ALIBABA CLOUD

Alibaba Cloud

Identity as a service

SSO Configuration

Document Version: 20220322

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	**Danger:** Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	**Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	**Notice:** If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	**Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Best Practices

1.1. Implement single sign-on for JIRA or Confluence

This topic describes how to use the SAML protocol to implement single sign-on for JIRA or Confluence in the IDaaS console. We demonstrate single sign-on for the Alibaba Cloud console here.

Background

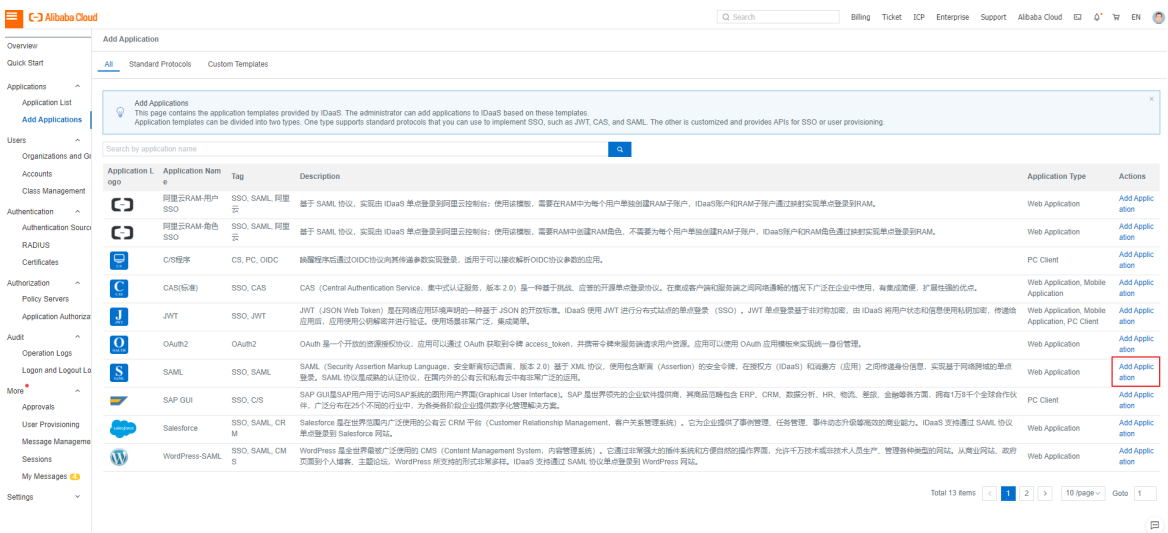
Employees of an enterprise need to access JIRA or Confluence in their daily work. They must enter the logon URLs of applications, account names, and passwords upon each logon. If multiple similar applications are involved, they must record multiple pairs of usernames and passwords and repeated logons are time-consuming.

