OAuth SSO

Overview

JFrog Platform is integrated with OAuth allowing you to delegate authentication requests to external providers and let users login to the system using their accounts with those providers.

Currently, the provider types supported are **Google**, **OpenID Connect**, **GitHub Enterprise,** and **Cloud Foundry UAA.** You may define as many providers of each type as you need.

From December 2021, JFrog Cloud users (only) can also join through an invite, and to then log in using Personal OAuth such as Google or GitHub. See Enabling and Disabling Personal OAuth SSO for details.

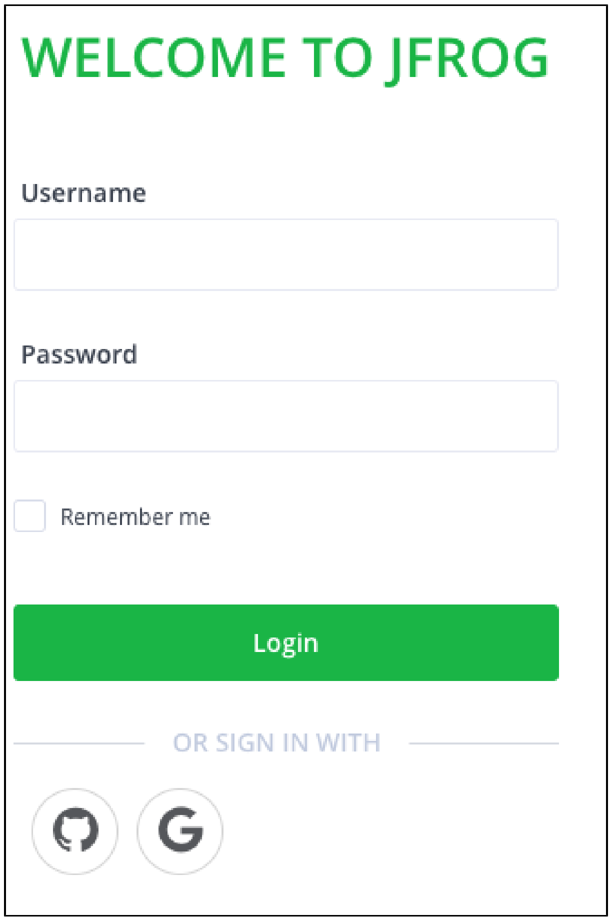
WebUI Changes implemented in Artifactory 7.38.x and above

Security is now called Authentication Providers. All the relevant text and images on this page have been updated to reflect this change.

Page Contents

Usage

When OAuth is enabled in the JFrog Platform, users may choose to sign in through any of the supported OAuth providers. To log in through a provider, simply click on the provider's button in the login screen.



You will be redirected to the login screen of the corresponding provider.

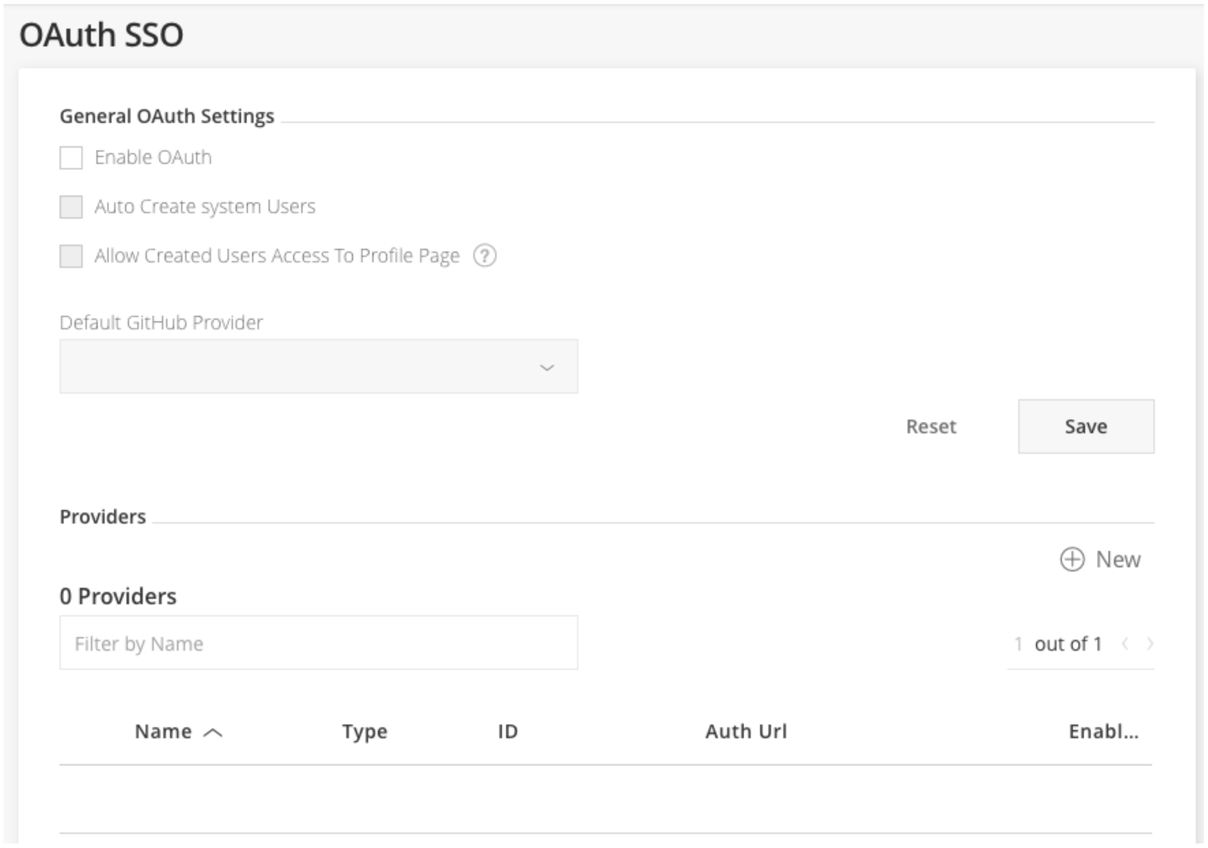
If you are already logged in to any of that provider's applications you will not need to log in again, but you may have to authorize the JFrog Platform to access your account information, depending on the provider type.

