

AQUILA - Salesforce Integration in Two Different Methods (JWT Bearer Flow and Username-Password Flow)

Salesforce requires secure communication protocols for authorization and data exchange between external applications and Salesforce orgs. This involves creating digital certificates, configuring external client apps, and establishing secure authentication methods. OpenSSL provides the cryptographic tools needed to generate private keys and self-signed certificates for secure communication over networks.

Integration Overview

This integration supports secure communication through:

- JWT (JSON Web Token) authentication using digital certificates
- OAuth authentication with external client apps
- Self-signed certificates and keystore management

Organizations can authorize Salesforce CLI commands and establish secure API connections using these authentication methods.

Compatibility

- Supports Salesforce CLI authorization via JWT Bearer Flow
- Compatible with macOS, Linux, and Windows operating systems
- Requires OpenSSL for certificate generation

Installing OpenSSL in your Log Collector (JWT Bearer Flow)

OpenSSL is an open-source software library that provides tools and protocols for secure communication over networks. It helps encrypt data so that information like passwords, credit card numbers, and private messages stay secure when sent over the internet.

Step 1:

In Linux:

Install OpenSSL on your system:

```
sudo apt install openssl
```

In Windows:

you can refer this manual for more info: [Openssl Installation in Windows 11](#)

Step 2:

Verify OpenSSL installation by running:

- macOS/Linux: `which openssl`
- Windows: `where openssl`

Creating a Private Key and Self-Signed Digital Certificate

A digital certificate and the private key used to sign the certificate are needed to authorize an organization using the `org login jwt` command. While it is strongly advised to utilize a certificate issued by a certifying authority, you can use OpenSSL to generate a self-signed certificate to get started.

This process produces two files:

- **server.key** — The private key used when authorizing an org with the `org login jwt` command
- **server.crt** — The digital certificate uploaded when creating the required external client app

Step 1:

Open a terminal (macOS and Linux) or command prompt (Windows).

Step 2:

Create a directory to hold the generated files and navigate to it:

```
mkdir /Users/jdoe/JWT  
cd /Users/jdoe/JWT
```

Step 3:

Create a private key and save it as server.key file:

Remember to change "<your password>" to the password of your choice. The password should be the same with the **server.pass.key** and **server.key**.

- server.pass.key command

```
openssl genpkey -aes-256-cbc -algorithm RSA -pass pass:<your password> -out server.pass.key -pkeyopt rsa_keygen_bits:2048
```

- server.key command

```
openssl rsa -passin pass:<your password> -in server.pass.key -out server.key
```

Step 4:

Use the server.key file to create a certificate signing request and save it as server.csr:

When prompted, provide your organization's details. Enter only the **Country Name, State or Province, Locality, and Organization Name**—you may leave all other fields blank.

Do not enter a password when generating the `server.csr`, as it may cause an authentication mismatch.

```
openssl req -new -key server.key -out server.csr
```

Step 5:

Create a self-signed digital certificate using the server.key and server.csr files:

```
openssl x509 -req -sha256 -days 365 -in server.csr -signkey server.key -out server.crt
```

Step 6:

Clone the server.key file and save it as server.pem

Important step to successfully integrate into SIEM

- For Linux

```
cp server.key server.pem
```

- For Windows

- o copy server.key server.pem

Creating User for JWT Bearer Flow/Username-Password Flow

- In Setup, enter Users in the Quick Find box, then select **Users**.
- Click **New User**.

The screenshot shows the Salesforce Setup interface. In the top left, a search bar contains 'users'. Below it, the left sidebar shows a list of setup items, with 'Users' highlighted under 'User Management Settings'. A red box and arrow labeled '1' point to the search bar, and another red box and arrow labeled '2' point to the 'Users' link. The main content area is titled 'All Users' and includes a table of existing users. A red box and arrow labeled '3' point to the 'New User' button in the top right of the table.

Action	Full Name	Alias	Username
☐ Edit	User, Integration	integ	
☐ Edit	User, Security	sec	
☐ Edit	Chatter Expert	Chatter	
☐ Edit	EPIC, OrgFarm	OEPIE	
☐ Edit		ben	
☐ Edit	dady, Mac	mdady	
☐ Edit	testing	benj	

- Fill out the form, and assign the System Administrator.
- Role > None Specified
- User License > Salesforce
- Profile > System Administrator

- Click **Save**.

