OpenID Authentication

The OpenID feature gives users the option to authenticate to File Catalyst TransferAgent using Okta server.

This guide will instruct you on how to set up OpenID authentication with FileCatalyst TransferAgent.

Required Software

You will need the latest version of the following FileCatalyst products.

1. FileCatalyst TransferAgent Deployment package v3.8.5+
2. FileCatalyst Server v3.8.5+
3. FileCatalyst Transfer Agent v3.8.5+

> **NOTE:** The FileCatalyst Deployment package includes the FileCatalyst Transfer Agent. When you run the deployment package, it will ask you to download the FileCatalyst Transfer Agent if it is not already installed.

Overview

There are three main steps to configuring FileCatalyst to use OpenID:

1. Set up an Okta Server application.
2. Add required configuration settings to your **loginconfig.js** file.
3. Add required configuration settings to your **fcconf.conf** file.

> **NOTE:**
> Before making changes to your configuration files, you must install or upgrade your instance of FileCatalyst Server and FileCatalyst TransferAgent Deployment.

Okta Server Setup

To use OpenID with FileCatalyst, you'll need to create a new Okta application or configure an existing one to use OpenID.

> **NOTE:**
> You need to know the domain name/URL where your TransferAgent Deployment package is deployed. For example: `https://[YOUR DOMAIN]/filecatalyst`

Create A New Okta Application

To create a new Okta application, login to your Okta Developer Portal as an administrator.

On the Applications screen, select the option to create a new application.

Select the following options when creating the new application:

- **Sign-in method**

  Select **OIDC - OpenID Connect.**

- **Application type**

  Select **Single-Page Application.**

- **Grant type**

  Ensure that both **Authorization code** and **Refresh Token** boxes are selected.

- **Sign-in redirect URI**

  This is where the user will be taken to after logging in. The recommended value is:

```
https://[YOUR DOMAIN]/filecatalyst/express/express-src/fclogin.html
```

- **Sign-out redirect URI**

  This is where the user will be taken to after signing out. Set this to be the same value as the sign-in redirect URI.

- **Assignments**

  These settings can be configured at your discretion. However, users must be assigned to a group before they can authenticate with Okta.

  1. **Allow everyone in your organization to access**

     This option will automatically create a group for all users. This is the simplest choice for most companies when prototyping with a sample Okta server.

  2. **Limit access to selected groups**

     This option requires you to select a previously-created group to grant access permission. This option is usually best for companies who want to test the Open ID feature using an existing Okta server. In this case, you may want to create a custom group that contains test users.

  3. **Skip group assignments for now**

     This option will allow you to skip the group creation process. However, you must eventually create a group with assigned users for them to authenticate with Okta.

> **NOTE:** Selecting this option may put your configuration into an error state.

- **Trusted origins**

  Set this value to be the root of your URL.

  > **NOTE:** For example: `https://[YOUR DOMAIN]/`

After you have configured the above server settings, save and activate the new application.

> **NOTE:** Take note of the ClientID of the application you just created.

Add A New Trusted Origin

After creating a new Okta application with the above configurations, add a new trusted origin by completing the following steps:

- Go to the API page under “Security."
- Select the Trusted Origins tab.
- Add a new trusted origin. Enter the URL of the web server where the deployment package will be deployed as the "Origin URL" value. For example: `https://[YOUR DOMAIN].`

loginconfig.js (TA Deployment)

To use the OpenID feature in FileCatalyst, you'll need to configure the **OAuthConfigs** object in the **loginconfig.js** file which is part of the TA Deployment.

