

ADFS for Native client

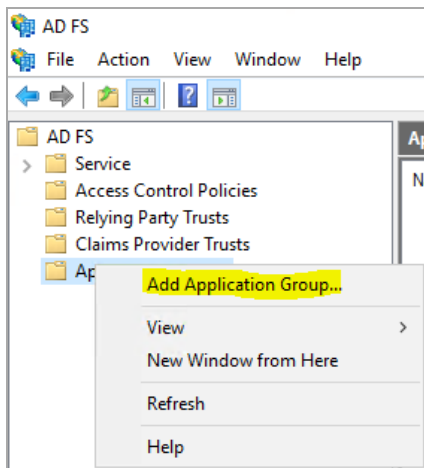
As of Content Manager 10 the Content Manager web applications (Service API, WebDrawer and Web Client) have an OpenID Connect authentication provider built in.

This section describes creating an ADFS application and configuring the Windows native Content Manager client.

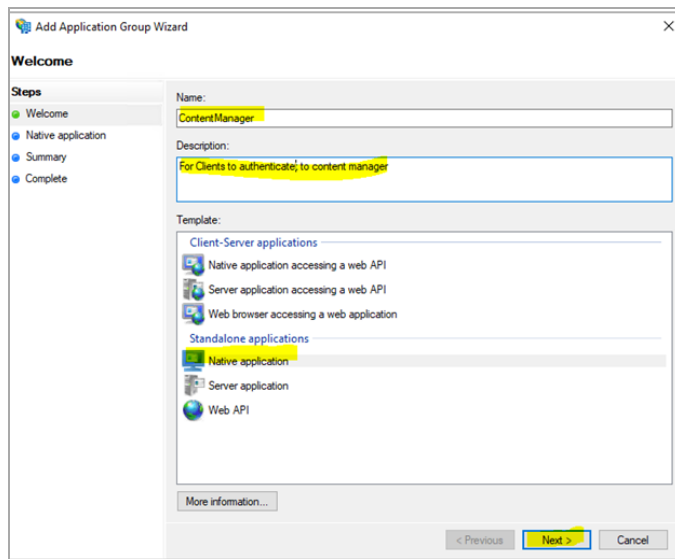
Configure ADFS

To setup ADFS to support OpenID for Content Manager native, perform the following steps:

1. Log on to your ADFS Server and create an application group.



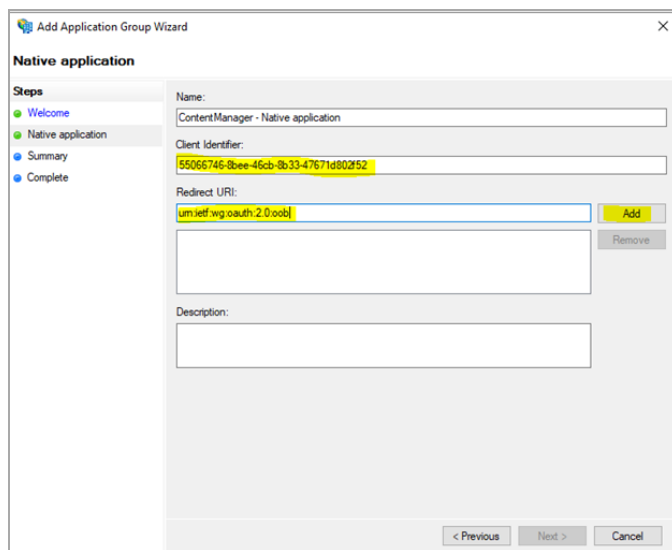
2. Enter the name and the description for the group, then select **Native application**. Click **Next**.



The 'Add Application Group Wizard' window is shown at the 'Welcome' step. The 'Name' field contains 'ContentManager'. The 'Description' field contains 'For Clients to authenticate to content manager'. The 'Template' section shows 'Client-Server applications' and 'Standalone applications'. Under 'Standalone applications', 'Native application' is selected. The 'Next >' button is highlighted.

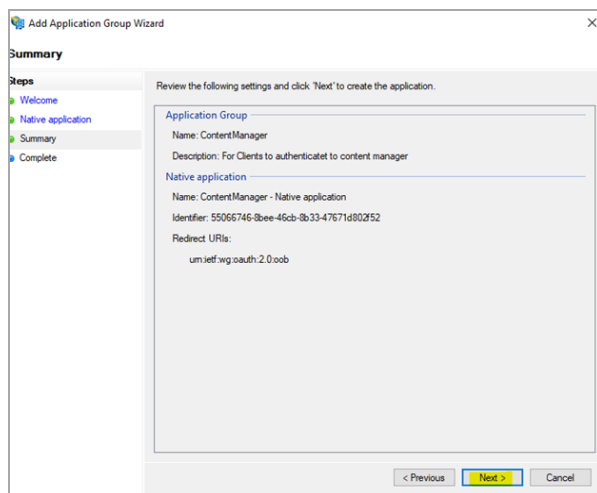
3. Configure the group.

Note the client identifier for later use and enter the value 'https://127.0.0.1' in Redirect URI, click **Add** and then click **Next**.



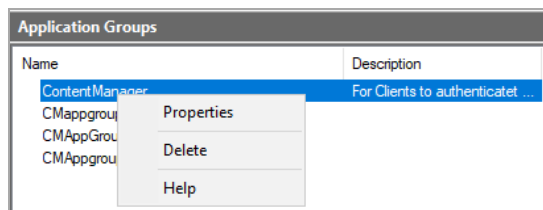
The 'Add Application Group Wizard' window is shown at the 'Native application' step. The 'Name' field contains 'ContentManager - Native application'. The 'Client Identifier' field contains '55066746-8bee-46cb-8b33-47671d80252'. The 'Redirect URI' field contains 'um:ietf-wg-oauth2.0.oob'. The 'Add' button is highlighted. The 'Description' field is empty. The 'Next >' button is highlighted.