Configuring OAuth

WebUI Changes implemented in Artifactory 7.38.x and above

Security is now called Authentication Providers. All the relevant text and images on this page have been updated to reflect this change.

To access OAuth integration settings, in the **Administration** module, select **Authentication Providers** | **OAuth SSO**.


OAuth SSO

General OAuth Settings

☐ Enable OAuth

☐ Auto Create system Users

☐ Allow Created Users Access To Profile Page

Default GitHub Provider

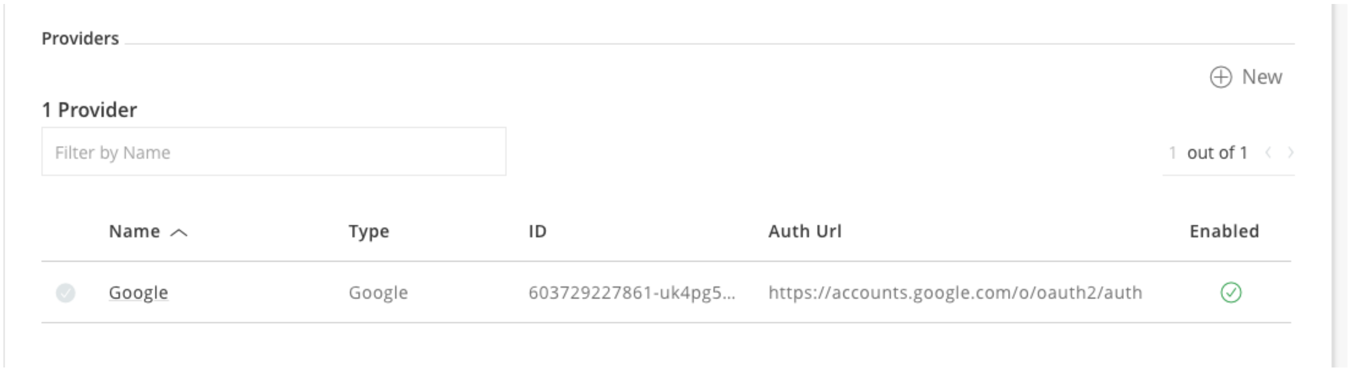
Enable OAuth	When selected, authentication with an OAuth provider is enabled and the system will display all OAuth providers configured. If not checked, authentication is by Artifactory user/password.
Auto Create System Users	When set, the system will automatically create new users for those who have logged in using OAuth, and assign them to the default groups.
Default Provider	Specifies the provider through which different clients (such as npm, for example) should authenticate their login to gain access to the JFrog Platform. **Default provider** Currently, only a GitHub Enterprise OAuth provider may be defined as the Default Provider.
Allow Created Users Access To Profile Page	When selected, users created after authenticating using OAuth, will be able to access their profile. This means they are able to generate their API Key and set their password for future use.

Custom URL base

For your OAuth settings to work, make sure you have your Custom Base URL configured.

Adding a New Provider

The list of providers defined in Artifactory is displayed in the **Providers** section.