Creating an External Client App in Salesforce (JWT Bearer Flow)

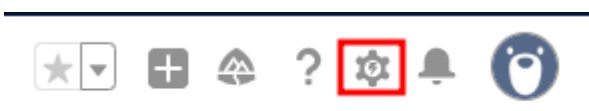
Salesforce CLI requires an external client app in the org that you're authorizing. An external client app is a packageable framework that enables a third-party application (Salesforce CLI) to integrate with Salesforce using APIs and security protocols. You must create your own external client app when authorizing the org with the `org login jwt` command.

Step 1:

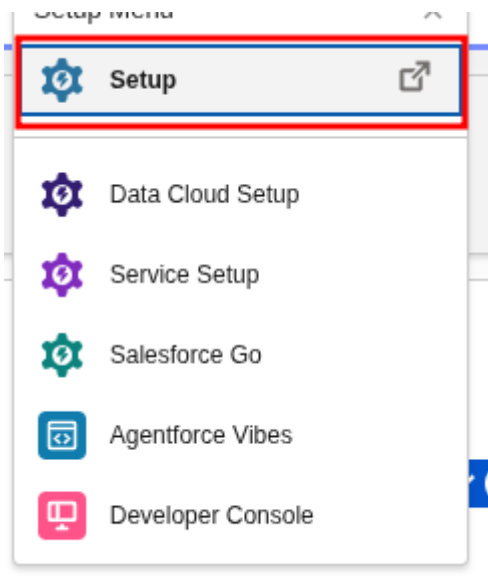
Log in to your Salesforce Organization.

Note: If the salesforce dashboard interface is in classic mode change it to lightning mode.

- In the Upper Right Corner click the gear icon.



- Select setup.



STEP 2:

To find the base URL and instance URL follow the guide below.

- In quick find box, enter **my domain** then select my domain under Company Settings.
- Under My Domain Details copy **Current My Domain URL** that's your base URL and Instance URL. (Give it to Cytech Support)

1. Click the 'Setup' button in the top left.

2. Click 'My Domain' in the left sidebar under 'Company Settings'.

3. Click the 'Current My Domain URL' field in the 'My Domain Details' section.

My Domain Settings

My Domain showcases your company's brand and keeps your data more secure. The domains that Salesforce hosts for your org include your company-specific My Domain name.

My Domain Details [Edit](#)

Current My Domain URL	https://orgfarm-08c92df946-dev-ed.sandbox.my.salesforce.com with partitioned enhanced domains
My Domain Name	orgfarm-08c92df946-dev-ed
Domain Suffix	Standard (*.my.salesforce.com)

[Edit](#)

Routing and Policies [Edit](#)

Salesforce Edge Network: Salesforce Edge Network applies to most provisioned and deployed domains for this org. See Salesforce Edge Network in Salesforce Help for details. This org uses Salesforce Edge Network. [i](#)