4. Review the settings and click **Next**.

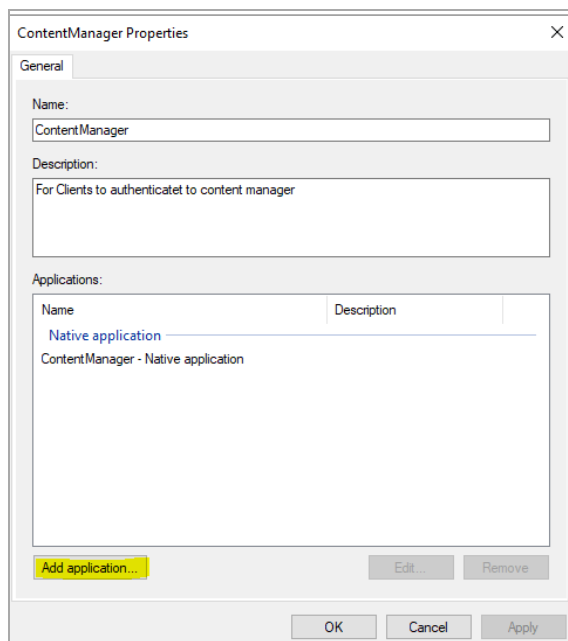


5. Modify application group properties.

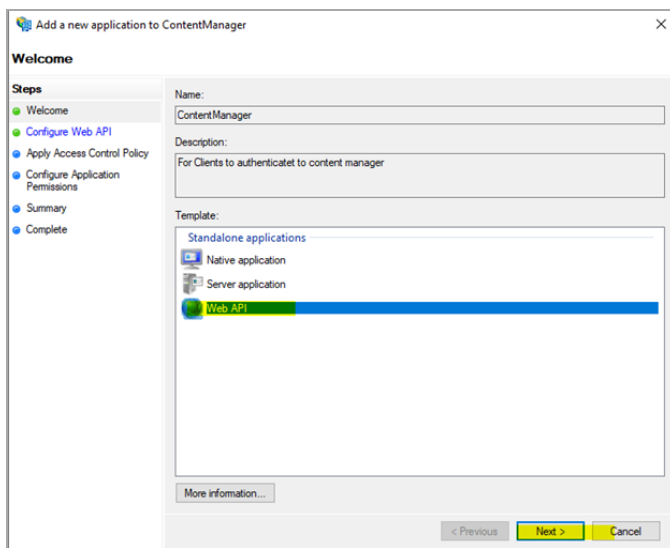
Once this has been completed, right-click the application group and select **Properties**.



6. Click **Add application**.

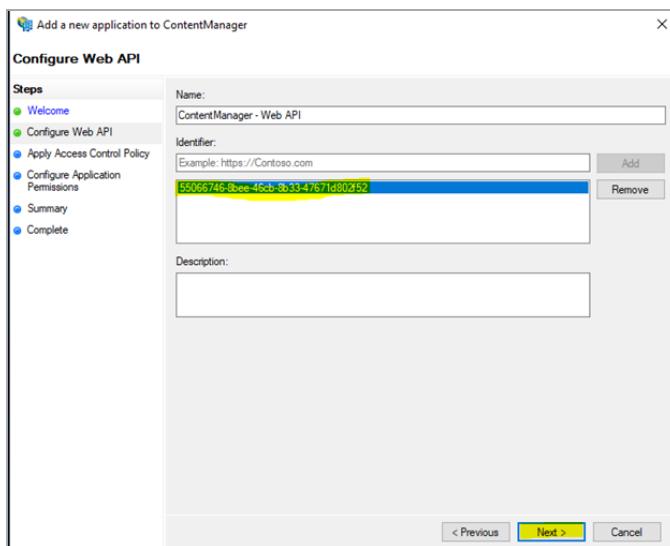


7. Select **Web API** as the **Template**.

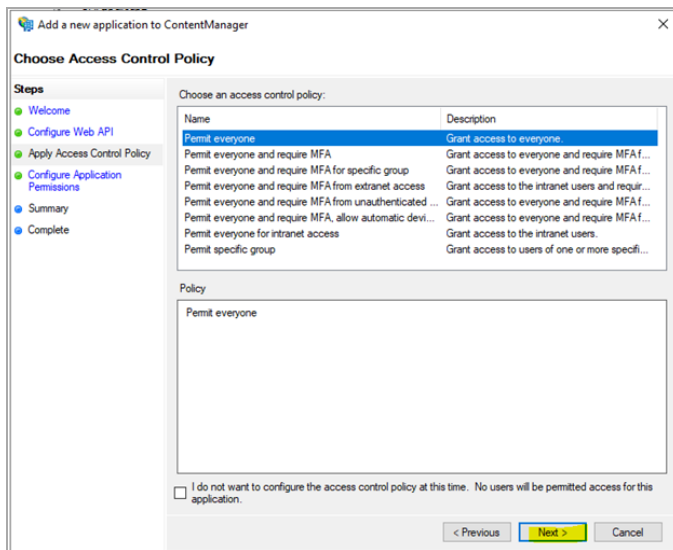


8. Configure the Web API.

Add the Redirect to be the same as the client ID of the Native application that you previously copied. Click **Next**.

