Using OAuth2 to Access REST API

Agiloft supports the OAuth Authorization Code Flow to obtain access tokens from our authorization server. It is a secure and flexible flow that can be used by both confidential and public clients. This requires you to configure an OAuth2 client in your knowledgebase (KB) by creating an API application, which requests permission and authentication from the user before performing any API requests on their behalf. When an API request is made, the user is redirected to the API application, asked to authenticate themselves and grant the requested permissions, and then redirected to the web application.

The Authorization Code Flow is considered to be the most secure OAuth flow because it does not expose the access token to the user. It is recommended for use by all clients, but it is especially important for confidential clients that need to store access tokens securely.

Here are some examples of who should use the OAuth Authorization Code Flow:

- Web applications that need to access protected resources on behalf of users.
- Mobile applications that need to access protected resources on behalf of users.
- APIs that need to be accessed by other applications.

Agiloft's implementation of OAuth2 complies with the OAuth 2.0 specification and provides a variety of security benefits, as described in RFC 6749.

> ⚠ **Prerequisites**
>
> You need the Advanced or Premium edition of Agiloft to use REST APIs. This feature works only with native Agiloft users; it is not compatible with LDAP.

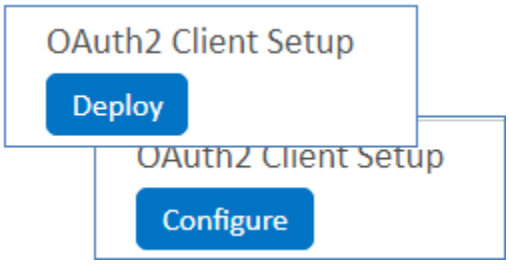
Creating an API Application

The OAuth2 client setup begins with creating and configuring a new API application. When you create the application, you associate it with a specific user in the KB to determine which permissions are used when the server handles a REST API call. All REST API requests will be executed on behalf of this user.

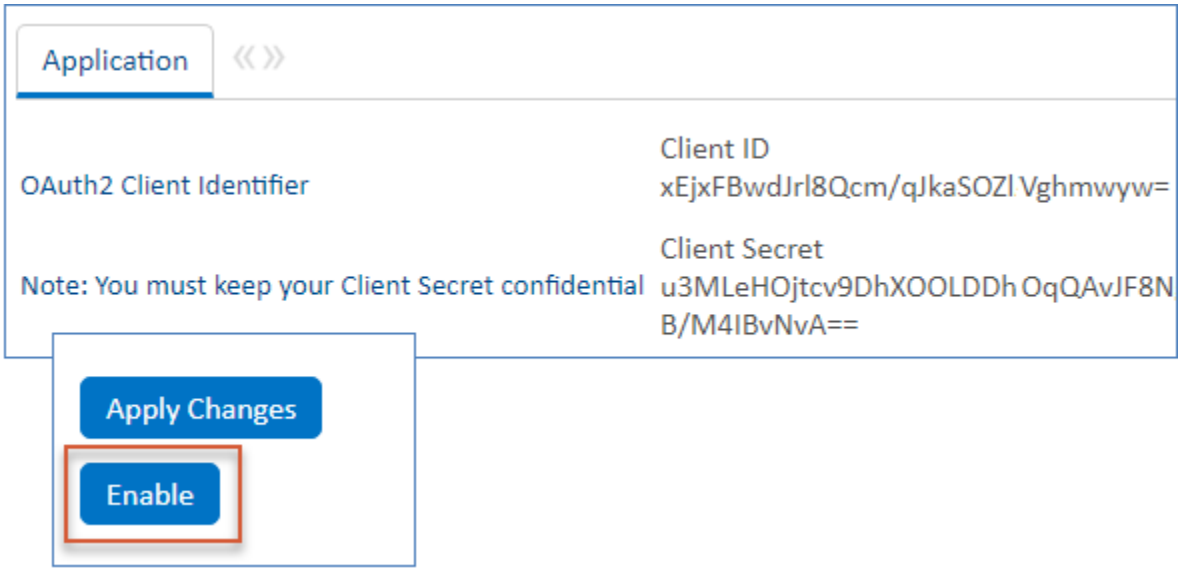
To create an API application:

1. Click the **Setup** gear in the top-right corner and go to **Integration > OAuth2 Client Setup**.
2. Click Deploy or Configure:
   - **Deploy**: The OAuth2 client hasn't been deployed yet. Clicking Deploy configures the OAuth2 client for use in your KB, which can take a few minutes to complete.

- **Configure**: The OAuth2 client has already been deployed. Clicking Configure takes you to the API Application screen.



