

# How to Set Up Log In with OAuth Tokens in Exalate for ServiceNow

Last Modified on 01/15/2026 7:47 pm EST

To sync your data, Exalate requires access to your ServiceNow instance. To avoid storing sensitive information, we provide an alternative way to authenticate to a ServiceNow instance without storing usernames and passwords. Exalate supports the OAuth2 protocol of ServiceNow.

## How to Authenticate and Set Up Exalate with OAuth2

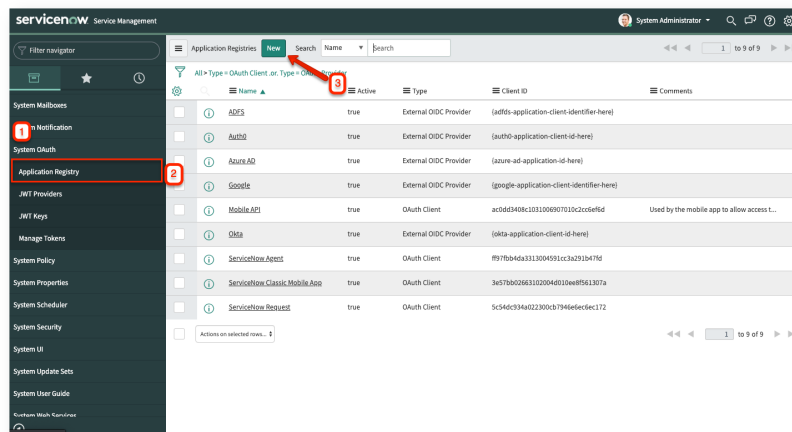
### 1. Enable OAuth on ServiceNow

To use OAuth API in ServiceNow, make sure the `com.snc.platform.security.oauth.is.active` system property is in `true`.

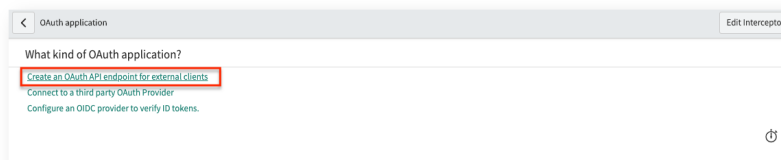
**Note:** For more information, please see the [ServiceNow documentation](#).

### 2. Create an endpoint for clients to access the instance

#### 1. After logging in - navigate to **System OAuth** → **Application Registry** → **New**.



#### 2. Select **Create an OAuth API endpoint for external clients**.



#### 3. Provide a name for the application registry and extend the access token lifespan.

Application Registries  
New record

OAuth client application details.

- Name: A unique name.
- Client ID: Client ID automatically generated by ServiceNow OAuth server.
- Client Secret: Client secret for the OAuth application. Leave it empty for auto-generation.
- Refresh Token Lifespan: Time in seconds the Refresh Token will be valid.
- Access Token Lifespan: Time in seconds the Access Token will be valid.
- Redirect URL: The redirect URLs authorization server redirect to. They must be absolute URLs and they are comma separated.

[More Info](#)

\* Name:  ⓘ

Application:  ⓘ

\* Client ID:

Accessible from:

Client Secret:

Active: ☒

\* Refresh Token Lifespan:

\* Access Token Lifespan:

Redirect URL:

Logo URL:

Comments:

The name is used to identify the application registry.

Exalate auto-renews the access token whenever the application lifespan expires. The lifespan is expressed in seconds, 7200 seconds is 2 hours.

4. Submit the entry, reopen the registry and then copy **client\_id** and **client\_secret**

Application Registries - Exalate

OAuth client application details.

- Name: A unique name.
- Client ID: Client ID automatically generated by ServiceNow OAuth server.
- Client Secret: Client secret for the OAuth application. Leave it empty for auto-generation.
- Refresh Token Lifespan: Time in seconds the Refresh Token will be valid.
- Access Token Lifespan: Time in seconds the Access Token will be valid.
- Redirect URL: The redirect URLs authorization server redirect to. They must be absolute URLs and they are comma separated.

[More Info](#)

\* Name:  ⓘ

Application:  ⓘ

\* Client ID:

Accessible from:

Client Secret:  ⓘ ⓘ

\* Refresh Token Lifespan:

\* Access Token Lifespan:

Redirect URL:

Logo URL:

Comments:

• 5. Now generate a 'refresh token' by entering the following curl command

```

read -r -d '' CSECRET <<'EOF'

EOF
read -r -d '' PASS <<'EOF'

EOF
curl --data-urlencode "grant_type=password" --data-urlencode "client_id=" --data-urlencode "client_secret=$CSECRET"
--data-urlencode "username=" --data-urlencode "password=$PASS" /oauth_token.do

# where
# * - The clientid from the application registry copied in step 4
# * - the clientsecret from the application registry copied in step 4
# * - the name of the proxy user
# * - the password of the proxy user
# * - the url of your servicenow instance

```

For instance,

```

read -r -d '' CSECRET <<'EOF'
foobar!askfj!0
EOF
read -r -d '' PASS <<'EOF'
ExalatePWD
EOF
curl --data-urlencode "grant_type=password" --data-urlencode "client_id=1234567890" --data-urlencode "client_secret=$CSECRET" --data-urlencode "username=ExalateIntegration" --data-urlencode "password=$PASS" https://dev12345.service-now.com/oauth_token.do

```

It returns a JSON

structure

```

{
  "access_token": "tqlvTscjoS2IV1yrasu-1234455443NUes4YEm1IBdX0EjHUmVB-Y3u6Zur8UgzLj_eTUeEBBmWtEgmw",
  "refresh_token": "fygKJXPAY3bl9tVaXk-1234455443LiMUeOH7RPYuWg1N2UKnUIZMzzm6UPsZ7DG4jeXPwIBaEw",
  "scope": "useraccount",
  "token_type": "Bearer",
  "expires_in": 7199
}

```

Copy the **access\_token** and **refresh\_token**

6. Access your exalate node at <https://snownode-aaaa-bbbb-cccc-dddd.exalate.cloud>

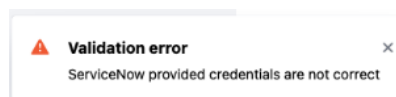
You can log in using the **access\_token** copied from step 5

## 7. Access general settings and hit configure

Enter the **clientid** and **clientsecret** copied in step 4, and the **refresh\_token** in step 5 in the different fields, and save

## 8. Get confirmation

A flag is raised in case no access is possible



## 9. All reads/searches/updates are now done with the proxy users

**Note:** This authentication can be used as long as the refresh token is valid. You can configure the token when setting up the endpoint on the first step. You can generate a new refresh token following step 2 once it is expired. We recommend setting a long lifespan on the refresh token. The default setting (8640000) equals 100 calendar days.

# What to do if the Refresh Token is Expired?

Product Repeat the steps as detailed above (from step 1)

[About Us](#)

[Release History](#)

**ON THIS PAGE**

[Glossary](#)

[API Reference](#)

[How to Authenticate and Set Up Exalate with OAuth2](#)

[Security](#)

[What to do if the Refresh Token is Expired?](#)

[Pricing and Licensing](#)

[Resources](#)

**resources**

[Subscribe for a weekly Exalate hack](#)

[Academy](#)

[Blog](#)

[YouTube Channel](#)

[Ebooks](#)

**Still need help?**

[Join our Community](#)

[Visit our Service Desk](#)

[Find a Partner](#)