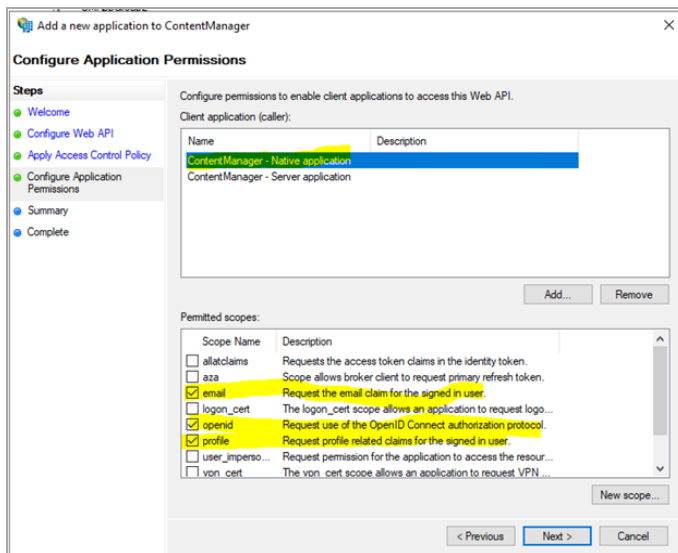


9. Select an appropriate Access Control Policy.

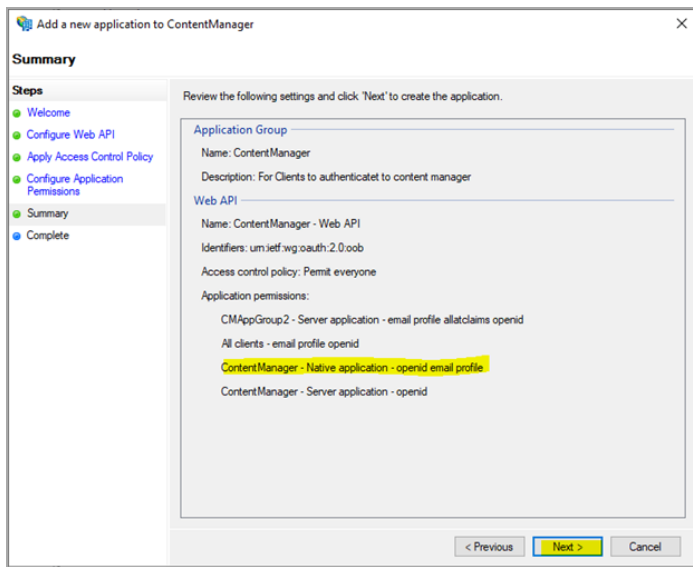


10. Set the scopes.

Select the Native Application and ensure **email**, **openid**, and **profile** are checked.

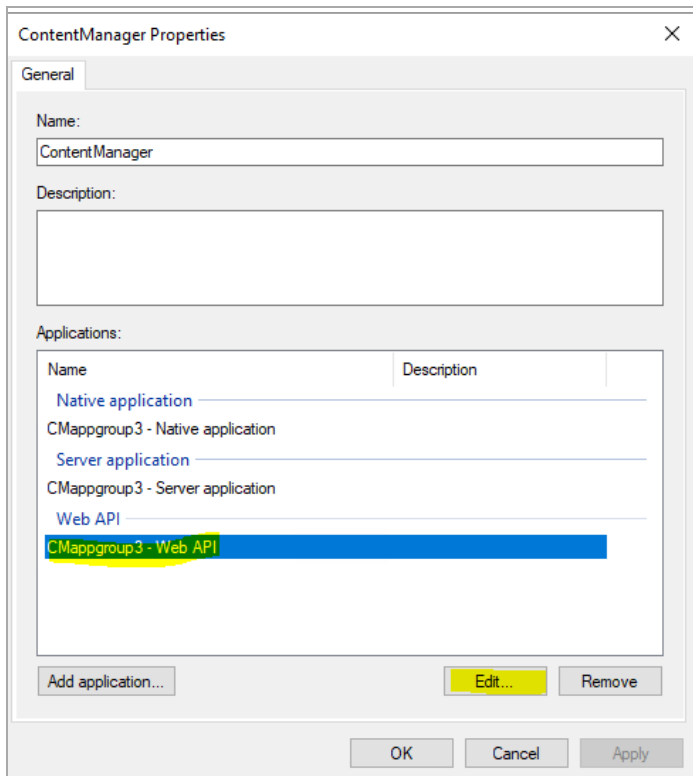


11. Review the configuration, click **Next** and then **Close**.



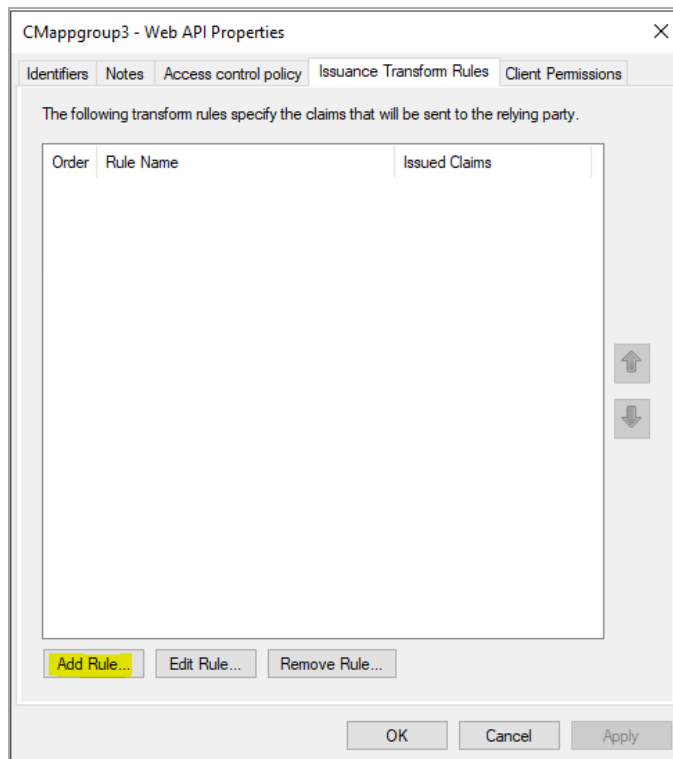
12. Set issuance rules.

Right-click the app group and select **Properties**, highlight the WEB API application and select **Edit**.



13. Add a rule.

Navigate to the **Issuance Transform Rules** Tab and click **Add Rule**.



14. Map attributes.

Give the rule a name, select **Active Directory** as the **Attribute Store**, and add the following mapping of LDAP attributes and click **OK**:

- **E-Mail-Addresses - E-mail Address**
- **User-Principal-Name - UPN**

Edit Rule - Claim rule

You can configure this rule to send the values of LDAP attributes as claims. Select an attribute store from which to extract LDAP attributes. Specify how the attributes will map to the outgoing claim types that will be issued from the rule.

