# Configure OAuth 2.0 Credentials

Last Modified on 07/17/2024 10:35 am EDT

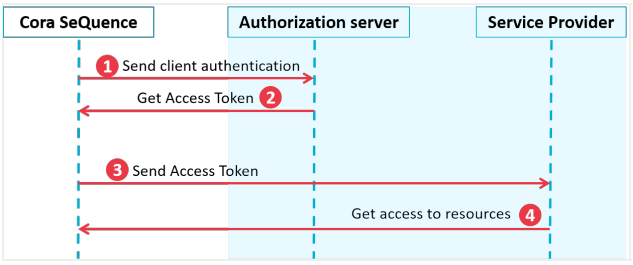
Starting with V10.0, Cora SeQuence has been renamed to Cora Orchestration.

## V10.6 and later

### Overview

Cora Orchestration supports OAuth 2.0 Client Credentials Grant flow to consume external services. In the Client Credentials Grant flow there is no user interaction. The application executes the authentication request directly, on behalf of itself. The Client Credential Grant flow is used for machine to machine authorization. The access token is issued on the server, authenticating only the client, not the user.

### Client Credentials Grant flow




1. Cora Orchestration authenticates itself to the authorization server and requests an access token.
2. The authorization server issues an Access Token.
3. The access token is used to authenticate Cora Orchestration to the service provider.
4. Data from the service provider is sent to Cora Orchestration.

For more information on the OAuth 2.0 Authorization Framework, see this page.

### Parameters

When you configure OAuth 2.0 credentials, all parameters are mandatory.

| Parameter | Description |
|---|---|
| Client Identifier | A unique string representing the registration information of the client at the authorization server. |
| Secret Source | The source from where secret keys are fetched. It can be an internal source or an external source like a key vault. |
| Client Secret | A value used to authenticate the client, Cora Orchestration, when requesting a token from the authorization server. |
| Token Endpoint | The URL used to interact with the authorization server to obtain the access token. |

| Parameter | Description |
| --- | --- |
| Scopes | The scope of the access requested. The value of the scope parameter is expressed as a list of space- delimited, case-sensitive strings. The strings are defined at the authorization server. |

## Prerequisites

Make sure that Cora Orchestration is registered with the authorization server, and that you have the required parameters:

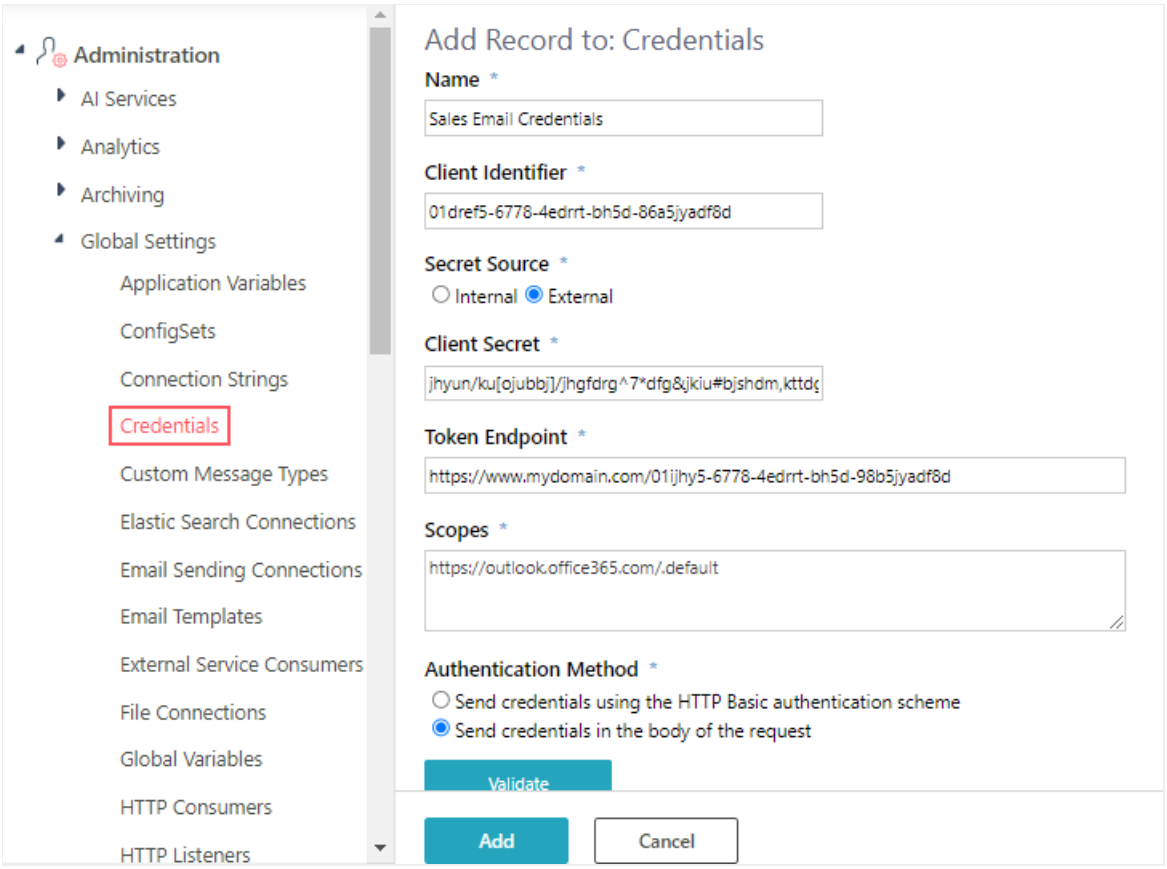
- Client ID
- Client secret
- OAuth 2.0 token endpoint
- Defined scopes