Solution

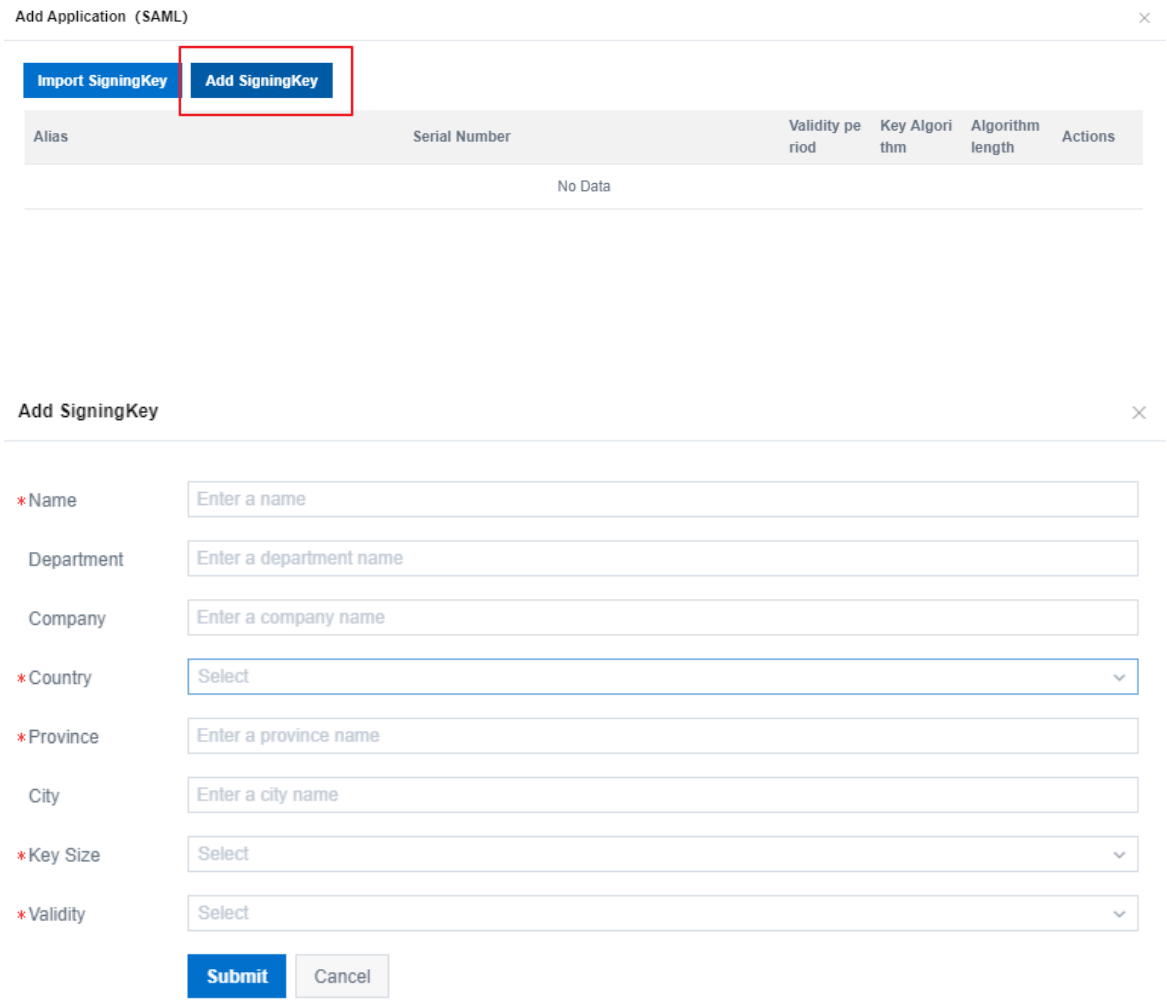
IDaaS can implement single sign-on for JIRA or Confluence. Employees can access all authorized applications through single sign-on.

Procedure

1. Log on to the IDaaS console as an IT administrator. For more information, see Logon in Administrator Guide.
2. In the left-side navigation pane, choose **Applications > Add Applications**. Find the SAML application and click **Add Application** in the Actions column.