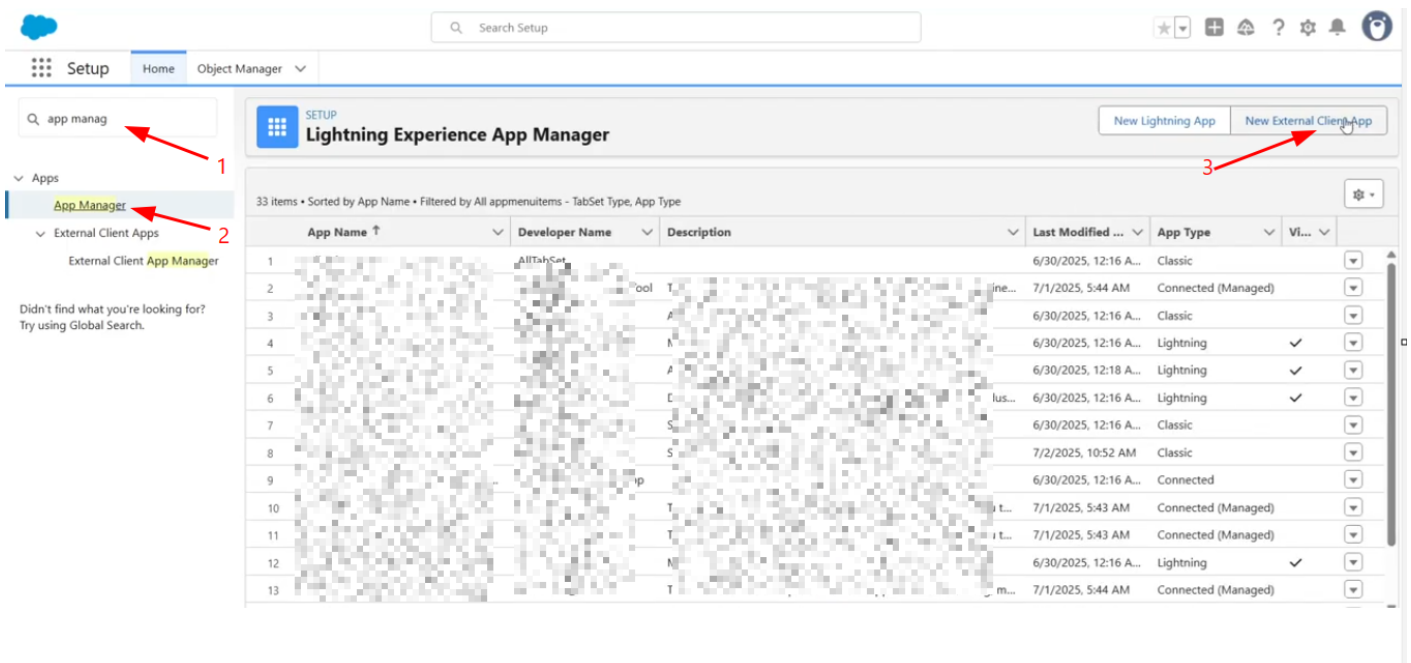
Routing Method

☒ Global [i](#)

☐ Use Salesforce locations in India only [i](#)

Step 3:

From the Quick Find box in Setup, enter **App Manager**, then click **App Manager**.



1

2

3

Lightning Experience App Manager

New Lightning App New External Client App

33 items • Sorted by App Name • Filtered by All appmenuitems - TabSet Type, App Type

	App Name ↑	Developer Name	Description	Last Modified ...	App Type	Vi...
1		AllTabSet		6/30/2025, 12:16 A...	Classic	
2				7/1/2025, 5:44 AM	Connected (Managed)	
3				6/30/2025, 12:16 A...	Classic	
4				6/30/2025, 12:16 A...	Lightning	✓
5				6/30/2025, 12:18 A...	Lightning	✓
6				6/30/2025, 12:16 A...	Lightning	✓
7				6/30/2025, 12:16 A...	Classic	
8				7/2/2025, 10:52 AM	Classic	
9				6/30/2025, 12:16 A...	Connected	
10				7/1/2025, 5:43 AM	Connected (Managed)	
11				7/1/2025, 5:43 AM	Connected (Managed)	
12				6/30/2025, 12:16 A...	Lightning	✓
13				7/1/2025, 5:44 AM	Connected (Managed)	

Step 3:

Click **New External Client App**.

Step 4:

Update the basic information as needed, such as the external client app name and your contact email address.

Note: The email address provided must be valid, as **Salesforce** will use it to communicate with your team regarding any updates or issues related to your application usage.

Basic Information

* External Client App Name my external app	* API Name my_external_app
* Contact Email [redacted]	* Distribution State Local
Contact Phone Enter a phone number...	Info URL Enter a URL...
Logo Image URL Enter a URL... Choose one of our sample logos.	Icon URL Enter a URL... Choose one of our sample logos.
Description [redacted]	

> API (Enable OAuth Settings)

Cancel Create

Step 5:

Under **API (Enable OAuth Settings)**, click **Enable OAuth**.