Providers

New

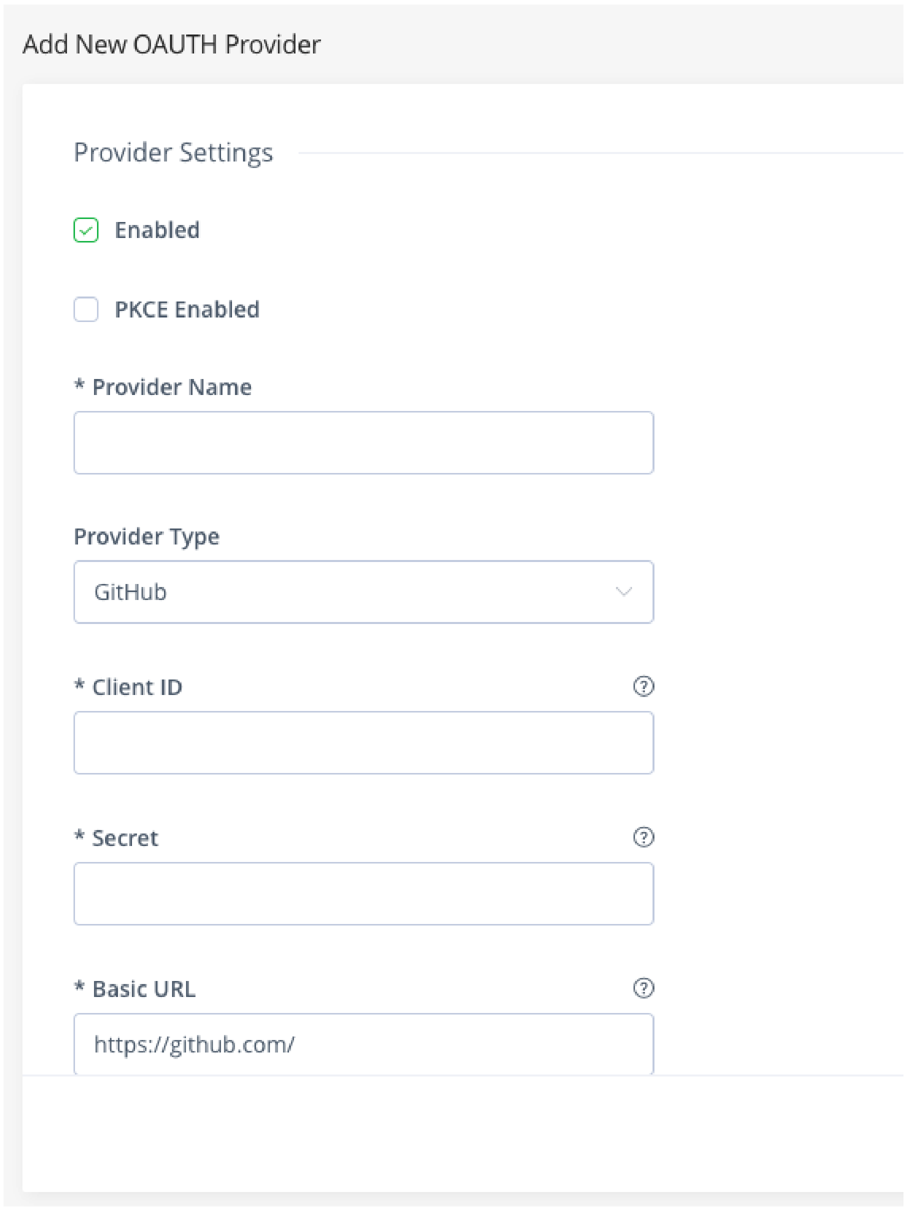
1 Provider

Filter by Name

1 out of 1

	Name	Type	ID	Auth Url	Enabled
	Google	Google	603729227861-uk4pg5...	https://accounts.google.com/o/oauth2/auth	✓

To add a new provider, click **New**. The JFrog Platform displays a dialog letting you enter the provider details. These may vary slightly depending on the provider you are configuring.


Add New OAUTH Provider

Provider Settings

☑ Enabled

☐ PKCE Enabled

* Provider Name

Provider Type: GitHub

* Client ID

* Secret

* Basic URL: https://github.com/

The following table describes the settings required by each supported provider, and the corresponding values you should use (where available):

	Description	GitHub.com	GitHub Enterprise	Google	Cloud Foundry UAA	OpenID
Enabled	If selected, this OAuth provider is enabled and will appear in the login dialog.					
PKCE Enabled	If selected, this OAuth provider will use an authorization code flow with PKCE.	X	X			
Provider Name	A logical name for this provider. For example, "Google OAuth", "GitHub OAuth". This must be unique within JFrog Platform.					
Provider Type	The provider type. Currently **GitHub**, **Git Enterprise, Google**, **OpenID**, and **Cloud Foundry** are supported.					
Client ID	The identity with which you identify your organization to the provider. This is provided by the OAuth provider when you set up your account with them.					
Secret	The secret allocated to your organization by the provider. This is provided by the OAuth provider when you set up your account with them.					
Domain	Specifies a domain filter which defines from which domains users may be authenticated. Normally, this will be your domain name. For example `jfrog.com`	X	X		X	X
Docker Login	Support for Docker login	X		X	X	X
npm Login	Support for npm login	X		X	X	X
Basic URL	The base URL of the Git server which should be used for authentication.	*https://github.com/*	*`<Server Base URL>`*	X	X	X
Auth URL	The URL through which the provider redirects you to the authentication page.	https://github.com/login/oauth/authorize Note: GitHub.com Accounts Any GitHub.com account that has access to the Artifactory URL will be allowed to login, including accounts that are outside your GitHub.com organization scope.	*`<Server Base URL>/login/oauth/authorize`*	*`https://accounts.google.com/o/oauth2/auth`*	*`<Server Base URL>/oauth/authorize`*	
API URL	The URL through which Artifactory can get extra information that it not directly available via OAuth.	*https://api.github.com/user*	*`<Server Base URL>/api/v3/user`*	*`https://www.googleapis.com/oauth2/v1/userinfo`*	*`<Server Base URL>/userinfo`*	
Token URL	The URL that Artifactory will go to to get a token to use the API.	*`https://github.com/login/oauth/access_token`*	*`<Server Base URL>/login/oauth/access_token`*	*`https://www.googleapis.com/oauth2/v3/token`*	*`<Server Base URL>/oauth/token`*	

Using Query Params

You may pass query params along with the Authorization URL.

Example

```
https://github.com/login/oauth/authorize?realm=Employees
```

Multiple query params should be separated with an ampersand.

Example

```
https://github.com/login/oauth/authorize?realm=Employees?client_id=XXXXXXXXXXX&scope=openid%20profile%20email
```