Claim rule name:

Rule template: Send LDAP Attributes as Claims

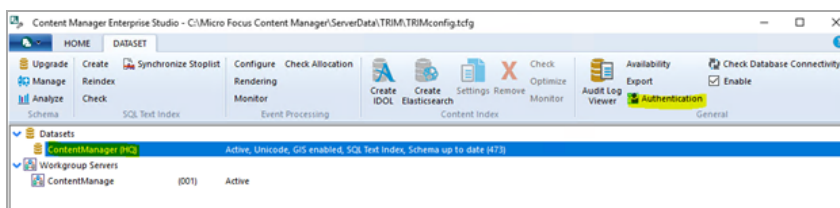
Attribute store:

Mapping of LDAP attributes to outgoing claim types:

	LDAP Attribute (Select or type to add more)	Outgoing Claim Type (Select or type to add more)
▶	E-Mail-Addresses	E-Mail Address
	User-Principal-Name	UPN
*		

Configure OpenID settings in Content Manager

1. Navigate to the Content Manager server and run the Content Manager Enterprise Studio as an Administrator. Right-click the dataset and select **Authentication**.



2. Enter the ADFS URL for the ADFS server, the client identifier which was noted earlier (Native Application) and click **Test Authentication**.

Configure Authentication Mechanisms for Dataset: ContentManager (HQ)

Windows OpenID Connect OneDrive Connect

☒ Enable OpenID Connect authentication

OAuth 2.0 Configuration

OpenID Issuer URL

Authorize Endpoint URL

Token Endpoint URL

Client ID

Client App Scope

Client Redirect URI

OpenID Connect Confidential Clients

Client ID	Scope
-----------	-------

3. Enter the user details for a test user and click **Sign in**.

Content Manager Third Party Authentication

Resource Domain

Sign in

Sign in as current user

© 2018 Microsoft

The users email in active directory should appear, if successful.

Content Manager

The user 'Admin@domain.com' has been successfully authenticated.

ADFS for Web Client and Service API

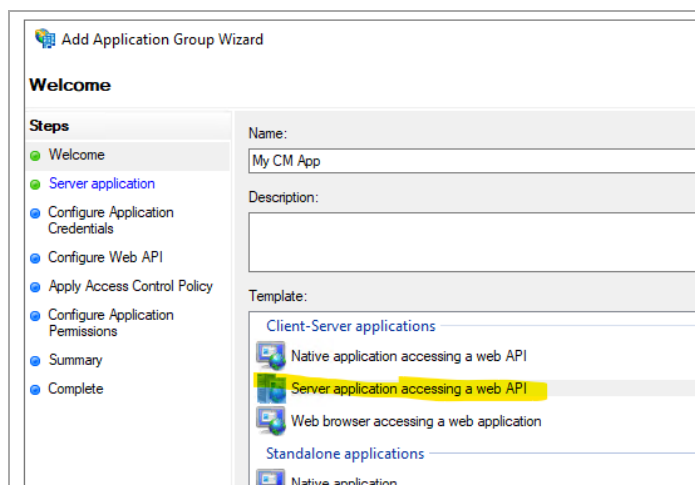
As of Content Manager 10 the web applications (Service API, WebDrawer and Web Client) have a built in OpenID Connect authentication provider.

This section describes creating an ADFS application and configuring the Content Manager web applications.

Create the ADFS application

To create the ADFS application,

1. Create a new **Application Group**.
2. Select **Server application accessing a web API** as the template.



3. Enter a name and click **Next**.

Note the client identifier.

4. Add a Redirect URI.

The redirect URI must be in lowercase and be the URL of the Content Manager web site with the suffix `/serviceapi/auth/openid`.

For example, `https://myserver/contentmanager/serviceapi/auth/openid`.

The screenshot shows the 'CM Web Client DC - Server application Properties' dialog box with the 'General' tab selected. The 'Name' field contains 'CM Web Client DC - Server application'. The 'Client Id' field contains '9eeda8c0-b3a2-43c5-aa45-34930f0dc0ff'. The 'Redirect URI' field contains 'https://auchurchla02/cm/serviceapi/auth/openid'. There are 'Add' and 'Remove' buttons next to the 'Redirect URI' field. The 'Description' field is empty.

5. Click **Next**.
6. Generate a shared secret and note the secret.

The screenshot shows the 'Configure Application Credentials' dialog box. On the left, there is a 'Steps' list with 'Configure Application Credentials' selected. The main area has two sections: 'Select credentials used by the application to authenticate itself with AD FS when requesting access tokens.' and 'Generate a shared secret'. The first section has two options: 'Register a key used to sign JSON Web Tokens for authentication' (unchecked) and 'Windows Integrated Authentication' (unchecked). The second section has 'Generate a shared secret' checked. Below this, the 'Secret' field contains 'bEa-ZY4\$NLwAW9mFoLDv23ax8sWso2bZwvlim_'. There is a 'Copy to clipboard' button next to the secret field. A yellow warning box at the bottom states: 'Copy and save the secret. You will not be able to view the secret after the application group is created. You can reset the secret later if required.'

7. Click **Next**.
8. Add an identifier. For example, `https://MyServer/contentmanager/`.

The screenshot shows the 'Add Application Group Wizard' dialog box, 'Configure Web API' step. On the left, there is a 'Steps' list with 'Configure Web API' selected. The main area has fields for 'Name' (My CM App - Web API), 'Identifier' (https://btpvm2237.myexch19.com/contentmanager/), and 'Description'. There is a 'Register' button next to the 'Identifier' field.

9. Click **Next**.
10. Choose an access control policy. For example, Give access to everyone.
11. In the **Configure Application permissions**, select **email**, **openid**, and **profile**.
12. Complete rest of the steps in the Application Group.

Add the settings to the Web Client

To configure the Web Client, edit the **hprmServiceApi.config** file and add (or edit) the authentication element to look similar to the example below.

- Client ID and secret (noted in the previous section)
- issuerUri is found in the ADFS console - Endpoints, in the OpenID Connect section.
- Name must be openid.

For example,

```
<authentication allowAnonymous="false"slidingSessionMinutes="30">
<openIdConnect>
<add
name="openid"
clientId="CLIENT_ID"
clientSecret="SECRET"
issuerURI="https://MyServer/adfs/.well-known/openid-configuration"/>
</openIdConnect>
</authentication>
```

Configure ADFS for the Office integration access

The office integration requires an access token to allow it to authenticate with the Web Client, this can be configured in ADFS.

1. Go to the **Application group** configured in the above section.
2. Click **Add Application** to add a native application.