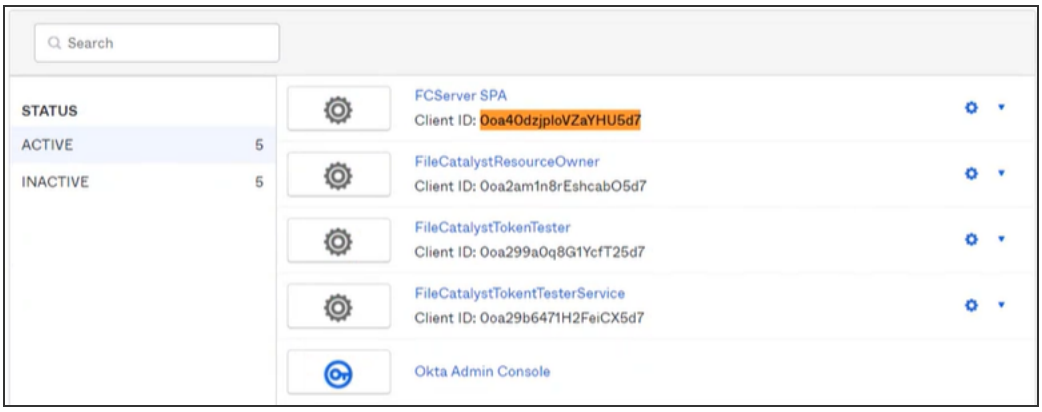
The **loginconfig.js** file is located here:

```
[TA Deployment Install]\express\express-src\loginConfig.js
```

In loginconfig.js, you'll need to configure the **OAuthConfigs** object with the following fields and values:

- **displayName**
  - The name of the OAuth button. You can leave this property empty.
- **buttonLabel**
  - The text displayed on the OAuth login button.
- **buttonClass**
  - Here you can add CSS classes to customize the look of the OAuth button.
- **OAuthType**

    - This is the name of your Okta authentication server. Set the value of this field to “Okta”. Okta is currently the only supported OpenID provider.
- **issuer**
    - The URL of the Okta authentication server.
    - You can find your authorization server in Okta under **API > Security**.
    - If you are using the default authentication server, the issuerURI is `https://[YOUR OKTA INSTANCE].okta.com/oauth2/default`
    - You will need to enter this same value for the IssuerURI in your **fcconf.conf** file.
- **clientId**
    - This is the Client ID value of the Okta application you created. Later, you will need to enter the same value for clientID in the **fcconf.conf** file.



- **redirectUri**
    - This is the URI where the user is redirected after using Okta to log into FileCatalyst.
    - This should be the value of the sign-in redirect URI that you entered while creating the application on the Okta server. It should be pointing to the **fclogin.html** page in the express-src directory.
- **signoutRedirectUri**
    - URL where the user is redirected after logging out.
    - This should be the same value as the redirectURI.
- **refreshUri**
    - The value for the refreshURI is the "token" endpoint of your Okta authorization server. It is in the format of `{issuerURI}/v1/token`.
    - So, if the issuerURI that you entered is `https://[YOUR OKTA INSTANCE].okta.com/oauth2/default` then the value for the refreshURI should be `https://[YOUR OKTA INSTANCE].okta.com/oauth2/default/v1/token`

Below is an example of the OAuthConfigs object:

```
const OAuthConfigs = [
```

```
  {
    displayName: "",
    buttonLabel: "Log in with OpenID",
    buttonClass: "btn btn-success btn-block", // change to suit your custom css or framework
    OAuthType: "Okta",
    issuer: "https://dev-11111111.okta.com/oauth2/default",
    // OpenID Connect APP Client ID
    clientId: "0oa4h605inRVIwA8x5d7",
    // Trusted Origin Redirect URI
    redirectUri: "https://[YOUR DOMAIN]/filecatalyst/express/express-src/fclogin.html ",
    signoutRedirectUri:
      "https://YOURDOMAIN/filecatalyst/express/express-src/fclogin.html ",
    refreshUri: "https://dev-11111111.okta.com/oauth2/default/v1/token",
  },
];
```

remoteNodes

You need to fill out the values for the **remoteNodes** object in the **loginConfig.js** file. You will notice that similar field exists in the **configuration.js** file located in the "js" directory in the root of your FileCatalyst deployment.

If you are upgrading from an existing deployment and you did *not* already provide credentials in the **configuration.js** file, then you can simply copy the values from **configuration.js**. The values provided in **loginConfig.js** take precedence.

> **NOTE:** If you do have credentials provided in **configuration.js** then the user will not be able to sign in with OpenID or be able to provide their own credentials.

At minimum, you must provide the value for the **remoteServer** field which is the IP address or hostname of the FileCatalyst Server that you set up.

For more information on theses values please read section titled **Remote Node Configuration (pg.config.remoteNodes)** of the of the Deployment Guide that is packaged with the FileCatalyst TransferAgent.

fcconf.conf (FileCatalyst Server)

> **NOTE:** You will need to restart FileCatalyst Server after making the following changes.

