Creatio Academy

OAuth 2.0 authorization

Set up OAuth 2.0 authorization for integrated applications

Version 8.0

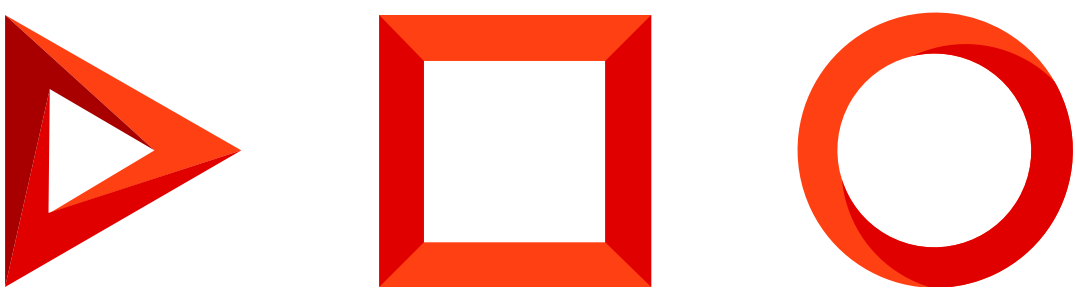


Table of Contents

Set up OAuth 2.0 authorization for integrated applications 4
- Install and set up the Identity Service 4
- Update the Identity Service using IIS 9
- Set up the Identity Service integration on Creatio's end 10
- Set up the OAuth 2.0 authorization 11

Set up OAuth 2.0 authorization for integrated applications

PRODUCTS: ALL CREATIO PRODUCTS

Use OAuth 2.0 protocol to securely authorize third-party applications and web services you integrate with Creatio. This technology does not pass Creatio logins and passwords to third-party applications. OAuth 2.0 also lets you restrict Creatio permissions for the integrated applications.

Install and set up the Identity Service

You can install and set up the Identity Service using IIS or, alternatively for Creatio version 8.0.8 and later, using Docker:

- Install and set up the Identity Service using IIS
- Install and set up the Identity Service using Docker

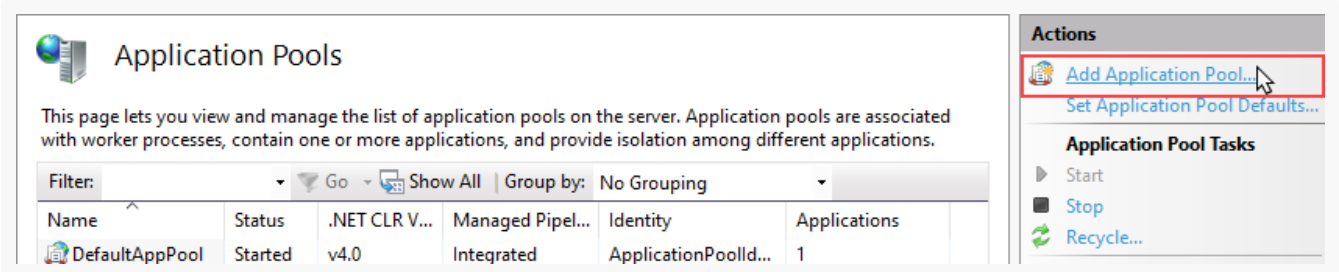
Install and set up the Identity Service using IIS

Deploy the database and Creatio application servers before installing and configuring the Identity Service. To install the Identity Service using IIS:

1. Access the Creatio application server.
2. Install additional components.
   For Creatio version 8.0.7 and earlier

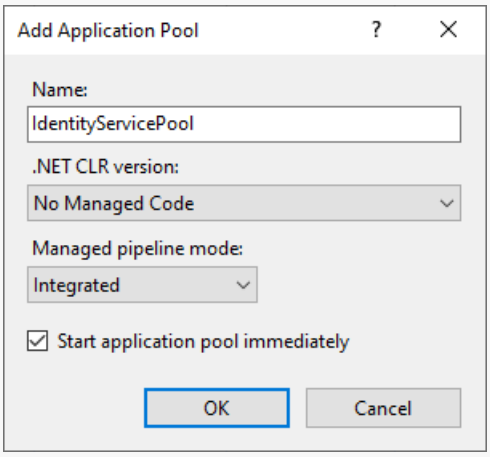
   For Creatio version 8.0.8 and later
3. Restart the IIS.
4. Navigate to the Creatio install file folder, find the **IdentityService.zip** archive, and unzip it.
5. Add the Identity Service application **pool** to the IIS.
   a. Go to the [*Application Pools*] section in the [*Connections*] area of the IIS management window.
   b. Select [*Add Application Pool...*] in the [*Actions*] area.

Fig. 1 Add the pool to the IIS



c. Specify the pool name in the pool settings window, for example, “IdentityServicePool.” Set the [*.NET CLR Version*] field to “No Managed Code.”

Fig. 2 Set up the Identity Service pool



6. Set up **access** to the application pool:

   a. Right-click the newly-created pool. Select [*Advanced Settings...*] in the context menu.

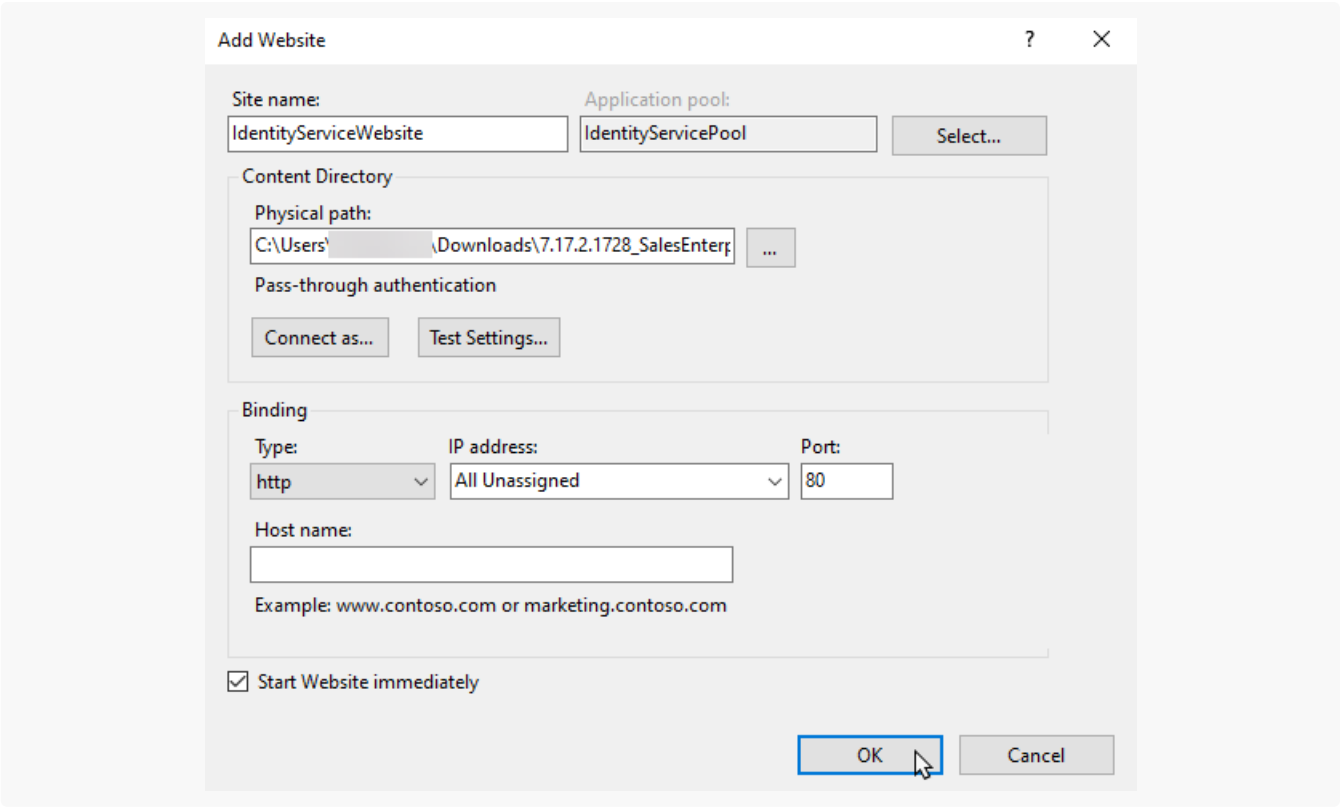
   b. Specify the user with Identity Service directory access permissions in the [*Identity*] field of the newly-opened window.