## Configuration

1. On the Admin console, go to **Administration > Global Settings > Credentials**.
2. Click **Add Credentials**, and then select **OAuth2 Client Credentials**.
3. Enter a significant name for the credential.
4. Enter the required parameters:
    - Client Identifier
    - Secret Source
    - Client Secret
    - Token Endpoint
    - Scopes
5. Under Authentication Method, select one of the following options:
    - **Send credentials using the HTTP Basic authentication scheme**: This is the default option. In this option Cora SeQuence sends its credentials via an HTTP Authorization Header with scheme Basic.
    - **Send credentials in the body of the request**: Recommended option for Azure implementations, which require passing parameters in the body of the request.

      NOTE
      It is not recommended to include client credentials in the request body, and this option should be limited to clients that are unable to use the HTTP basic authentication scheme.

Example of configured OAuth 2.0 credentials




6. To verify that the settings you entered are correct, click **Validate**.

   NOTE
   Validation is not a *mandatory* step. The values you add to the credential may not be valid at design time. For example, the information was provided by the customer, but has not yet been updated in the runtime environment.

7. Click **Add**.

The OAuth 2.0 client credentials are added to the list of credentials and are available for setting up Email Listener integration activities that use the EWS protocol.

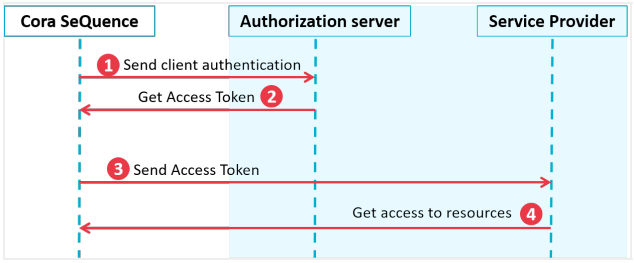
## V8.7-V10.5

### Overview

Starting with V8.7, Cora SeQuence supports OAuth 2.0 Client Credentials Grant flow to consume external services. In the Client Credentials Grant flow there is no user interaction. The application executes the authentication request directly, on behalf of itself. The Client Credential Grant flow is used for machine to machine authorization. The access token is issued on the server, authenticating only the client, not the user.

Currently, you can use OAuth 2.0 credentials only for Email Listener activities, but in the future, it will support other integration activities.

#### Client Credentials Grant flow




1. Cora SeQuence authenticates itself to the authorization server and requests an access token.
2. The authorization server issues an Access Token.
3. The access token is used to authenticate Cora SeQuence to the service provider.
4. Data from the service provider is sent to Cora SeQuence.

For more information on the OAuth 2.0 Authorization Framework, see this page.