To use the OpenID feature in FileCatalyst Server, you'll need to add the **FCServer.server.config.auth.openID** property to the **fcconf.conf** file which belongs to the FileCatalyst Server.

The **fcconf.conf** file is located here:

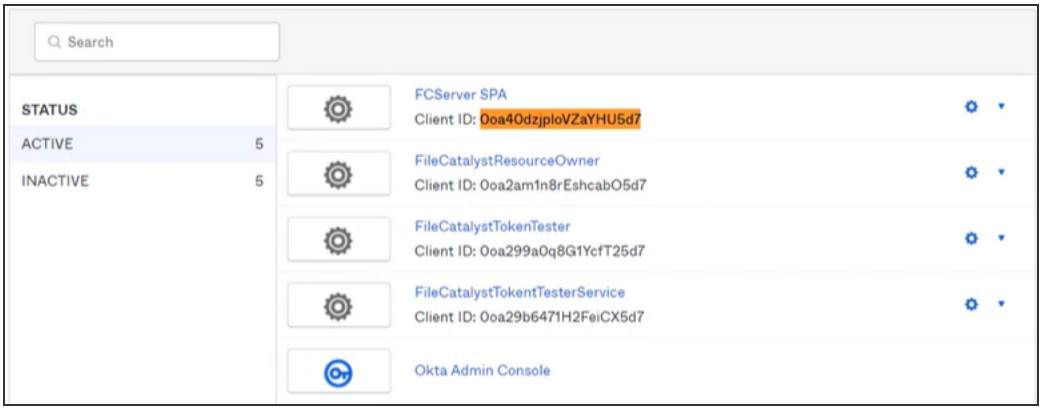
```
[FileCatalyst Server Install Location]\fcconf.conf
```

In fcconf.conf, you'll need to add an entry called **FCServer.server.config.auth.openID** with the following parameters and format:

> **NOTE:** {"audienceURI":"<VALUE>","clientID":"<VALUE>","issuerURI":"<VALUE>","metadata":"{}","serviceName":"<VALUE>","usernameClaim":"<VAL>"}

The required parameters and their values are outlined below:

- **audienceURI**
  - To find the correct audienceURI value, login into Okta as an administrator and navigate to **API > Security**. Your authorization server will be listed here with the audienceURI value.
  - If you are using the default authentication server, the audienceURI is `api://default`
- **issuerURI**
  - To find the correct issuerURI value, login into Okta as an administrator and navigate to **API > Security**. Your authorization server will be listed here with the issuerURI value.
  - If you are using the default authentication server, the issuerURI is `https://[YOUR OKTA INSTANCE].okta.com/oauth2/default`
- **clientID**
  - This value is the Client ID of the Okta application you created. This will be the same value that you entered in the **clientId** field for **loginConfig.js**.
  - You can find the Client ID by viewing the Okta server that you developed in the Okta Server Setup section.



- **metadata**
  - This field should be set to be empty curly brackets: {}
- **serviceName**
  - This value can be any alphanumeric string.
- **usernameClaimTest**

- To find the value for the field usernameClaim, login to Okta as an administrator and edit the authorization server of your choice. On the API page click on the claims tab. The name for the ID Claim Type is the proper value for the usernameClaim field. For example, for the “default” authorization server, the name for the ID Claim Type is “login”. See the below section for more information on setting up a username claim.

Below is an example of the configuration values that needs to be added to the fcconf.conf file:

```
FCServer.server.config.auth.openID=
{"audienceURI":"api://default","clientID":"0oa44444zzzzz1111122","issuerURI":"https://dev-
51111111.okta.com/oauth2/default","metadata":"{}","serviceName":"TEST_OKTA","usernameClaim":"login"}
```

> **NOTE:** The FCServer.server.config.auth.openID entry and values must all be on one line.

Setting Up A Username Claim

If your username claims do not get automatically added in Okta, you will need to add them manually.

To do so, in Okta, navigate to:

Security > API > Select the "edit" pencil > Claims > Edit claim

If your Username Claims are not present, add new claims as needed with the following values:

- ID token - **Always**
- Value type - **Expression**
- Value user - **Login**