3. On the API Application screen, click New to open the API Application Settings wizard.
4. Complete the required fields:
    a. **Name**: The name that the application uses with Agiloft's REST API.
    b. **Display Name**: The name of the application that appears in the KB.
    c. **Associate this Application with Contact ID**: The ID of the user associated with the application. All permissions for this user are used when the REST API is called. This field is automatically populated when you select a user in the Full Name field.
    d. **Full Name**: The name of the user associated with the application.
    e. **Redirect URI**: The webpage where you want to redirect users at the end of the authorization process. The redirect URI specified in your OAuth request must belong to this URI.
    f. **Token Expiry in Minutes**: The amount of time in minutes before the access token expires. The default value is 15, and the acceptable range is from 1 to 60.
5. Click Apply Changes.
6. Click Enable. This activates the application and generates the Client ID and Client Secret, which are used during the token exchange.



⚠ If you need to block your application for any reason, click Disable, which replaces Enable after it's been clicked.

Changing Application Settings

Verify that your application settings are correct so that you don't need to change them later. If you need to change your application's settings but have already completed the authentication procedure described below, you need to re-authenticate for your changes to take effect. If you need to make changes but have already received an access token, you need to revoke the access token, make the changes, and then obtain a new access token for your changes to take effect. See Revoking Tokens for more information.

Changes to the values in the Associate this Application with Contact ID field or the Redirect URI field can sometimes block your application. If this happens, recreate the application with the correct settings.

⚠ You can change the Token Expiry in Minutes value without performing the re-authentication procedure. Changes to this field take effect when the next access token is requested.

OAuth Token Exchange

After the API application is created, the OAuth authorization process occurs through a series of HTTP calls made to various endpoints on the KB server. The following section describes the process and serves as a reference for the parameters in each client request and server response.

Authorization Request

The OAuth process starts with the client directing the user's browser to make a request to the `/ewws/oauth` endpoint, where the Content-Type must be in the form `application/x-www-form-urlencoded`. The request includes the following query string parameters:

Parameter	Value	Required	Description
response_type	code	Yes	Defines the response type and must always have a value of `code`.
client_id	Obtained from API Application Settings wizard	Yes	Identifies an application.
redirect_uri	Absolute URI defined on the API Application Settings wizard, such as `https://example.agiloft.com`	Yes	Defines where the user is redirected at the end of the authorization process. The value must belong to the set of values specified on API Application Settings wizard.

scope	permissions_for: {CONTACT_ID}	Yes	Defines the permissions that the user is asked to approve, which are determined by the Contact ID defined in the API Application Settings wizard. For example, if the Associate this Application with Contact ID field is set to 222, then this parameter should be set to `permissions_for:222`.
state	Any string	Recommended	Denotes a value that is returned to the client as a parameter at the end of the authorization process, which verifies the validity of the request. Although this parameter is not required, we highly recommended using it to protect against cross-site request forgery (CSRF), as described in RFC 6749.

Example Authorization Request

```
https://example.agiloft.com/ewws/oauth?response_type=code&redirect_uri=https%3A%2F%
2Ftest.com%2Freceiver&
    client_id=Bvn7k4fIdMEZQrJJ7ZCIQgErlTDbX9L73LThA5YA4W0%3D&scope=permissions_for%
3A213&
    state=LQKFNL023478_3259423
```

The authorization request asks the user to authenticate themselves and then authorize the request, which redirects the user to the URI value defined for the `redirect_uri` parameter.

Successful Authorization Request

If the request succeeds, the following query string parameters are sent to the client in response:

Parameter	Value
state	Returns the value of the state parameter from the authorization request.
code	Indicates the authorization code for use in the upcoming access token request. This code is valid for five minutes. Protecting the authorization code, which is a secret, is one of the security implications of this OAuth flow.
api_access_point	Defines the base URI for all REST API calls that use an access token, if any. If this value is empty, use the base URI for your KB's server.
client	Provides system information.

Example Response to a Successful Request

```
HTTP/1.1 302 Found
Location: https://example.agiloft.com/receiver?client=&state=LQKFNL023478_3259423&
    code=EFLJHELFH23487402387LKFEHJFEHF=
```

Failed Request

If the request fails, the following query string parameters are sent to the client in response:

Parameter	Value
error	Defines the error code. See below for a description of each error code.
error_description	Describes why the request failed.