7. Create a new Identity Service **site** in the IIS.

   a. Click [*Sites*] on the IIS control panel and select [*Add Website*] from the context menu.

   b. Specify the site name, the pool and the path to the Identity Service root directory.

Fig. 3 Set up the site in the IIS



8. Connect the site to your Creatio **DBMS**. To do so, edit the **appSettings.json** configuration file in the Identity Service root directory:

   a. Set the “DbProvider” parameter to “MsSqlServer” or “Postgres.”

   b. Specify the connection string in the “MsSqlConnection” or “PostgresConnection” setting. We recommend using the same connection string you specified in Creatio. The user that connects to the database must have permissions to create and update the tables.

   Note. To connect the Identity Service to Creatio with Oracle DBMS, deploy an additional PostgreSQL or Microsoft SQL database instance.

9. Set up the Identity Service **system user**. To do so, specify the unique ClientId, ClientName, and ClientSecret values in the “Clients” block of the **appSettings.json** configuration file. The file is located in the Identity Service root catalog. Creatio and the Identity Service will use these values to interact with each other. All parameters support uppercase and lowercase letters, numbers, and special characters, for example, brackets or punctuation marks.

   Recommended parameters:

   ClientId — 16 characters.

   ClientSecret — 32 characters.

   ClientName — any number of characters.

   “Clients” block setup example:

```
"[{\"ClientId\":\"{generate ClientId}\",\"ClientName\":\"{generate name}\",\"Secrets\":[\"{ge
```

Note. To avoid errors on Identity Service launch, specify the full path to openssl.pfx in the "**X509CertificatePath**" setting of the appsettings.json file. openssl.pfx is located in the root of the Identity Service directory.

10. Switch the Identity Service to **HTTPS**. The setup process is similar to switching Creatio to HTTPS. Read more: Switch Creatio website from HTTP to HTTPS.
11. Enable the Identity Service **logging**.
    a. Navigate to the Identity Service directory, open the web.config file, and set the "stdoutLogEnabled" parameter to "true."
    b. Specify where you would like to store the Identity Service logs in the file's "stdoutLogFile" parameter.
    c. Open the appsettings.json file in the Identity Service root directory and configure the log level:

```
"Logging": {
  "LogLevel": {
    "Default": "Error"
  }
}
```

Install and set up the Identity Service using Docker

Deploy the database and Creatio application servers before installing and configuring the Identity Service.

Note. You can only install and set up the Identity Service using Docker for Creatio 8.0.8 and later. For Creatio version 8.0.7 and earlier, please install and set up the Identity Service using IIS.

To install the Identity Service using Docker:

1. Navigate to the Creatio install file folder, find the **IdentityService.zip** archive, which includes the **Dockerfile-OAuth** file for IdentityService, and unzip it.
2. Connect the site to your Creatio **DBMS**. There are three ways to do this:
   - Edit the **appSettings.json** configuration file in the Identity Service root directory before building.
   - Edit the **Dockerfile-OAuth** file and add environment variables using the **ENV directive**. For example, specify `ENV DbProvider=Postgres`, which will set the DbProvider parameter to "Postgres" when the container starts.
   - Specify the **environment variables** when running the container. For example, run the container as follows: `docker run --env=DbProvider=Postgres`.