Add a new application to CM Web Client Sample

Native application

Steps

- Welcome
- Native application
- Summary
- Complete

Name:
CM Web Client Sample - Native application 1

Client Identifier:
b93b7949-4715-4238-9a66-01b9663b4e75

Redirect URI:
Example: https://Contoso.com
https://myserver/contentmanager

Buttons: Add, Remove

Preserve the Client ID for later use.

3. Complete rest of the steps for the native application.
4. Edit the Web API application and in the **Client Permissions**, add the new client application selecting the **scopes, email, openid, and profile**.

The screenshot shows the 'Client Permissions' tab of the 'CM Web Client Sample - Web API Properties' dialog. It contains a table of client applications and a list of permitted scopes.

Name	Description
CM Web Client Sample - Server application	
CM Web Client Sample - Native application	

Buttons: Add..., Remove

Permitted scopes:

Scope Name	Description
<input type="checkbox"/> allatclaims	Requests the access token claims in the identity token.
<input type="checkbox"/> aza	Scope allows broker client to request primary refresh token.
<input checked="" type="checkbox"/> email	Request the email claim for the signed in user.
<input type="checkbox"/> logon_cert	The logon_cert scope allows an application to request logo...
<input checked="" type="checkbox"/> openid	Request use of the OpenID Connect authorization protocol.
<input checked="" type="checkbox"/> profile	Request profile related claims for the signed in user.

- 5. In the **Issuance Transform Rules**, add a new Rule.
- 6. Select the **Send LDAP Attributes** and click **Next**.
- 7. Choose the **Active Directory** as the attribute store.
- 8. Map the following two claims:
 - **Display Name - Name**
 - **User-Principal-Name - UPN**

The screenshot shows the configuration for the 'Send LDAP Attributes as Claims' rule. It includes fields for the claim rule name, attribute store, and a mapping table.

Claim rule name: MyRule

Rule template: Send LDAP Attributes as Claims

Attribute store: Active Directory

Mapping of LDAP attributes to outgoing claim types:

	LDAP Attribute (Select or type to add more)	Outgoing Claim Type (Select or type to add more)
▶	Display-Name	Name
	User-Principal-Name	UPN
*		

- 9. Click **Finish**.

Add Office integration to the settings of the Web Client

In the Content Manager Workgroup Server, navigate to the installation folder, for example **C:\Program Files\Micro Focus\Content Manager\Web Client\ADFS**, and update the **config.xml** as shown below:

- clientAuthority - ADFS authority (https://{adfsdomain}/adfs)
- clientResourceUri - openid (scope of authentication)
- clientID - Native application ClientID
- clientReturnUri – Web Client url

For example,

```
<adfsClient>  
<clientAuthority>https://test.com/adfs</clientAuthority>  
<clientResourceUri>openid</clientResourceUri>  
<clientID>ab999999-999d-9aeb-a999-999b99999a99</clientID>  
<clientReturnUri>https://test.com/contentmanager/</clientReturnUri>  
</adfsClient>
```

Azure AD for WebClient, Mobile App and Service API

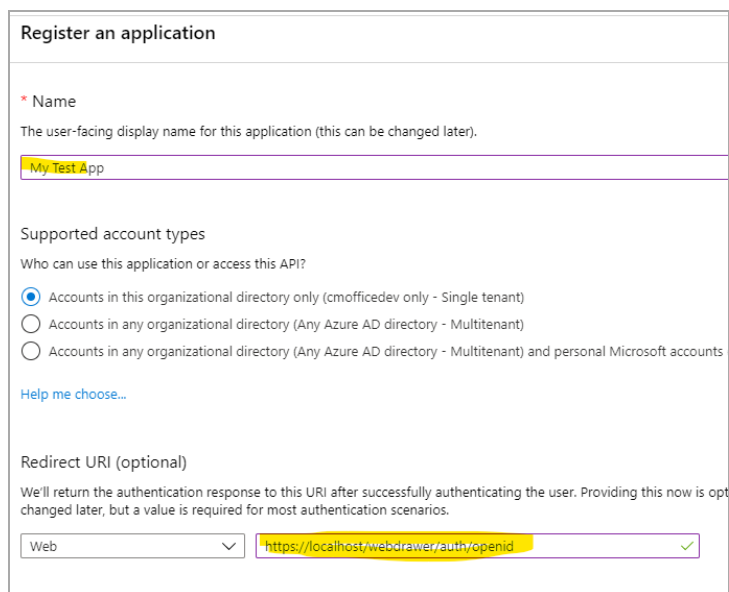
As of Content Manager 10 the web applications (Service API, WebDrawer and Web Client) have a built in OpenID Connect authentication provider.

This section describes creating an Azure AD application and configuring the web application. This document also provides detailed steps required to allow the Content Manager Mobile App to authenticate.

Create the Azure AD application

To create the Azure AD application:

1. From the portal.azure.com, go to Azure AD.
2. Go to **App Registrations** and select **New Registration**.



The screenshot shows the 'Register an application' form in the Azure portal. The form has a title bar 'Register an application'. Below it, there is a section for 'Name' with a red asterisk indicating it is required. The text below says 'The user-facing display name for this application (this can be changed later)'. The input field contains 'My Test App'. Below this is a section for 'Supported account types' with the text 'Who can use this application or access this API?'. There are three radio button options: 'Accounts in this organizational directory only (cmofficedev only - Single tenant)' (selected), 'Accounts in any organizational directory (Any Azure AD directory - Multitenant)', and 'Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts'. There is a link 'Help me choose...'. Below this is a section for 'Redirect URI (optional)' with the text 'We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional, but a value is required for most authentication scenarios.' There is a dropdown menu set to 'Web' and an input field containing 'https://localhost/webdrawer/auth/openid' with a green checkmark icon.

3. Enter a name.
4. Under Redirect URI leave **Web** selected.

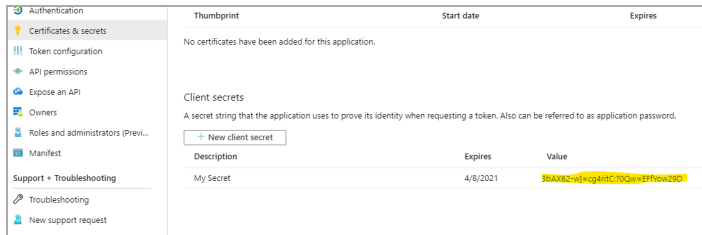
The value in the Redirect URI is important. It must be lowercase and must be the URL to your application. For example, <https://mydomain.com/cmwebdrawer> followed by the path to the authentication provider, for example, `/auth/openid`.