API (Enable OAuth Settings)

☐ Enable OAuth

Step 6:

Under **App Settings**, in the **Callback URL** box, enter the URL below:

https://<base-url>/callback

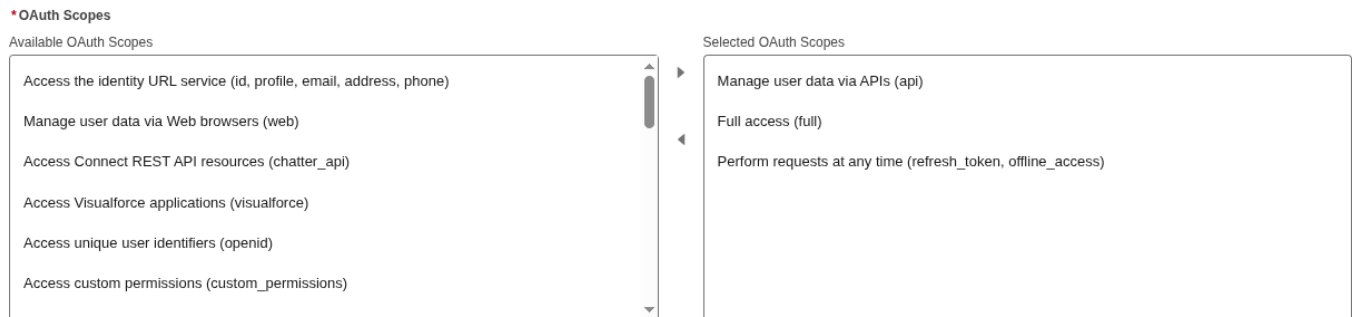
* Callback URL

https://[redacted].my.salesforce.com/callback

Step 7:

In the **OAuth Scopes** section, select these scopes:

- **Manage user data via APIs (api)** - Gives you access to user data.
- **Perform requests at any time (refresh_token, offline_access)** - Permits you to get an OAuth access token.
- **Full access (full)** - grant all access to the permission for integration.



Step 8:

(Required for JWT) In the **Flow Enablement** section, select **Enable Client Credentials Flow** and **Enable JWT Bearer Flow**.

- **Enable Client Credentials Flow** - Allows your app to exchange its client credentials for an access token. And be able to access the credential Client ID.
- **Enable JWT Bearer Flow** - A secure, server-to-server authentication method used to integrate external applications with Salesforce without requiring manual user login.

Flow Enablement

- ☒ Enable Client Credentials Flow
- ☐ Enable Authorization Code and Credentials Flow
- ☐ Enable Device Flow
- ☒ Enable JWT Bearer Flow
- ☐ Enable Token Exchange Flow

Step 9:

(Required for JWT) Click **Upload Files** and upload your digital certificate file (**server.crt**).

☒ Enable JWT Bearer Flow

Certificate Upload

 Upload Files Or drop files

Select a certificate

Step 10:

In **Security** section check the following:

- Require secret for Refresh Token Flow
- Enable Refresh Token Rotation

Security

- ☒ Require secret for Web Server Flow
 - ☒ Require secret for Refresh Token Flow
 - ☐ Require Proof Key for Code Exchange (PKCE) extension for Supported Authorization Flows
 - ☒ Enable Refresh Token Rotation
 - ☐ Issue JSON Web Token (JWT)-based access tokens for named users
-

Step 11:

Click **Create** and **Edit** to configure additional settings.

Step 12:

(Required for JWT) Click the **Policies** tab and configure the following:

- Open **OAuth(Open Authorization) Policies**
- In the **Plugin Policies** section, set **Permitted Users** to **Admin approved users are pre-authorized**
- **(Optional) In OAuth Start URL** use your organization base URL example: (https://fun-dream-996.my.salesforce.com/)
- Click **OK**
- In the **App Policies** section, select the profiles and permission sets that are pre-authorized to use this external client app

Plugin Policies

* Permitted Users

OAuth Start URL

Enter a URL...

All users can self-authorize

All users can self-authorize

Admin approved users are pre-authorized

Apex Plugin Class

Search classes

ed. You can define them [here](#) before assigning to your app.

▼ App Policies

* Start Page ⓘ

None

Select Profiles

Available Profiles

Analytics Cloud Integration User

Chatter Free User

Analytics Cloud Security User

Selected Profiles

System Administrator

Step 14:

In the **OAuth Flows and External Client App Enhancements** section **Enable Client Credentials Flow** and add username.

OAuth Flows and External Client App Enhancements

☒ Enable Client Credentials Flow

* Run As (Username)

00000000-3dvj@force.com

Step 15:

In the **App Authorization** section, under **OAuth(Open Authorization) Policies**, click **Expire refresh token after a specific time**.