Regardless of the chosen method, configure the following properties:

a. Set the “DbProvider” parameter to “MsSqlServer” or “Postgres.”

b. Specify the **connection string** in the “MsSqlConnection” or “PostgresConnection” setting. We recommend using the same connection string you specified in Creatio. The user that connects to the database must have permission to create and update the tables.

c. Set up the Identity Service **system user**. To do so, specify the unique ClientId, ClientName, and ClientSecret values in the “Clients” setting. Creatio and the Identity Service will use these values to interact with each other. All parameters support uppercase and lowercase letters, numbers, and special characters, for example, brackets or punctuation marks.

   Recommended parameters:

   ClientId — 16 characters.

   ClientSecret — 32 characters.

   ClientName — any number of characters.

   “Clients” setting setup example:

   ```
   "[{\"ClientId\":\"{generate ClientId}\",\"ClientName\":\"{generate name}\",\"Secrets\":[\"
   ```

d. Configure **RedisConnection** or leave it blank if configuring security settings for the IdentityService configuration is not required. The RedisConnection setting stores the machineKey value to prevent spoofing during runtime.

3. Build the **Docker image** after configuring using the command:

   ```
   docker build -t creatio-identity-service -f ./Dockerfile-OAuth .
   ```

4. Run the **container** using the following command:

   ```
   docker run --env=PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin \
   --env=ASPNETCORE_URLS=http://+:80 --env=DOTNET_RUNNING_IN_CONTAINER=true \
   --env=DOTNET_VERSION=6.0.15 --env=ASPNET_VERSION=6.0.15 --workdir=/app \
   -p 80:80 -d creatio-identity-service:latest
   ```

5. Switch the Identity Service to **HTTPS**. Before configuring HTTPS, obtain a digital certificate from the certification center in PFX format. To switch IdentityService to HTTPS, run the docker container as follows:

   ```
   docker run --env=PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
   -e ASPNETCORE_URLS="https://+;http://+" -e ASPNETCORE_HTTPS_PORT=443
   -e DOTNET_RUNNING_IN_CONTAINER=true -e DOTNET_VERSION=6.0.15
   -e ASPNET_VERSION=6.0.15 -e ASPNETCORE_Kestrel__Certificates__Default__Password={Certificate
   -e ASPNETCORE_Kestrel__Certificates__Default__Path={Certificate Path}
   -v %USERPROFILE%\.aspnet\https:/https/ --workdir=/app -p http_port_number:80 -p https_port_nu
   -d creatio-identity-service:latest
   ```

http_port_number is a port number. Docker will serve the **HTTP** version of IdentityService on this port.

https_port_number is a port number. Docker will serve the **HTTPS** version of IdentityService on this port.

{Certificate Password} is the password for the openssl.pfx certificate.

{Certificate Path} is the path to the openssl.pfx certificate.

6. Enable the Identity Service **logging**.

   a. Navigate to the Identity Service directory, open the web.config file, and set the “stdoutLogEnabled” parameter to “true.”

   b. Specify where you would like to store the Identity Service logs in the file's “stdoutLogFile” parameter.

   c. Open the appsettings.json file in the Identity Service root directory and configure the log level:

```
"Logging": {

  "LogLevel": {

    "Default": "Error"

  }

}
```

Update the Identity Service using IIS

The Identity Service archive is provided with the Creatio distribution. Please update the Creatio application before updating the Identity Service. To ensure that all required components are up-to-date:

1. Access the Creatio **application server**.
2. Install **.NET 6 Runtime**. You can download it from the Microsoft website.
3. Install **.NET 6 Hosting Bundle**. You can download it from the Microsoft website.
4. **Restart** the IIS server.