## Parameters

When you configure OAuth 2.0 credentials, all parameters are mandatory.

| Parameter | Description |
|---|---|
| Client Identifier | A unique string representing the registration information of the client at the authorization server. |
| Client Secret | A value used to authenticate the client, Cora SeQuence, when requesting a token from the authorization server. |
| Token Endpoint | The URL used to interact with the authorization server to obtain the access token. |
| Scopes | The scope of the access requested. The value of the scope parameter is expressed as a list of space- delimited, case-sensitive strings. The strings are defined at the authorization server. |

## Prerequisites

Make sure that Cora SeQuence is registered with the authorization server, and that you have the required parameters:

- Client ID
- Client secret
- OAuth 2.0 token endpoint
- Defined scopes

## Configuration

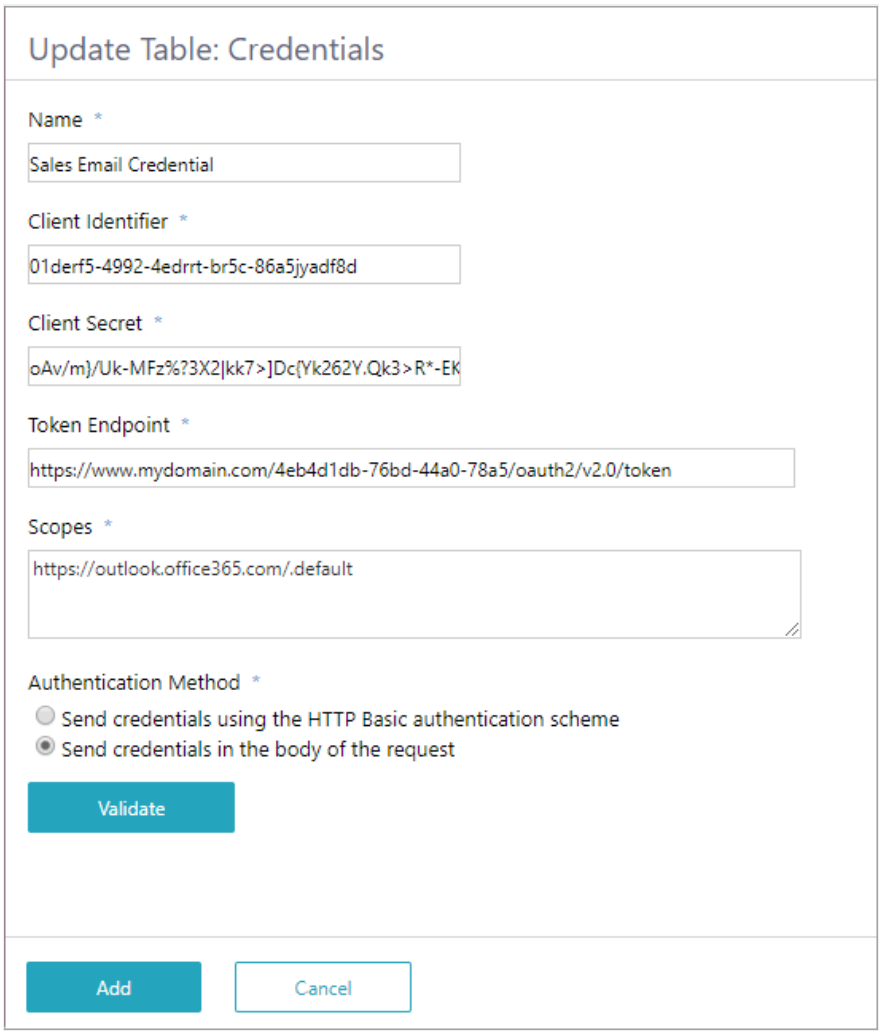
1. On the Admin console, go to **Administration > Global Settings > Credentials**.
2. Click **+ Add Credentials**, and then select **OAuth2 Client Credentials**.
3. Enter a significant name for the credential.
4. Enter the required parameters:
    - Client Identifier
    - Client Secret
    - Token Endpoint

- Scopes

5. Under Authentication Method, select one of the following options:
   - **Send credentials using the HTTP Basic authentication scheme**: This is the default option. In this option Cora SeQuence sends its credentials via an HTTP Authorization Header with scheme Basic.
   - **Send credentials in the body of the request**: Recommended option for Azure implementations, which require passing parameters in the body of the request.

     NOTE:
     It is not recommended to include client credentials in the request body, and this option should be limited to clients that are unable to use the HTTP basic authentication scheme.

     Example of configured OAuth 2.0 credentials

6. To verify that the settings you entered are correct, click **Validate**.

   NOTE:
   Validation is not a *mandatory* step. The values you add to the credential may not be valid at design time. For example, the information was provided by the customer, but has not yet been updated in the runtime environment.

7. Click **Add**.

The OAuth 2.0 client credentials are added to the list of credentials and are available for setting up Email

Listener integration activities that use the EWS protocol.