Step 16:

Configure token expiration settings:

- **Refresh Token Validity Period:** Enter 365
- **Refresh Token Validity Unit:** Select **Day(s)**

Step 17:

In the **Session Timeout in Minutes** box, enter **15**.

Step 18:

In IP Relaxation Select **Relax IP restrictions** and the Click **Save**.

The screenshot shows the 'App Authorization' configuration page. On the left, under 'Refresh Token Policy', the option 'Expire refresh token after specific time' is selected. Below this, the '* Refresh Token Validity Period' is set to '365' and the '* Refresh Token Validity Unit' is set to 'Day(s)'. On the right, under '* IP Relaxation', the dropdown menu is set to 'Relax IP restrictions'. Below this, 'Enable Single Logout' and 'High Assurance Session Required' are unchecked. The 'Session Timeout In Minutes' is set to '15'.

Step 19:

Enable Allow Access to External Client App Consumer Secret via REST API

External Client App Setting > Allow Access to External Client App Consumer Secret via REST API

The screenshot shows the 'External Client App Settings' page. A search bar at the top left contains the text 'external'. A red box labeled '1' highlights the 'Setup' icon. A red box labeled '2' highlights the 'Settings' link in the left sidebar. A red box labeled '3' highlights the 'Allow access to External Client App consumer secrets via REST API' toggle, which is currently turned 'On'.

Step 20:

Enable Event log files

Event Monitoring Settings > Generate event log files

The screenshot shows the Salesforce 'Event Monitoring Settings' page. A search bar at the top left contains 'event monitoring Settings' (1). The left sidebar shows 'Event Monitoring' expanded, with 'Event Monitoring Settings' selected (2). The main content area has a 'Generate event log files' toggle switch turned on (3). Other settings like 'Delete event monitoring data', 'Enable Lightning Logger Events', and 'Enable Test Threat Detection Events (Beta)' are shown as 'Off'.

Your external client app is now ready to use.

How to Find Client ID (Consumer Key) in External Client App

- type **external** in quick find search bar and click **external client app manager**
- under **External Client App Name** locate the app you created earlier and click it.

The screenshot shows the Salesforce 'External Client App Manager' page. A search bar at the top left contains 'external' (1). The left sidebar shows 'Apps' expanded, with 'External Client Apps' selected, and 'External Client App Manager' highlighted (2). The main content area shows a table with 1 item. The 'External Client App Name' column contains 'Testing_JWT' (3). The 'Contact Email' column contains 'sged@nash1000@gmail.com'. The 'App Authorization' column contains 'Admin approved users are pre-authorized'.

- under settings tab click **OAuth Settings** then you can the view your **client key** and **client secret** after the verification process.



Manage External Client Apps

Testing_JWT

Contact Email

...@gmail.com

App Authorization

Admin approved users are pre-authorized

Type

Local

App Status

Enabled

Policies

Settings

Package Defaults

Configure policies to customize the external client app and plugins for this Salesforce organization