To update the Identity Service:

1. Navigate to the Identity Service application folder and **back up** the files. For example, copy the directory to a backup location.
2. Back up the **database** of the current version of Identity Service. Learn more: Creating database backup.
3. **Stop** the Identity Service pool and application in the IIS.
4. Navigate to the Creatio install file folder, find the **IdentityService.zip** archive, and unzip it.
5. **Replace** all files In the Identity Service application folder with the unzipped files.
6. Set up the Identity Service as described in Install and set up the Identity Service using IIS.
7. Start the Identity Service pool in the IIS.

8. Verify that the Identity Service is running by opening `[your-identity-app]/.well-known/openid-configuration`, where **[your-identity-app]** is the URL of your Identity Service application.

   Note. If the Identity Service is not running as expected, recover the backed-up Identity Service files, restore the database from the backup, and restart the pool and the Identity Service application.

As a result, the Identity Service for Creatio will be updated. If you encounter any issues, please contact Creatio support for assistance.

Set up the Identity Service integration on Creatio's end

1. **Enable** the OAuth 2.0 integration in Creatio. To do so, execute this script in your Creatio database. Use it for both Microsoft SQL and PostgreSql.

```
UPDATE "AdminUnitFeatureState"

    SET "FeatureState" = 1

WHERE "FeatureId" = (

    SELECT

        "Id"

    FROM "Feature"

    WHERE "Code" = 'OAuth20Integration')
```

2. Fill in the system settings in the “OAuth 2.0” group:

   a. “**Authorization server Url for OAuth 2.0 integrations**” (“OAuth20IdentityServerUrl” code) — the IdentityServer URL, for instance, http://isEnpointExample.

   b. “**Client id for OAuth 2.0 integrations**” (“OAuth20IdentityServerClientId” code) — the Identity Service user ID you specified in the “ClientId” parameter of the appSettings.json file when setting up the IdentityServer.

   c. “**Client secret for OAuth 2.0 integrations**” (“OAuth20IdentityServerClientSecret” code) — the Identity Service user's secret key you specified in the “ClientSecret” parameter of the appSettings.json file when setting up the IdentityServer.

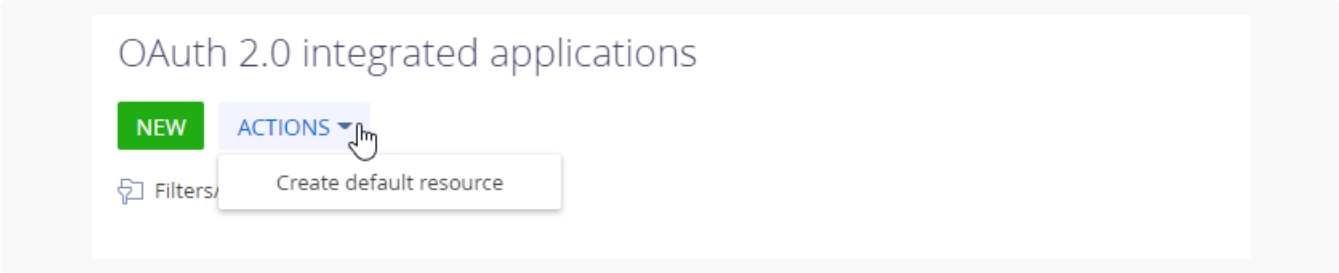
3. Create a default resource. You only need to perform this action once when setting up the Identity Service integration.

   a. Click to open the System Designer.

   b. Open the [*OAuth 2.0 integrated applications*] section.

c. Select [*Create default resource*] in the [*Actions*] menu.

Fig. 4 Add a default resource



This will create a default resource record with your Identity Service account details.