The following error codes are possible:

Code	Description
invalid_request	The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed.
unauthorized_client	The client is not authorized to request an authorization code using this method.
access_denied	The resource owner or authorization server denied the request.
unsupported_response_type	The authorization server does not support obtaining an authorization code using this method.
invalid_scope	The requested scope is invalid, unknown, or malformed. See the scope parameter description in the Authorization Request section above.
server_error	The authorization server encountered an unexpected condition that prevented it from fulfilling the request.

Access Token Request

If the authorization request is successful, the client makes an HTTP POST to the `/ewws/otoken` endpoint by using the `api_access_point` value retrieved in the previous step. The following parameters are used:

Parameter	Value	Required	Description
grant_type	authorization_code	Yes	Defines the grant type, which must always be `authorization_code`.
code	The authorization code obtained in the previous step	Yes	Confirms that the previous step was successful.
client_id	Obtained from the API Application Settings wizard	Yes	Identifies the application.
redirect_uri	Must match the value used in the previous step	Yes	This value must belong to the set of values specified in API Application Settings wizard.

Example Access Token Request

```
POST /ewws/otoken HTTP/1.1
Host: example.agiloft.com
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&
code=EFLJHELFH23487402387LKFEHJFEHF&
client_id=Bvn7k4fIdMEZQrJJ7ZCIQgErlTDbX9L73LThA5YA4W0%3D&
redirect_uri=https%3A%2F%2Ftest.com%2Freceiver
```

Successful Access Token Response

The authorization code from the request is exchanged for OAuth2 tokens, and the server returns a JSON array with the following keys:

Parameter	Value
access_token	The access token, which can be used as an authorization header in the Agiloft REST API.
refresh_token	The refresh token, which can be used to get a new access token. Refresh tokens expire after 28 days of inactivity.
token_type	The value will always be `Bearer`.
expires_in	The expiration time in minutes for the access token.

Example Token Response

```
{
  "access_token":"eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIyX2FkbWluI",
  "refresh_token":"LKHEFOP932875KJGKJG32423542LHLKHFD_FDKLJ.OJ==",
  "token_type":"Bearer",
  "expires_in":15
}
```

Refresh Request

At any time the client can use the refresh token to receive a new access token. To start a request, the client makes an HTTP POST to the `/ewws/otoken` endpoint by using the `api_access_point` value retrieved during the authorization request. The following parameters are used:

Parameter	Value	Required	Description

grant_type	refresh_token	Yes	The value must always be `refresh_token`.
md5_secret	The first 20 characters of the Client Secret, obtained from API Application Settings wizard	Yes	Authenticates the application.
refresh_token	Refresh token received during the previous step	Yes	Defines the refresh token.

Example Refresh Token Request

```
POST /ewws/otoken HTTP/1.1
Host: example.agiloft.com
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token&
md5_secret=F64EC9CE9852EE154696&
refresh_token=LKHEFOP932875KJGKJG32423542LHLKHFD_FDKLJ.OJ%3D%3D
```

The refresh token can then be used to generate a new access token. The server returns a JSON array with the following keys:

Parameter	Value
access_token	The access token, which can be used as an authorization header in the Agiloft REST API.
token_type	The value will always be `Bearer`.
expires_in	The expiration time in minutes of the access token.

Example Response

```
{
  "access_token":"wehRT74iOiJIUzI1NiJ0.weJzdWIiOiIyX2FkbWluI",
  "token_type":"Bearer",
  "expires_in":15
}
```

Revoking Tokens

Access tokens and refresh tokens can be revoked at any time by the authorizing user. REST operations cannot be performed with a token that has been revoked. If an access token is revoked and it has a corresponding refresh token, the refresh token is also revoked. If a refresh token is revoked, all the access tokens issued from that refresh token are also revoked.

Refresh tokens can be revoked by making a POST call to the `/ewws/orevoke` endpoint by using the `api_access_point` value retrieved during the authorization request. The following parameters are used:

Parameter	Value	Required	Description

revoke_for	OAuth2 refresh token or Client Secret key from API Application Settings wizard	Yes	Defines the refresh token or Client Secret Key.

Example Request to Revoke a Refresh Token

```
POST /ewws/orevoke HTTP/1.1
Host: example.agiloft.com
Content-Type: application/x-www-form-urlencoded

revoke_for=LKHEFOP932875KJGKJG32423542LHLKHFD_FDKLJ.OJ%3D%3D
```

Response to Revoking a Token

If the request to revoke the token succeeds, an HTTP success code 200 is returned with an empty body or without any body at all. If the request fails, an HTTP code 400 is returned with the error code `INVALID_REQUEST`.

Example Response to a Failed Request

```
{
  "error" : "INVALID_REQUEST",
  "error_description" : "Invalid token."
}
```