Enabling Authorization Code Flow with PKCE

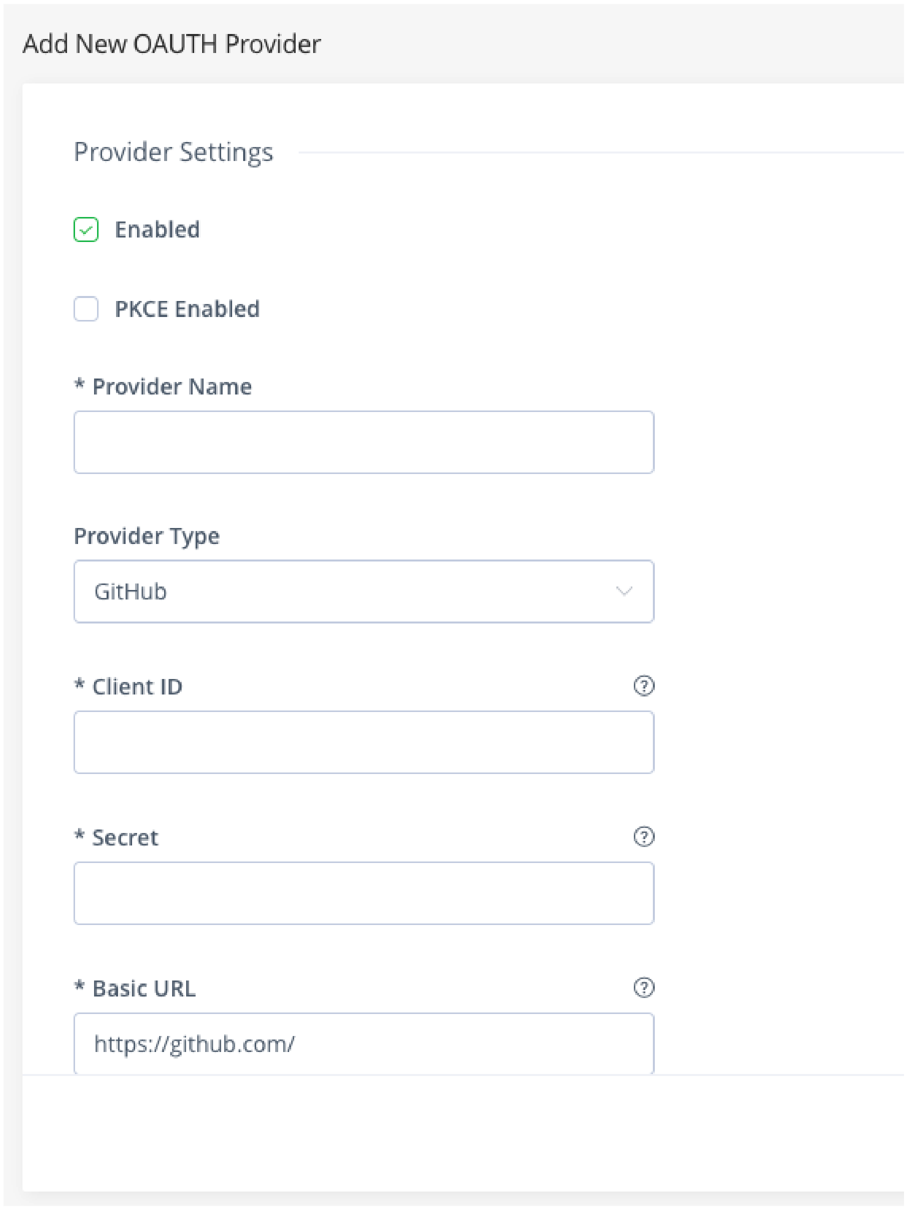
WebUI Changes implemented in Artifactory 7.38.x and above

Security is now called Authentication Providers. All the relevant text and images on this page have been updated to reflect this change.

From Artifactory version 7.38, you can apply the Authorization Code Flow with PKCE (Proof Key for Code Exchange) flow for your OAuth Provider authorization provider. This flow is the alternative to the usage of Secrets, which is automatically disabled once you enable PKCE on your OAuth provider.

To enable PKCE on your OAuth provider:

1. In the Administration module, navigate to the **Administration** module, and select **Authentication Providers** | **OAuth SSO**.
2. Scroll to the `Providers` *section* and click `New` or edit an existing provider.



Add New OAUTH Provider

Provider Settings

✓ Enabled

PKCE Enabled

* Provider Name

Provider Type: GitHub

* Client ID

* Secret

* Basic URL: https://github.com/

3. Select the `PKCE Enabled` check box.
   As a result, the `Secret` field is disabled.

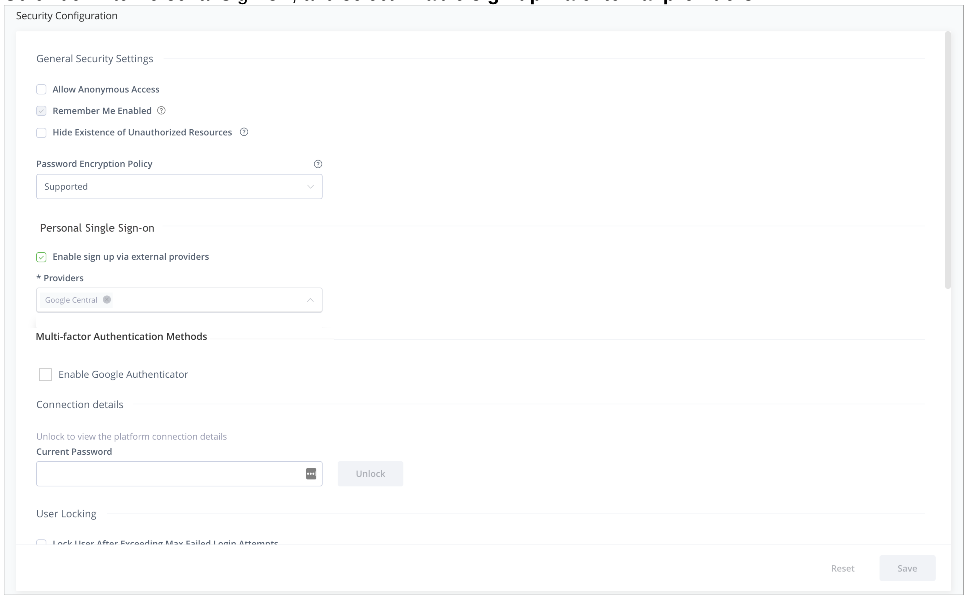
Enabling and Disabling Personal OAuth SSO

WebUI Changes implemented in Artifactory 7.38.x and above

Identity & Access is now called User Management. All the relevant text and images on this page have been updated to reflect this change.

1. To enable Personal OAuth SSO, in the **Administration** module, select **User Management** | **Security**.

2. Scroll down to Personal Sign On, and select **Enable sign up via external providers**.



Security Configuration

General Security Settings

- [ ] Allow Anonymous Access
- [✓] Remember Me Enabled
- [ ] Hide Existence of Unauthorized Resources