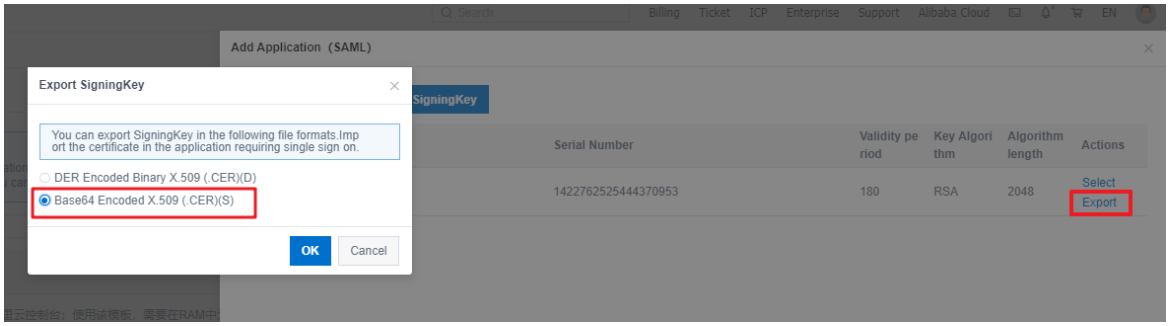


3. Click Add SigningKey. Configure the parameters and click Submit.



4. Export the SigningKey file.

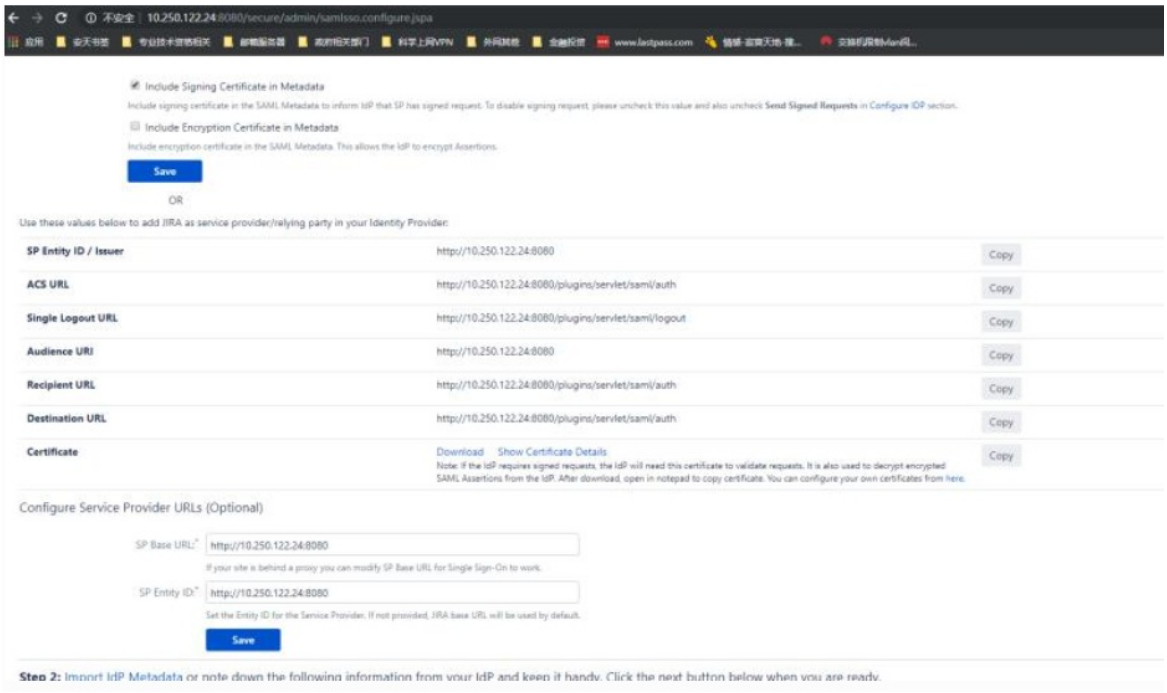
   Find the new SigningKey in the SigningKey list and click Export in the Actions column. Open the exported file in a text editor. Obtain the -- BEGIN CERTIFICATE-- --END CERTIFICATE -- information.