▼ App Policies

▼ OAuth Settings

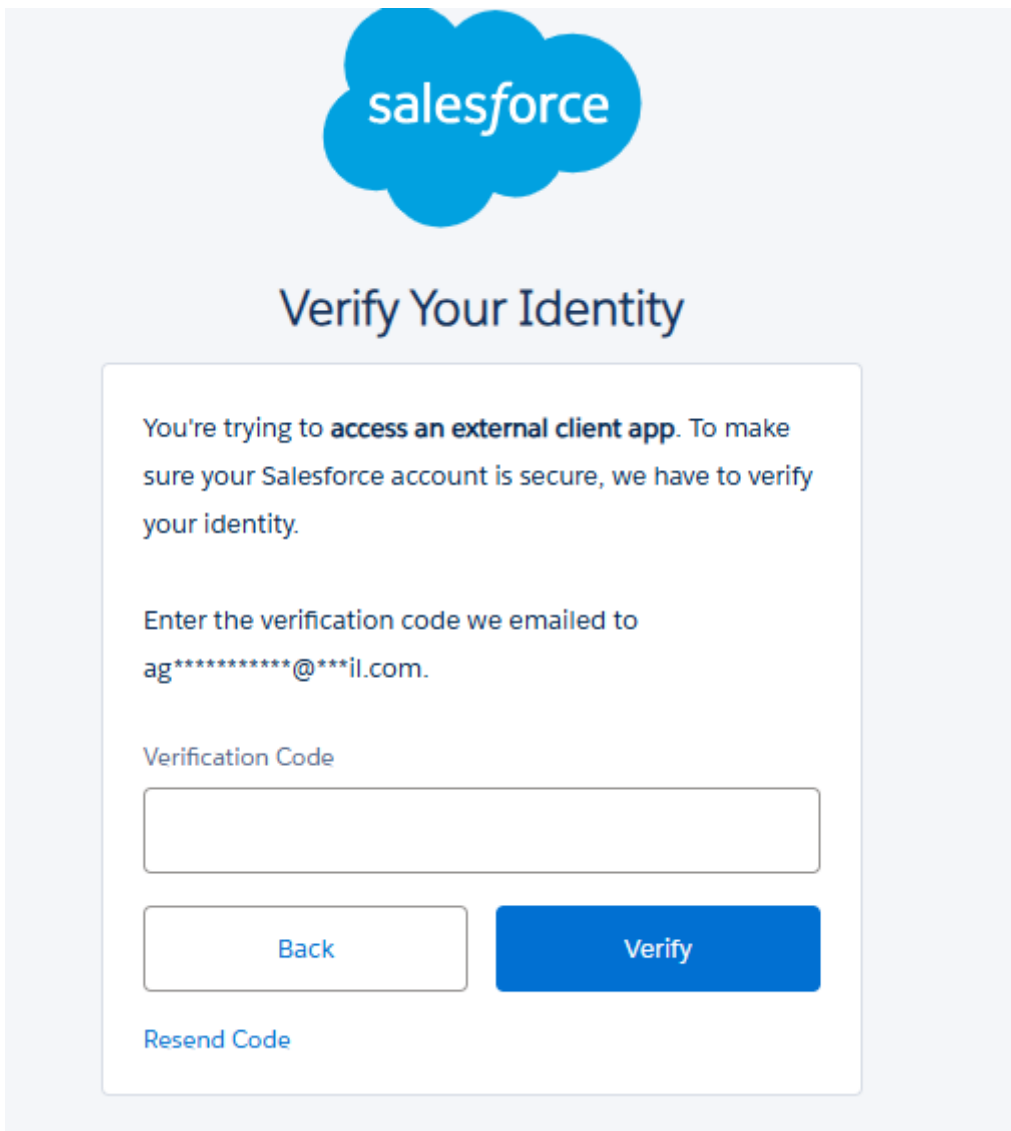
App Settings

Consumer Key and Secret [↗](#)

* Callback URL

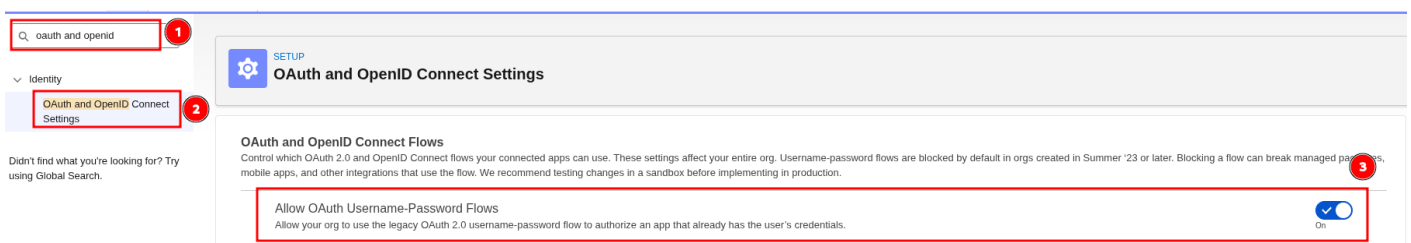
http://localhost:1717/OauthRedirect

* OAuth Scopes



Create A Connected Apps (Username-Password Flow)

For security reasons, Salesforce blocks the OAuth 2.0 Username-Password flow by default in recent releases. Prefer the JWT bearer flow. If you must use the Username-Password flow, in **OAuth and OpenID Connect Settings**, select **Allow OAuth Username-Password Flows**. For more information, see the Salesforce release note: [Username-Password OAuth flow blocked by default](#).