The `/auth/` component is fixed but the 'openid' is the name you will supply in **hptrim.config** later and so can be any string, as long as it matches the value in **hptrim.config**.

For the Web Client the path must include the path to the Service API, for example,
<https://mydomain.com/contentManager/serviceapi/auth/openid>.

5. Add a secret.

From **Certificates and Secrets**, add a secret and note the secret.



6. Configure Tokens.

From the **Authentication** section, select the tokens to be issued and check the option **ID Tokens**.

7. Configure permissions.

From the **API Permissions**, add the following delegated Microsoft Graph permissions:

- email
- offline_access
- openid
- profile
- User.Read
- Files.Read
- Files.Read.All
- Files.ReadWrite
- Sites.Read.All
- Sites.ReadWrite.All
- User.Read

Select **Grant admin consent** to grant access to all permissions.

+ Add a permission ✓ Grant admin consent for cmtrunk			
API / Permissions name	Type	Description	Admin consent re
▼ Microsoft Graph (11)			
email	Delegated	View users' email address	No
Files.Read	Delegated	Read user files	No
Files.Read.All	Application	Read files in all site collections	Yes
Files.ReadWrite	Delegated	Have full access to user files	No
Files.ReadWrite.All	Delegated	Have full access to all files user can access	No
offline_access	Delegated	Maintain access to data you have given it access to	No
openid	Delegated	Sign users in	No
profile	Delegated	View users' basic profile	No
Sites.Read.All	Delegated	Read items in all site collections	No
Sites.ReadWrite.All	Delegated	Edit or delete items in all site collections	No
User.Read	Delegated	Sign in and read user profile	No

Configure for mobile

Add the mobile redirect URI

If this Azure app is to be used to connect to the Content Manager Mobile App then,

1. Go to the **Authentication** section and select **Add a platform**.
2. Choose **Mobile and desktop applications** and enter `trimapp://mobile` in the **Custom Redirect URIs** field.

Add the mobile redirect URI to the Service API

The redirect URI set above must be included in the Service API `hptrim.config`. To do this, add the **authentication** and **openIdConnect** elements.

For example,

```
<authentication allowAnonymous="false"slidingSessionMinutes="60">
<openIdConnect>
<addname="mobile"
clientId="[CLIENT ID from Azure App]"
clientSecret="[SECRET from Azure APP]"
issuerURI="[OpenID Connect metadata document from endpoints in Azure App]"
redirectUri="trimapp://mobile" />
</openIdConnect>
</authentication/>
```

Configure authentication in hptrim.config

To use the Azure AD app created above, edit the `hptrim.config` (or `hprmServiceAPI.config` in the Web Client) so that it has an authentication similar to the one below:

1. The name must match the last segment of the Redirect URI path.
2. Client ID is the application ID from the Azure AD Overview.
3. The secret is the one saved when creating the App. If it was not saved, created a new one in Certificates and Secrets.
4. Get the issuerUri from **Overview > Endpoints > OpenID Connect metadata document**.

For example,

```
<authentication allowAnonymous="false"slidingSessionMinutes="60">
<openIdConnect>
<add
name="openid"
clientId="ae99999d-99e9-9ecc-b9eb-99d9d999dd"
clientSecret="_MqXXXXXXXXXXXXXG[sp3GrMfD:"
issuerURI="https://login.microsoftonline.com/09999ee9-9999-9999-9d9a-
999999999/v2.0/.well-known/openid-configuration"/>
</openIdConnect>
</authentication>
```

Configuring refresh interval for authentication

The **refreshInterval** configuration setting lets users define the OpenID Connect keyset refresh frequency in minutes (e.g., **refreshInterval**=60 for 60 minutes), resolving key ID (kid) rollover issues.

To configure refresh interval, edit the **hptrim.config** (or **hprmServiceAPI.config** in the Web Client) and add the **refreshInterval** configuration attribute to the **<authentication >** parameter.

For example,

```
<authentication corsAllowedOrigins="https://localhost" allowAnonymous="false"
slidingSessionMinutes="30" redirectURI="" refreshInterval="20">
...
...
...
</authentication>
```

Enable redirect

The Web Client will not re-direct the authentication endpoint unless the Html feature is enabled in **hprmServiceAPI.config**, perform the following steps:

1. Edit the **hprmServiceAPI.config** file.
2. Find the property named **serviceFeatures**.
3. Add the feature **Html**.

Allow users (Web Client only)

For the Web Client, find the **web.config** file and find the **Location** element with the path **serviceapi** in the **web.config** file. It should contain the element `<allow users="*" />` within its authorization element.

For example,

```
<location path="serviceapi">
<system.web>
<httpHandlers>
<add path="*" type="ServiceStack.WebHost.Endpoints.ServiceStackHttpHandlerFactory,
ServiceStack" verb="*" />
</httpHandlers>
<authorization>
<allow users="*" />
</authorization>
</system.web>
...
</location>
```

Logout

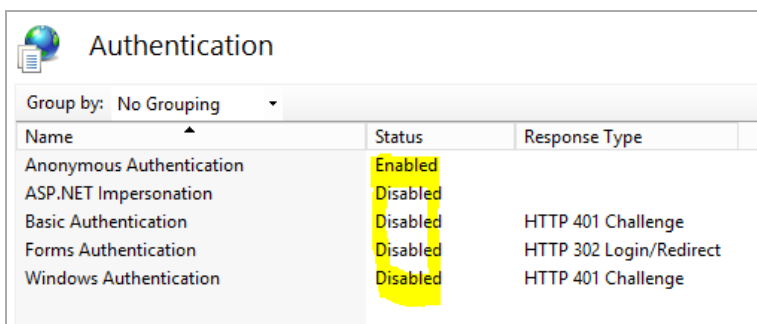
For WebDrawer the logout link is configured in the uiSettings. It should contain `~/auth/logout`. In the Web Client a logout link will be displayed automatically when OpenID Connect authentication is enabled.