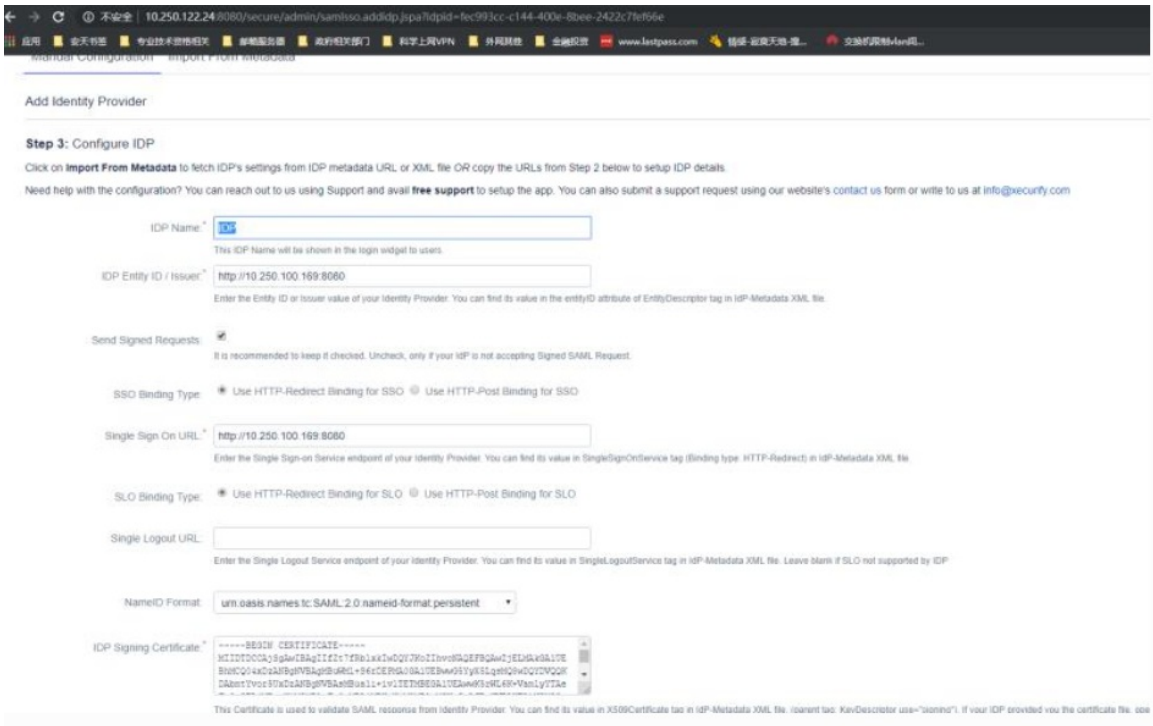
5. Configure miniOrange Single Sign On for Confluence.

   Configure SP Base URL.

   Set SP Base URL and SP Entity ID to your Confluence information. Use the default values for other parameters.



Configure the IDP parameters on the second tab.