Password Encryption Policy: Supported

Personal Single Sign-on

- [✓] Enable sign up via external providers

* Providers: Google Central

Multi-factor Authentication Methods

- [ ] Enable Google Authenticator

Connection details

Unlock to view the platform connection details

Current Password

Unlock

User Locking

- [ ] Lock User After Exceeding Max Failed Login Attempts

Reset Save

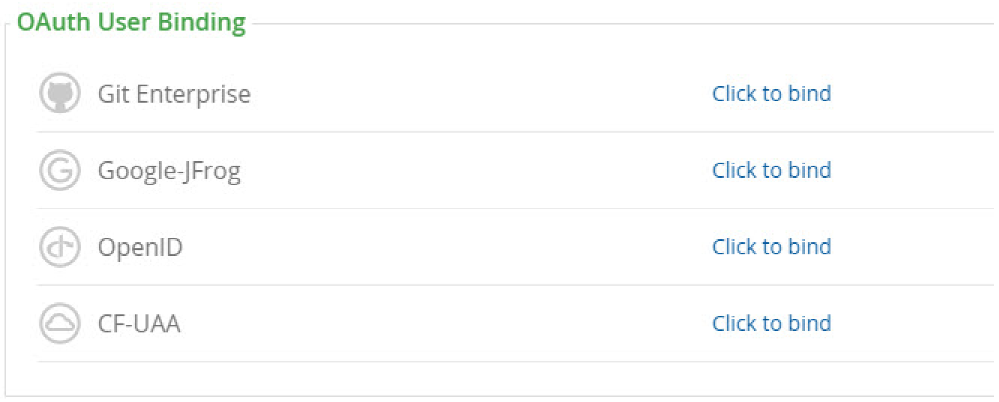
3. Then, from the Providers dropdown list, select the providers you wish to enable in the sign up screen, for example Google, GitHub, etc.

Binding Existing User Accounts

To log in using any of your OAuth provider accounts If you already have an **internal (not external realms such as LDAP, SAML...)** account in the system, bind your JFrog Platform account to the corresponding account.

To bind your account, go to your Profile page and enter your JFrog Platform password to unlock it.

Under **OAuth User Binding**, select **Click to bind** next to the OAuth provider you wish to bind to.



OAuth User Binding

Git Enterprise	Click to bind
Google-JFrog	Click to bind
OpenID	Click to bind
CF-UAA	Click to bind

Creating OAuth Provider Accounts

In order to use OAuth authentication, you need to set up an account with each OAuth provider you wish to use in order to get the various parameters (such as Provider ID and Secret) you will need to set up OAuth integration in the system.

GitHub OAuth Setup

Caution: Access to GitHub.com Accounts

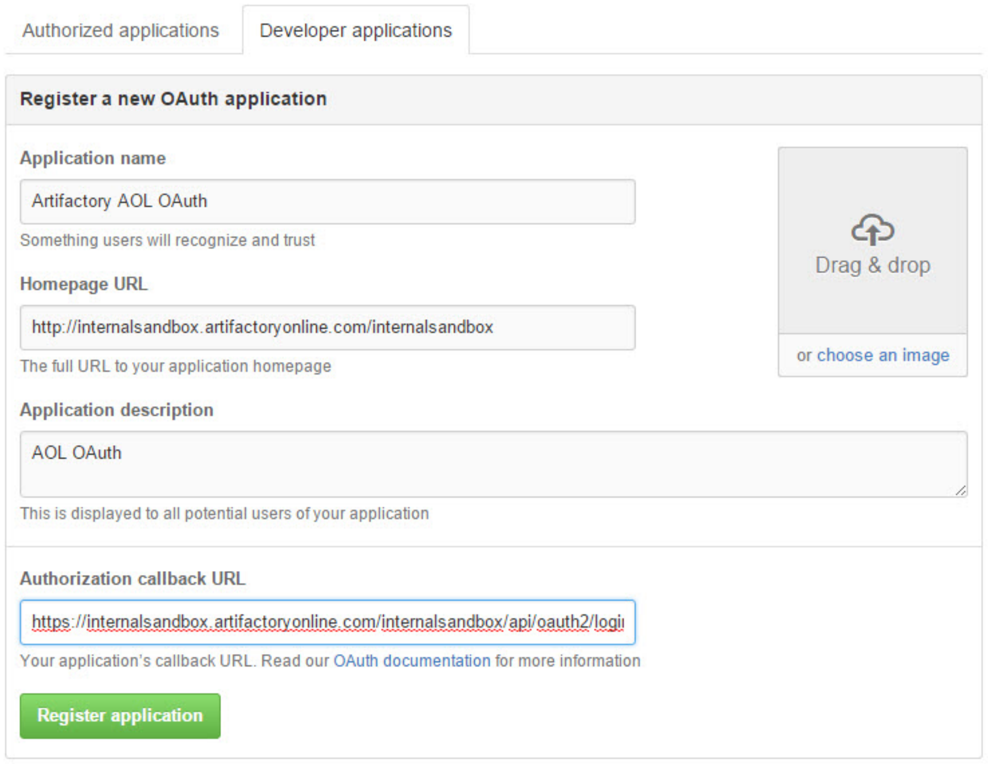
Any GitHub.com account that has access to the JFrog Platform URL will be allowed to login, including accounts that are outside your GitHub .com organization scope. This does not apply to GitHub Enterprise.

To set up your OAuth account on GitHub:

1. Log in to your GitHub account. Under your personal profile settings, select **Applications** and click the Developer Applications tab.

2. Click **Register new application.**
3. Set the Application name. For example, the JFrog Platform Cloud OAuth.
4. Set the **Homepage Url**. This is your Artifactory server host URL (`https://<artifactory-server>/`).
   For example, *`https://mycompany.jfrog.io/mycompany/`*
5. Set the **Authorization Callback URL** as follows:

   a. For on-prem installation: *`http://<server_host>/artifactory/api/oauth2/loginResponse`*
      For example, *`http://mycompany.artifactory.com/artifactory/api/oauth2/loginResponse`*
   b. For cloud: *`https://<server_name>.jfrog.io/artifactory/api/oauth2/loginResponse`*
      For example, *`https://mycompany.jfrog.io/artifactory/api/oauth2/loginResponse`*
6. Click **Register application** to generate your **Client ID** and **Client Secret**.
   Make a note of these; you will need them to configure OAuth authentication through GitHub on the JPD.