For example,

```
<uiSettings
logoutLink="~/auth/logout"
...
/>
```

Allow anonymous access in IIS

The IIS will not handle authentication, so use IIS Manager to allow anonymous access only.

A screenshot of the IIS Manager 'Authentication' feature view. The 'Group by' dropdown is set to 'No Grouping'. A table lists various authentication methods with their status and response types. The 'Status' column for 'Anonymous Authentication' is highlighted in yellow and shows 'Enabled', while all other methods are 'Disabled'.

Name	Status	Response Type
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Disabled	HTTP 401 Challenge

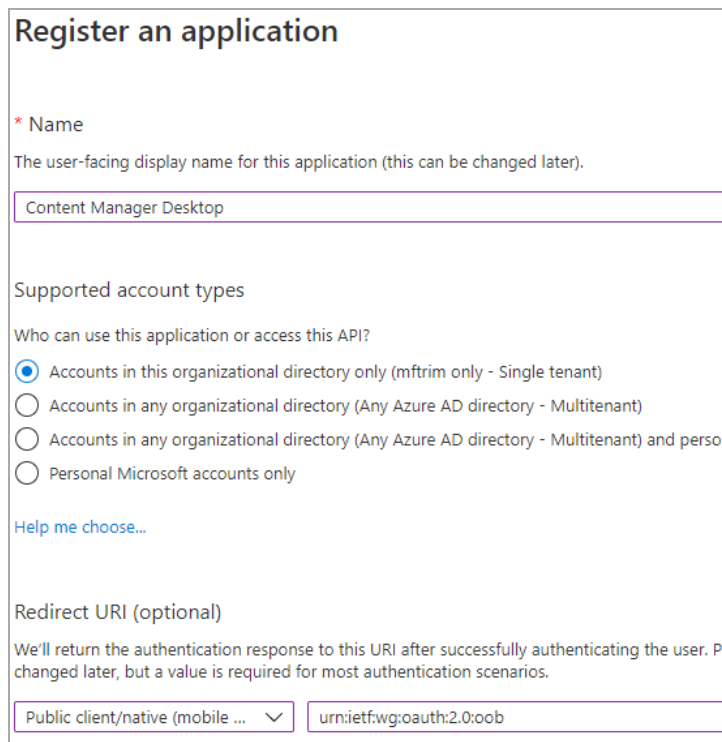
Azure AD for Content Manager desktop

OpenID Connect may be used to authenticate with the Content Manager desktop client, this section describes how to configure this.

Create the Azure AD application

To create the Azure AD application:

1. From the portal.azure.com, go to Azure AD.
2. Go to **App Registrations** and select **New Registration**.



The screenshot shows the 'Register an application' form in the Azure portal. The form has the following sections:

- Name:** A text input field with the value 'Content Manager Desktop'. Above the field is a red asterisk and the word 'Name'. Below the field is the text 'The user-facing display name for this application (this can be changed later)'.
- Supported account types:** A section titled 'Who can use this application or access this API?' with four radio button options:
 - ☒ Accounts in this organizational directory only (mftrim only - Single tenant)
 - ☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant)
 - ☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts
 - ☐ Personal Microsoft accounts onlyA link 'Help me choose...' is located below the options.
- Redirect URI (optional):** A section with the text 'We'll return the authentication response to this URI after successfully authenticating the user. This can be changed later, but a value is required for most authentication scenarios.' Below this text are two input fields:
 - A dropdown menu with the selected value 'Public client/native (mobile ...' and a downward arrow.
 - A text input field with the value 'urn:ietf:wg:oauth:2.0:oob'.

3. Enter a name.
4. Select **Public native client** in the Redirect URI.
5. Enter the redirect Uri as 'urn:ietf:wg:oauth:2.0:oob'.
6. Configure permissions.

From the **API Permissions**, add the following delegated Microsoft Graph permissions:

- email
- offline_access

- openid
- profile
- User.Read
- Files.Read
- Files.Read.All
- Files.ReadWrite
- Sites.Read.All
- Sites.ReadWrite.All
- User.Read

Select **Grant Admin Access** to grant access to all permissions.

API / Permissions name	Type	Description	Admin consent
▼ Microsoft Graph (11)			
email	Delegated	View users' email address	No
Files.Read	Delegated	Read user files	No
Files.Read.All	Delegated	Read all files that user can access	No
Files.ReadWrite	Delegated	Have full access to user files	No
Files.ReadWrite.All	Delegated	Have full access to all files user can access	No
offline_access	Delegated	Maintain access to data you have given it access to	No
openid	Delegated	Sign users in	No
profile	Delegated	View users' basic profile	No
Sites.Read.All	Delegated	Read items in all site collections	No
Sites.ReadWrite.All	Delegated	Edit or delete items in all site collections	No
User.Read	Delegated	Sign in and read user profile	No

Configure authentication in Content Manager Enterprise Studio

To use the Azure AD app created above, open the Content Manager Enterprise Studio and perform the following:

1. From the database, select **Authentication > OpenID**.
2. The OpenID Issuer URL is taken from the **Azure App - Overview > Endpoints > OpenID Connect metadata document**.
3. The Client ID is taken from the **Azure App - Overview > Application ID**.
4. Client secret should be empty.

5. Client app scope should contain **openid email offline_access**.

Properties - DB244_14

Identification Connection Document Storage Options

Authentication Microsoft 365 Google Drive Bulk Load Storage

☒ Enable Integrated Windows authentication (Active Directory)

☐ Enable Explicit Windows authentication

☒ Enable OpenID Connect authentication

OpenID Issuer URL

Client ID

Set Client Secret

Client Scope

openid email office_access

Identity Claim

email

Redirect URL

https://127.0.0.1

Test Authentication

OpenID Connect Confidential Clients

Client ID	Scope

Add Remove Test

OK Cancel Help

Configure Azure AD for Office integration access

The office integration requires an access token to allow it to authenticate with the Web Client, this can be configured in Azure AD:

1. Create an Azure App for Web Authentication, you may use the one you created to authenticate with the Content Manager Web Client.
2. Create (or edit) the file ADFS\config.xml in your Content Manager Web Client installation folder and set it as follows:
 - **clientAuthority** - for single tenant application, use <https://login.microsoftonline.com/{TENANTID}>. For multiple tenant application, use <https://login.microsoftonline.com/common>.
 - **clientResourceUri** - use the scope (**access_as_user**) uri as resource uri. For example, api://test.domain.com/5a93xxx8-07d5-43x9-82e0-d90025ee6xx3/access_as_user.
 - **clientId** - the Application ID from your Content Manager Desktop Azure App.
 - **clientReturnUri** - urn:ietf:wg:oauth:2.0:oob
3. From the Azure App for Web Authentication,

- a. Go to **Expose an API** and select **Add a Scope**.
 - b. Enter the following values:
 - **Scope name:** access_as_user
 - **Who can consent:** Admins and Users
 - **Admin consent display name:** Office can act as the user
 - **Admin consent description:** Enable Office to call the add-in's web APIs with the same rights as the current user
 - **User consent display name:** Office can act as you
 - **User consent description:** Enable Office to call the add-in's web APIs with the same rights that you have
 - c. Click **Add scope**.
4. From the Content Manager Desktop Azure App, go to **API Permissions** and perform the following:
- a. Select **Add a Permission** and choose **My APIs**.
 - b. Select the Web Client Application.
 - c. Select the **access_as_user permissions**.
 - d. Select **Add permission**.
5. The Web Client authentication information needs to be updated to be made aware of the new client, perform the following steps:
- a. From the Content Manager Desktop Azure App, copy the **Application ID URI**. For example, api://cf2501bd-19f9-4ad6-96dc-f5cf7b2b3bf9.
 - b. Edit the Web Client **hprmServiceAPI.config** file.
 - c. In the **add** element of the openIDConnect element, add a new attribute called **appIdURI**. This is a case sensitive name.
 - d. The value of **appIdURI** should be the **Application ID URI** from the Content Manager Desktop Azure App.

For example,

```
<adfsClient>
<clientAuthority>https://login.microsoftonline.com/09a67e4a-46xx-485d-929a-
1add8eexxxb</clientAuthority>
<clientResourceUri>api://test.domain.com/5a93xxx8-07d5-43x9-82e0-
d90025ee6xx3/access_as_user</clientResourceUri>
<clientId>5a93xxx8-07d5-43x9-82e0-d90025ee6xx3</clientId>
<clientReturnUri>urn:ietf:wg:oauth:2.0:oob</clientReturnUri>
</adfsClient>
```

Google authentication

As of Content Manager 10 the web applications (Service API, WebDrawer and Web Client) have a built in OpenID Connect authentication provider.

This section describes creating Google credentials and configuring the web application.

Create the Google credentials

To create the Google credentials:

1. Go to <https://console.developers.google.com/>.
2. Select **Credentials > OAuth Client ID**.
3. Set Application type as **Web Application**.
4. Add your domain in the **Authorized JavaScript origins**.

The value in the Redirect URI is important. It must be lowercase and must be the URL to your application. For example, <https://mydomain.com/cmwebdrawer> followed by the path to the authentication provider, for example, `/auth/openid`.

The `/auth/` component is fixed but the 'openid' is the name you will supply in **hptrim.config** later and so can be any string, as long as it matches the value in **hptrim.config**.

For the Web Client the path must include the path to the Service API, for example, <https://mydomain.com/contentManager/serviceapi/auth/openid>.

5. On saving, the Client ID and Client Secret will be displayed, note them for later use.

Configure authentication in hptrim.config

To use the Google credentials created above, edit the **hptrim.config** (or **hprmServiceAPI.config** in the Web Client) so that it has an authentication as follows:

1. The name must match the last segment of the Redirect URI path.
2. Client ID and secret as noted in above section.
3. The issuerURI is: <https://accounts.google.com>

For example,

```
<authentication allowAnonymous="false" slidingSessionMinutes="2">
<openidConnect>
<addname="openid"
clientID="999999999999-abcdefghijklmemopqrs.apps.googleusercontent.com"
clientSecret="j1-BiX7685hjgf99999y"
issuerURI="https://accounts.google.com"
/>
```

```
</openIdConnect>  
</authentication>
```

Enable redirect

The Web Client will not re-direct the authentication endpoint unless the **Html** feature is enabled in **hprmServiceAPI.config**. To do this:

1. Edit **hprmServiceAPI.config**
2. Find the property named **serviceFeatures**.
3. Add the feature **Html**.

Logout

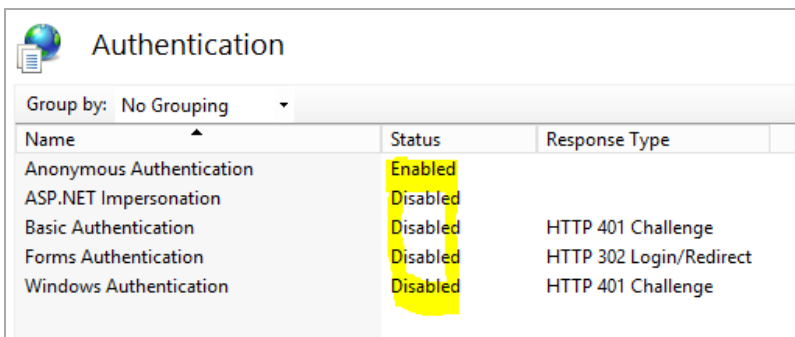
For WebDrawer the logout link is configured in the uiSettings. It should contain ~/auth/logout. In the Web Client a logout link will be displayed automatically when OpenID Connect authentication is enabled.

For example,

```
<uiSettings  
logoutLink="~/auth/logout"  
...  
>
```

Allow anonymous access in the IIS

The IIS will not handle authentication, so use IIS Manager to allow anonymous access only.

The screenshot shows the 'Authentication' section of the IIS Manager console. It features a table with columns for 'Name', 'Status', and 'Response Type'. The 'Status' column for all listed authentication methods is highlighted in yellow. The methods listed are Anonymous Authentication (Enabled), ASP.NET Impersonation (Disabled), Basic Authentication (Disabled), Forms Authentication (Disabled), and Windows Authentication (Disabled).

Name	Status	Response Type
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Disabled	HTTP 401 Challenge