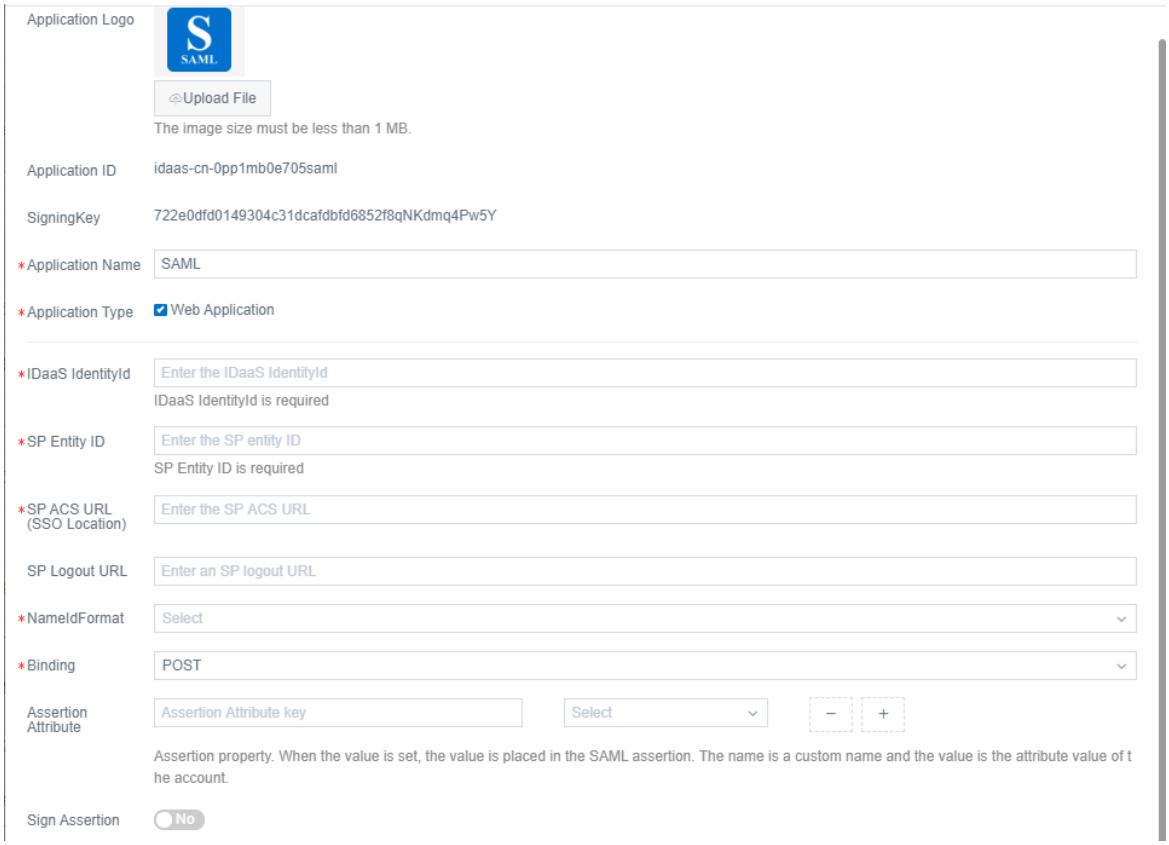


Parameter description

- IDP Name: Specify a name as needed.
- IDP Entity ID/Issuer: Enter the portal URL of the IDaaS user account.
- Send Signed Requests: Select this field.
- SSO Binding Type: Select the first option.
- Single Sign On URL: Enter the portal URL of the IDaaS user account.

- NameID Format: Select SAML:2.0 nameid-format persistent. The value must be consistent with that on IDaaS.
- IDP Signing Certificate: Enter the SigningKey information obtained in the preceding operation.

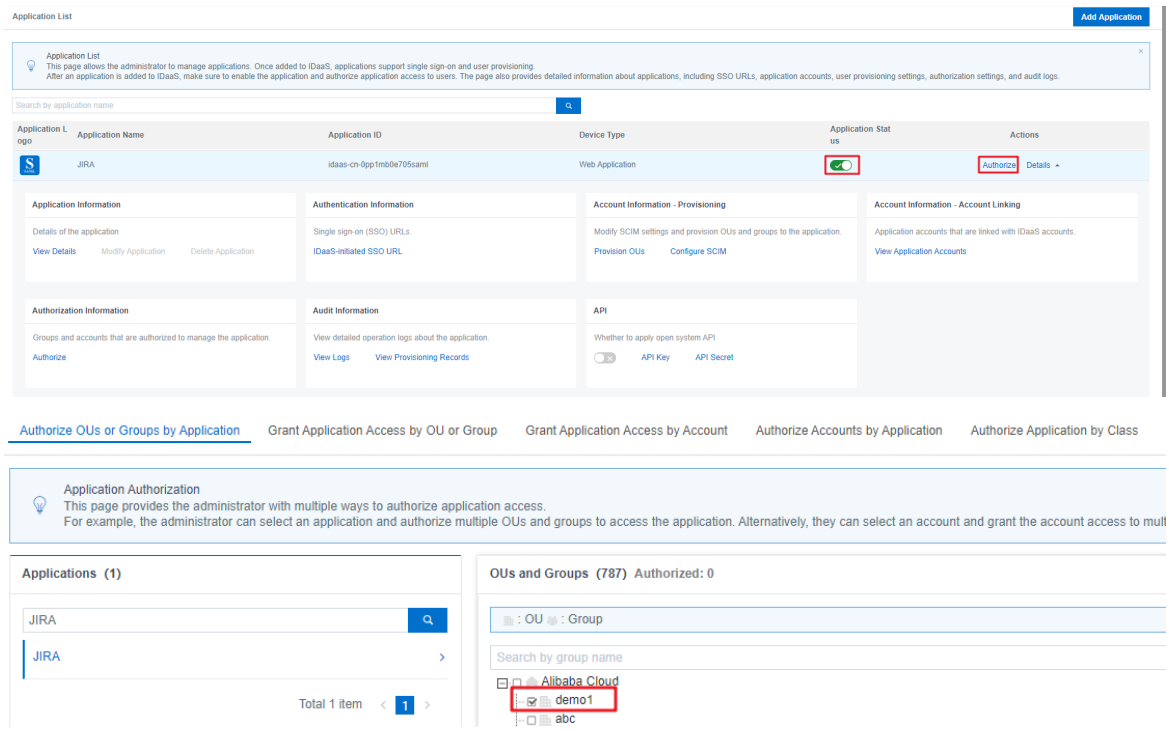
6. Configure SAML settings on IDaaS.