Register a new OAuth application

Application name

Artifactory AOL OAuth

Something users will recognize and trust

Homepage URL

http://internalsandbox.artifactoryonline.com/internalsandbox

The full URL to your application homepage

Application description

AOL OAuth

This is displayed to all potential users of your application

Authorization callback URL

https://internalsandbox.artifactoryonline.com/internalsandbox/api/oauth2/logi

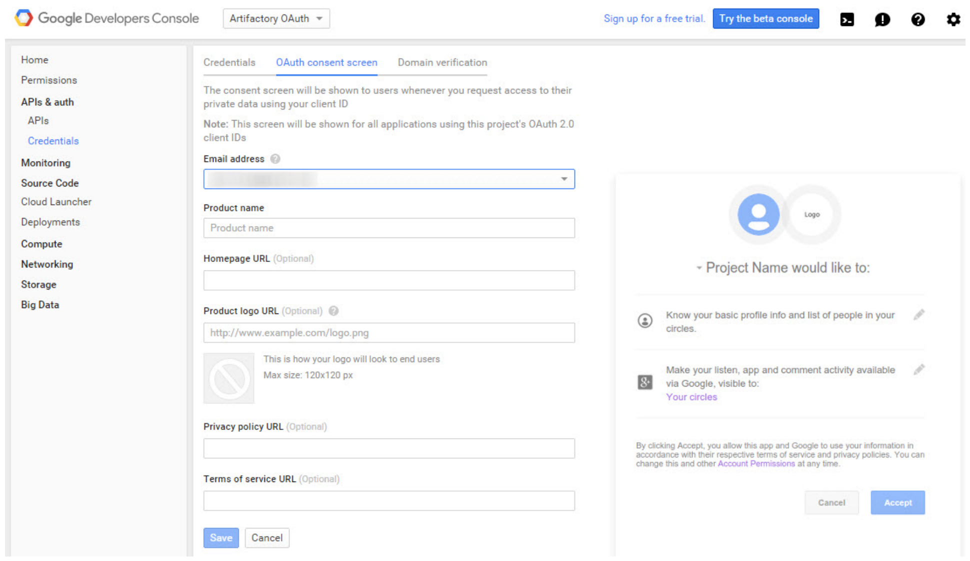
Your application's callback URL. Read our OAuth documentation for more information

Google OAuth Provider Setup

To set up your OAuth account on Google, execute the following steps:

1. Login to Google Developer Console.
2. Create a new project. For example, "Artifactory OAuth".
3. Once the project is created, in the left navigation bar, select **APIs & auth** | **Credentials.**

4. Select the **OAuth consent screen** tab and configure the consent page that the end users will see when logging in with the Google credentials.



Google Developers Console | Artifactory OAuth | Sign up for a free trial. | Try the beta console

Home
Permissions
APIs & auth
APIs
Credentials
Monitoring
Source Code
Cloud Launcher
Deployments
Compute
Networking
Storage
Big Data

Credentials | OAuth consent screen | Domain verification

The consent screen will be shown to users whenever you request access to their private data using your client ID

Note: This screen will be shown for all applications using this project's OAuth 2.0 client IDs

Email address: [redacted]

Product name: Product name

Homepage URL (Optional)

Product logo URL (Optional): http://www.example.com/logo.png

This is how your logo will look to end users
Max size: 120x120 px

Privacy policy URL (Optional)

Terms of service URL (Optional)

Save | Cancel

Logo

Project Name would like to:

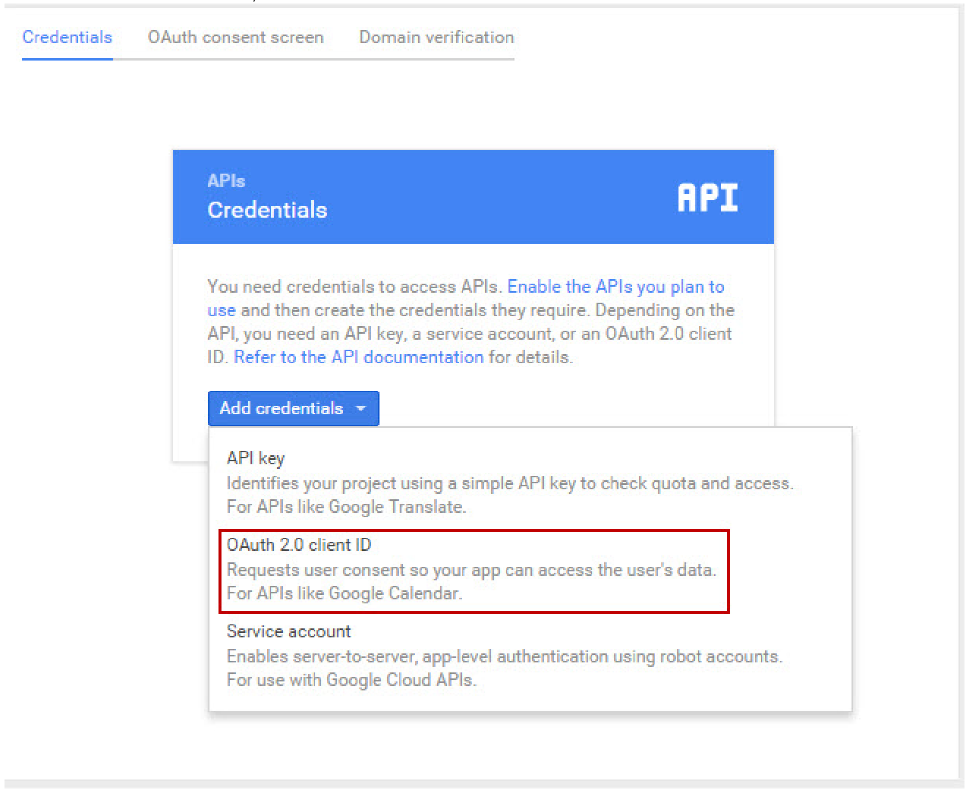
Know your basic profile info and list of people in your circles.

Make your listen, app and comment activity available via Google, visible to:
Your circles

By clicking Accept, you allow this app and Google to use your information in accordance with their respective terms of service and privacy policies. You can change this and other Account Permissions at any time.