- Log in to Salesforce (Lightning UI).
- From **Setup**, in **Quick Find** enter **External Client Apps** and select **Settings**. Turn on **Allow creation of connected apps**. To create a connected app, select **New Connected App**.

External Client App Settings

External Client Apps
Integrate third-party apps with Salesforce using APIs and standard protocols, such as SAML, OAuth, and OpenID Connect using External Client Apps. These settings apply to all External Client Apps configured in your org.

Allow access to External Client App consumer secrets via Metadata API
Allow the View External Client App Consumer Secrets in Metadata user permission to be assigned to users. Users with this permission can view External Client App consumer secrets in Metadata API responses. If this setting is disabled and you want to edit it, contact Salesforce Customer Support.

Allow access to External Client App consumer secrets via REST API
Allow users to access External Client App consumer secrets through the OAuth credentials resources for Connect REST API.

Connected Apps
Use connected apps to integrate third-party apps with Salesforce through APIs and standard protocols, such as SAML, OAuth, and OpenID.

[Learn more about Connected Apps and External Client Apps.](#)

[New Connected App](#)

- Fill **Basic Information**: **Connected App Name**, **API Name**, **Contact Email**.

New Connected App

Save
Cancel

Basic Information

Connected App Name

API Name

Contact Email

Contact Phone

Logo Image URL

Icon URL

Info URL

Description

- In **API (Enable OAuth Settings)**, check **Enable OAuth Settings**.
- **Callback URL**:
 - Web apps: your app callback (for example, `https://yourapp.example.com/callback`).
 - Not used by the JWT or Username-Password flows, but Salesforce requires a value; you can enter your instance URL.
- **Select OAuth scopes**:
 - **Manage user data via APIs (api)**
 - **Perform requests at any time (refresh_token, offline_access)**
 - **Full access (full)**
 - **Enable Client Credentials Flow**
 - **Enable Refresh Token Rotation**

▼ API (Enable OAuth Settings)

Enable OAuth Settings ☒

Enable for Device Flow ☐

Callback URL

Use digital signatures ☐

Selected OAuth Scopes

Available OAuth Scopes

- Access Agent-to-Agent API (a2a_api)
- Access Analytics REST API Charts Geodata resources (eclair_api)
- Access Analytics REST API resources (wave_api)
- Access Connect REST API resources (chatter_api)
- Access Einstein GPT services (einstein_gpt_api)
- Access Headless Forgot Password API (forgot_password)
- Access Headless Passwordless Login API (pwdless_login_api)
- Access Headless Registration API (user_registration_api)
- Access Interaction API resources (interaction_api)
- Access Lightning applications (lightning)

Add

Remove

Selected OAuth Scopes

- Full access (full)
- Manage user data via APIs (api)
- Perform requests at any time (refresh_token, offline_access)

Require Proof Key for Code Exchange (PKCE) Extension for Supported Authorization Flows ☒

Require Secret for Web Server Flow ☒

Require Secret for Refresh Token Flow ☒

Enable Client Credentials Flow ☒

Enable Authorization Code and Credentials Flow ☐

Enable Token Exchange Flow ☐

Enable Refresh Token Rotation ☐

Issue JSON Web Token (JWT)-based access tokens for named users ☐

Limit Idle Refresh Token Time-to-Live (TTL) to 30 Days ☐

- Click **Save**. It can take up to 10 minutes for the Connected App to propagate.
- After saving, open **Manage Consumer Details** to obtain **Consumer Key** and **Consumer Secret**.

Manage Consumer Details Appears only once so better to copy consumer key and consumer secret in a safe place.