Parameter description

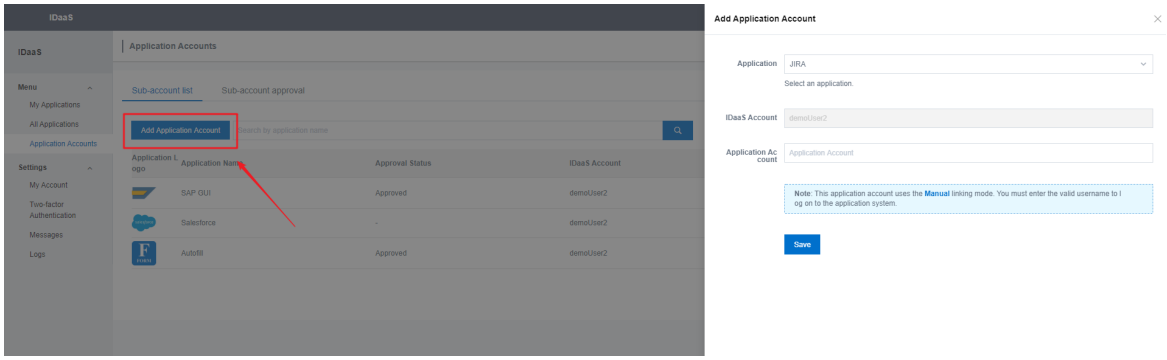
- SP Entity ID: Enter SP Base URL for your Confluence information. The value must be consistent with that in Confluence.
- IDaaS IdentityId: the portal URL of the IDaaS user account. The value must be consistent with that of IDP Entity ID/Issuer specified in the preceding operation.
- NameIdFormat: Select SAML:2.0 nameid-format persistent. The value must be consistent with that in Confluence.
- SP ACS URL (SSO Location): Obtain the SP ACS URL information from SP Base URL.