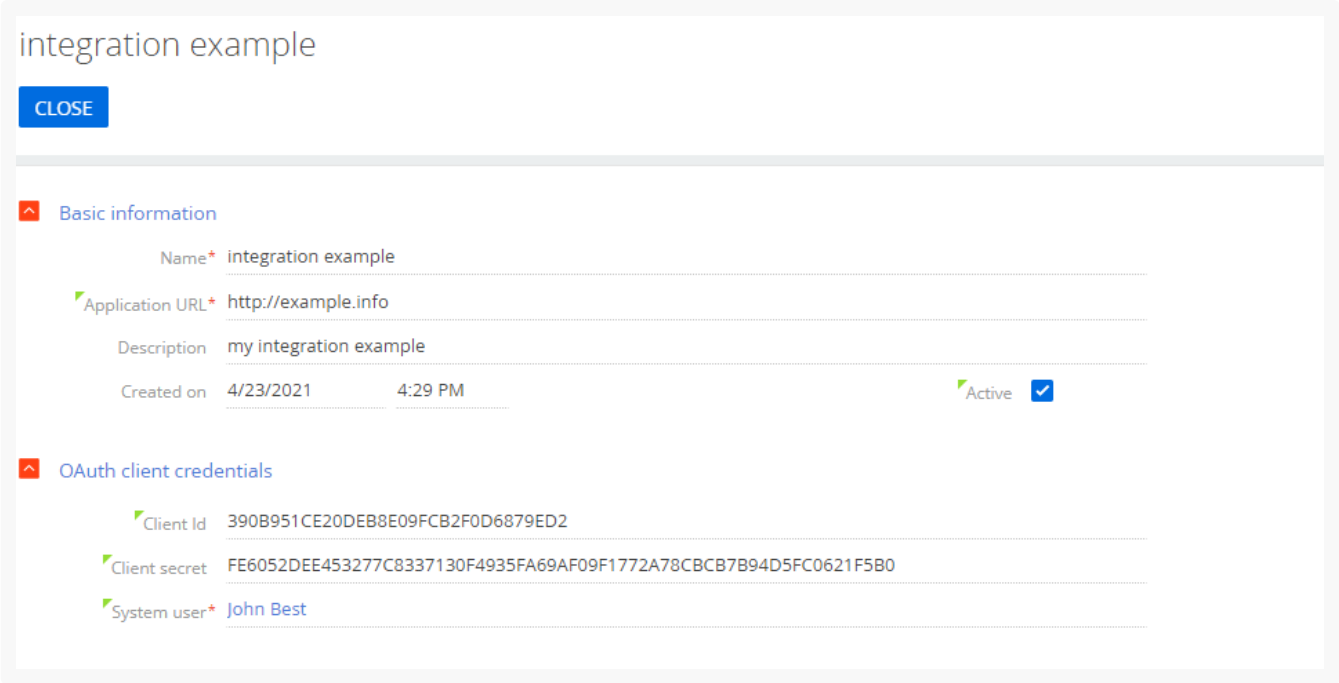
Set up the OAuth 2.0 authorization

Once you install the Identity Service and connect it to Creatio, add a client record for each application you are going to authorize with OAuth 2.0. To do so:

1. Click to open the System Designer.
2. Open the [*OAuth 2.0 integrated applications*] section.
3. Click [*New*].
4. Fill in the client parameters for the relevant application on the newly-opened page.
   a. [*Name*] — the title that the integration list and the logs will use.
   b. [*Application URL*] — the URL of the integrated application or the web service.
   c. [*Description*] — the purpose of the integration.
   d. [*Active*] — enables and disables the integration.
   e. [*System user*] — the Creatio user with sufficient permissions for this integration. We recommend permitting this user to only read and edit the fields the integrated application or the web service need to change. For example, if you are integrating a web service that passes the currency exchange rates to Creatio, grant permissions to only read and edit the [*Rate*] and [*Start*] fields of the [*Currency*] lookup.

      Creatio automatically populates the **client account details** (the ID and the secret).

Fig. 5 Set the client



5. Save the record.
6. Repeat steps 3-6 for all applications you need to authorize with OAuth 2.0.