Cancel | Accept

5. Back in the **Credentials** tab, Click **Add Credential** and select **OAuth 2.0 client ID.**



Credentials | OAuth consent screen | Domain verification

APIs
Credentials

You need credentials to access APIs. Enable the APIs you plan to use and then create the credentials they require. Depending on the API, you need an API key, a service account, or an OAuth 2.0 client ID. Refer to the API documentation for details.

Add credentials

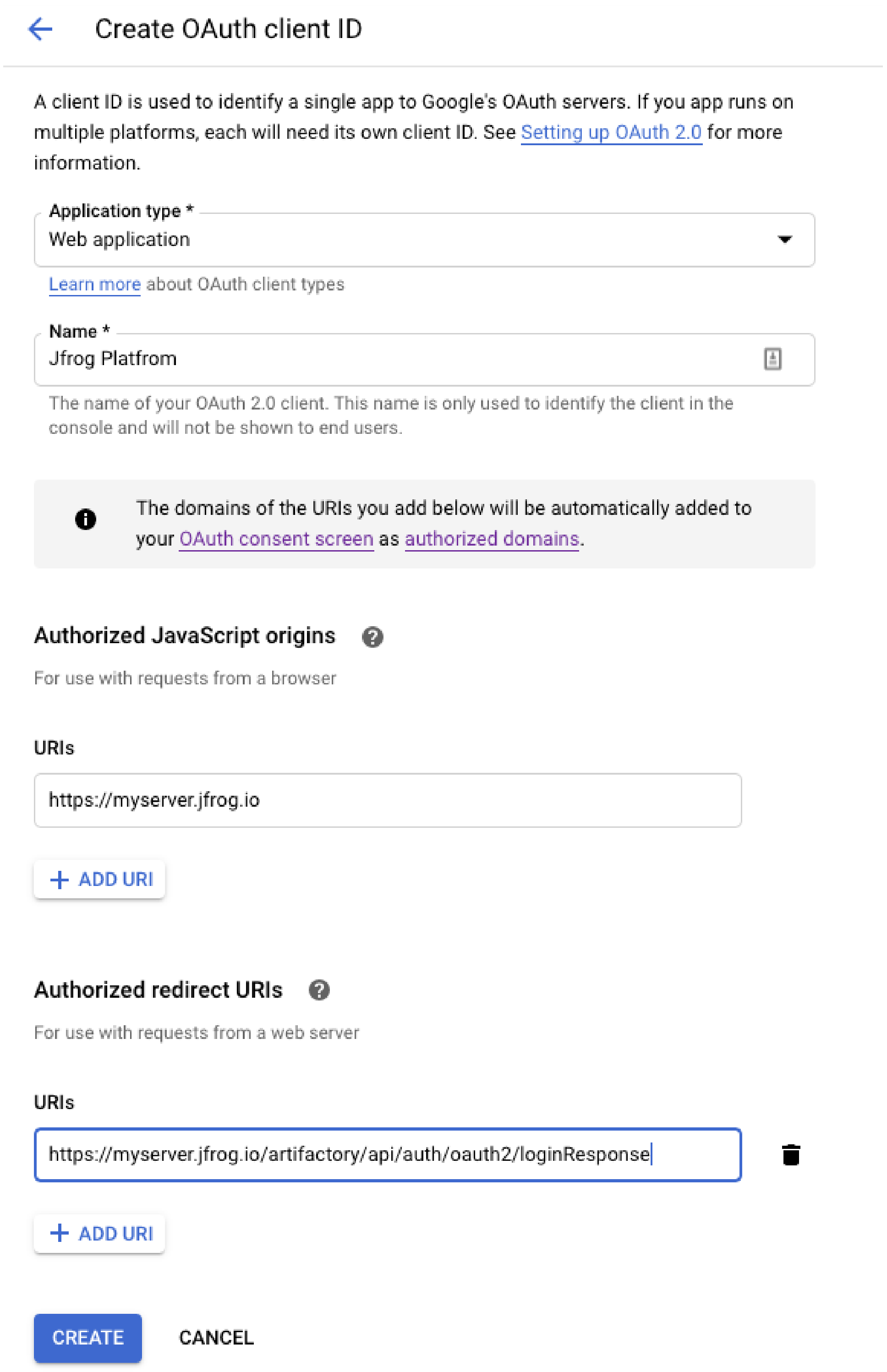
API key
Identifies your project using a simple API key to check quota and access.
For APIs like Google Translate.

OAuth 2.0 client ID
Requests user consent so your app can access the user's data.
For APIs like Google Calendar.

Service account
Enables server-to-server, app-level authentication using robot accounts.
For use with Google Cloud APIs.

6. Under **Create client ID,** select **Web application**.
7. Enter a **Name** and set the **Authorized redirect URIs**
   For on-prem: `https://<server_host>/artifactory/api/auth/oauth2/loginResponse`

For cloud: `https://<server_name>.`*`jfrog.io/artifactory/api/oauth2/loginResponse`*



Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If you app runs on multiple platforms, each will need its own client ID. See Setting up OAuth 2.0 for more information.

Application type *
Web application

Learn more about OAuth client types

Name *
Jfrog Platfrom

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

The domains of the URIs you add below will be automatically added to your OAuth consent screen as authorized domains.