SETUP

Manage Connected Apps

Review and lock security controls. [Review Controls](#)

Connected App Name
Cytech_Connected_App

« Back to List: Custom Apps

2

Edit Delete **Manage** Migrate to External Client App

Version 1.0

API Name Cytech_Connected_App

Created Date 6/9/2026, 5:54 AM

By: [benjie Janlay](#)

Contact Email [benjie.janlay@cytech.com](#)

Contact Phone

Last Modified Date 6/9/2026, 7:02 AM

By: [benjie Janlay](#)

Description

Info URL

▼ API (Enable OAuth Settings)

Consumer Key and Secret **1** **Manage Consumer Details**

Selected OAuth Scopes

- Manage user data via APIs (api)
- Full access (full)
- Perform requests at any time (refresh_token, offline_access)

Callback URL

Enable for Device Flow ☐

Require Proof Key for Code Exchange (PKCE) Extension for Supported Authorization Flows ☒

Require Secret for Web Server Flow ☒

Require Secret for Refresh Token Flow ☒

Enable Client Credentials Flow ☒

Enable Authorization Code and Credentials Flow ☐

Enable Token Exchange Flow ☐

Enable Refresh Token Rotation ☐

- Then Click Manage to OAuth Policies
 - Permitted Users > All users may self-authorize
 - IP Relaxation > Relax IP Restrictions
 - Refresh Token Policy > Expire refresh token after "365" days
 - Client Credentials Flow > "the user you created earlier"

Verify if LoginEvent is enable in Quick find > Event Manager > enable all

Hyperforce Assistant

- Events
 - Event Manager
 - Event Relays
 - Event Studio
- Integrations
 - Platform Events
- Privacy Center
 - Consent Event Stream
- Identity
 - Identity Provider Event Log
- Security
 - Event Monitoring
 - Event Log File Browser
 - Event Monitoring Settings
 - Transaction Security Policies

Didn't find what you're looking for? Try using Global Search.

Name	Subscription Channel	Type	Description	Streaming Data	Storing Data	
API Anomaly Event	/event/ApiAnomalyEvent	Event Monitoring	Track anomalies in how users make API calls	✓	✓	⌵
API Event	/event/ApiEventStream	Event Monitoring	Track user API queries in your org	✓	✓	⌵
Bulk API Result Event	/event/BulkApiResultEvent	Event Monitoring	Track when a user downloads the results of a Bulk API request	✓	✓	⌵
Concurrent Long Running Apex Error Event	/event/ConcurLongRunApexErrEvent	Event Monitoring	Track when a Concurrent Long Running Apex error has occurred	✓	Not applicable	⌵
Credential Stuffing Event	/event/CredentialStuffingEvent	Event Monitoring	Track when a user successfully logs in to Salesforce during an identified credential stuffing attack	✓	✓	⌵
File Event	/event/FileEvent	Event Monitoring	Track file activity. For example, track when a user downloads or previews a file	✓	✓	⌵
Guest User Anomaly Event	/event/GuestUserAnomalyEvent	Event Monitoring	Tracks data access anomalies that are caused by guest user misconfiguration	✓	✓	⌵
Identity Provider Event	Not applicable	Event Monitoring	Track identity provider activities	Not applicable	✓	⌵

- JWT Authentication Client Key Path (full file folder path of server.pem not in root directory)
 - ex: Users/jdoe/JWT/server.pem
- Username (can be found in View Profile > Settings > Personal Information) or If you have dedicated user for integration can be found in (Quick Find > Users > Username)
 - example format: ADMIN-3dvj@force.com
- Client ID (Consumer Key)
 - example format:
3MVxxxxxCx.CV6cbh7fSpKs_5iexxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxZKbaepcxIJUhO1
- Instance URL
 - example format: <https://company.my.salesforce.com>
- API Version
 - in quick find type **Apex Classes**

Developer Console					New	G
Action	Name ↑	Namespace Prefix	Api Version	Status		
Edit Security	DeveloperEditionUtils	devedapp	64.0	Active		
Edit	DeveloperEditionUtilsTest	devedapp	64.0	Active		
Edit Security	PostInstallScript	devedapp	64.0	Active		
Edit	PostInstallScriptTest	devedapp	64.0	Active		

For Username-Password Flow:

- Username
- Password
- Consumer Key
- Consumer Secret
- Instance URL

Provide this required fields to [CyTech Support](#).

Reference Link:

[Create an External Client App in Your Org | Salesforce DX Developer Guide |](#)

[Salesforce Developers](#)

If you need further assistance, kindly contact our support at support@cytechint.com for prompt assistance and guidance.

Revision #28

Created 11 February 2026 13:14:08 by Benjie Janlay Jr.

Updated 17 June 2026 08:40:30 by Benjie Janlay Jr.