Authorized JavaScript origins

For use with requests from a browser

URIs

https://myserver.jfrog.io

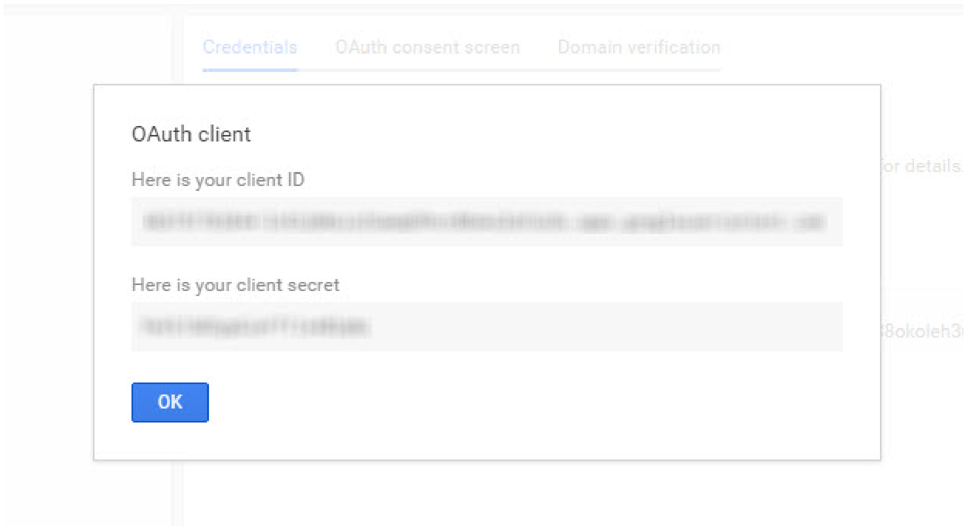
Authorized redirect URIs

For use with requests from a web server

URIs

https://myserver.jfrog.io/artifactory/api/auth/oauth2/loginResponse

8. Click **Create** to generate your **Client ID** and **Client Secret**.

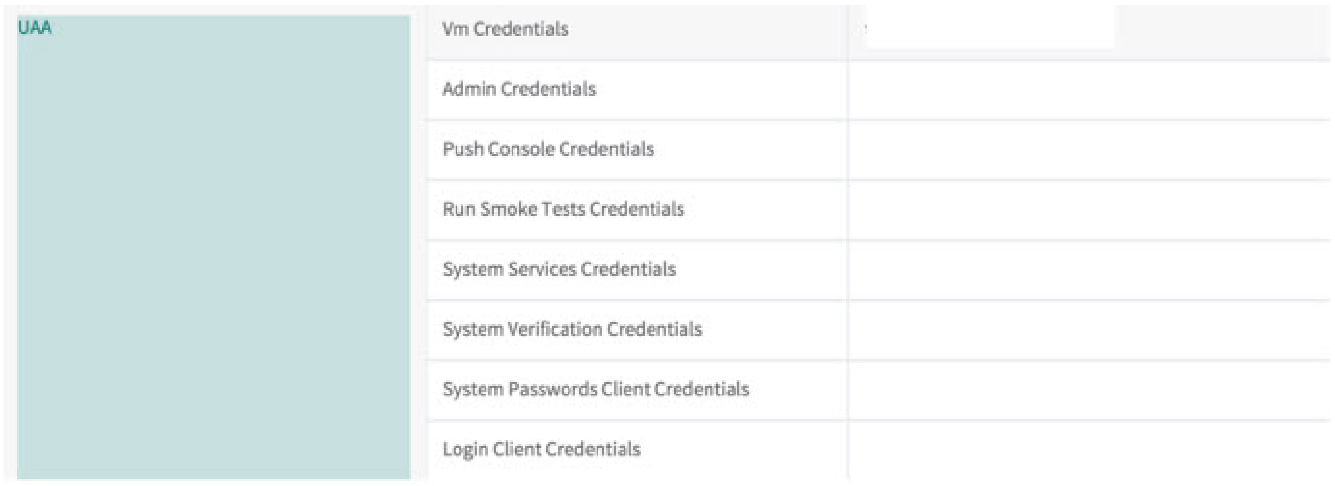


Make a note of these; you will need them to configure OAuth authentication through Google on the JFrog Platform.

Cloud Foundry UAA Setup

OAuth authentication with Cloud Foundry UAA is supported.

To setup your OAuth authentication with Cloud Foundry UAA, fill in the fields as needed.



UAA	Vm Credentials	
	Admin Credentials	
	Push Console Credentials	
	Run Smoke Tests Credentials	
	System Services Credentials	
	System Verification Credentials	
	System Passwords Client Credentials	
	Login Client Credentials	

Using Secure OAuth

To use secure OAuth with a valid certificate from a CA trusted by Java, all you need to do is use a secure OAuth URL in your settings.

If you want to use OAuth with a non-trusted (self-signed) certificate, please see Trusting a new CA or a Self-Signed Certificate.

Using API Key with OAuth Users

While OAuth provides access to the UI, it is also possible for OAuth users to generate an API Key that can be used instead of a password for basic authentication or in a dedicated REST API header, this is very useful when working with different clients, e.g. docker, npm, maven, etc. or using Artifactory REST API.

In order to allow OAuth users access to an API key you will need to make sure that the "**Auto Create System Users**" and "**Allow Created Users Access To Profile Page**" check boxes are checked. This means that OAuth users are also saved in the database and can access their profile page in order to generate, retrieve and revoke their API key.

Using OAuth on High Availability Setup

The OAuth protocol requires the client to give permission to a specific application. JPD will redirect the user to the configured application URL and one permission is granted user will be navigated back.

The limitation on this process when working in High Availability setup is that the user must return to the same node, otherwise the authentication process will fail, in order to achieve this a sticky session configuration should include the `/artifactory/api/oauth2/`.

The example below shows NGINX configuration.

NGINX Reverse Proxy Configuration

```
location ~ (/artifactory/webapp/|/artifactory/ui/|/artifactory/api/oauth2/) {
        proxy_http_version      1.1;
        proxy_pass              http://<UPSTREAM_NAME>;
        proxy_intercept_errors  on;
        proxy_pass_header       Server;
        proxy_connect_timeout   75s;
        proxy_send_timeout      2400s;
        proxy_read_timeout      2400s;
        proxy_set_header        Host $host;
        proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header        X-Forwarded-Proto $http_x_forwarded_proto;
        proxy_set_header        X-Real-IP $remote_addr;
        proxy_set_header        X-JFrog-Override-Base-Url $http_x_forwarded_proto://$host;
}
```