7. Enable and authorize the application on IDaaS.

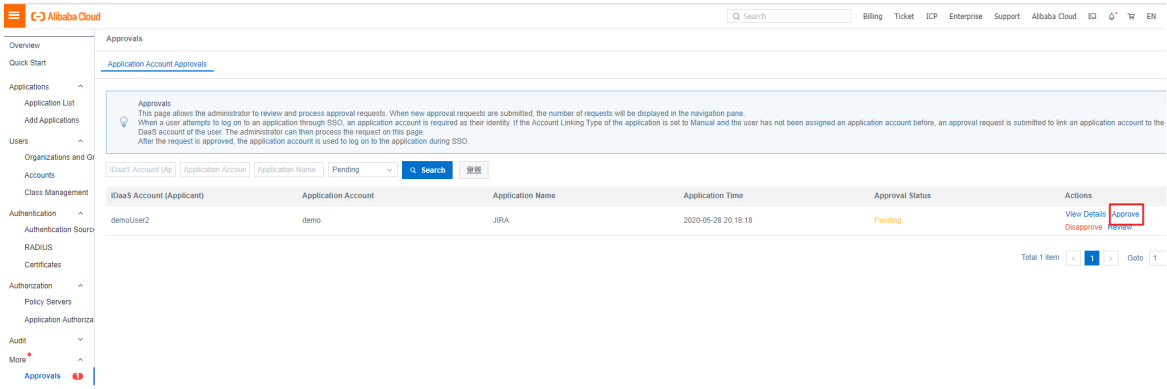


8. Log on the IDaaS console as the authorized user. Add an application account for the application.

   The application account is the account used in JIRA or Confluence.

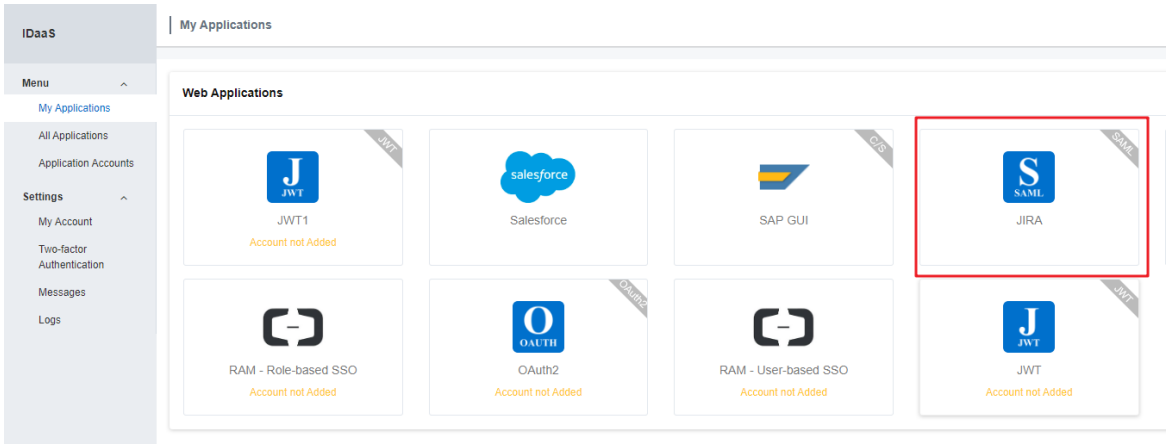


9. The IT administrator reviews and approves the new application account.



10. Log on to JIRA or Confluence from the IDaaS console in a single sign-on manner.

    You click the application icon on the My Applications page and log on to JIRA or Confluence in a single sign-on manner.



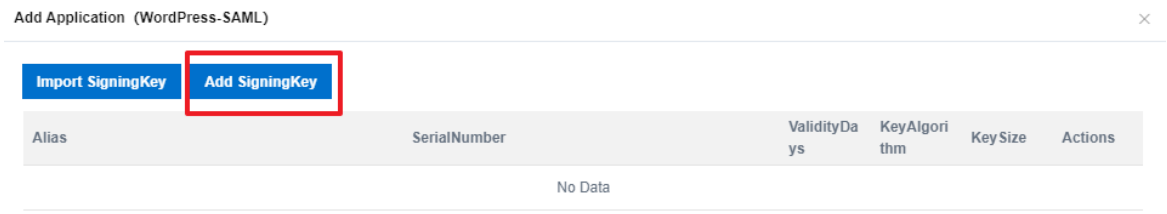
1.2. WordPress-SAML application usage

WordPress-SAML applications

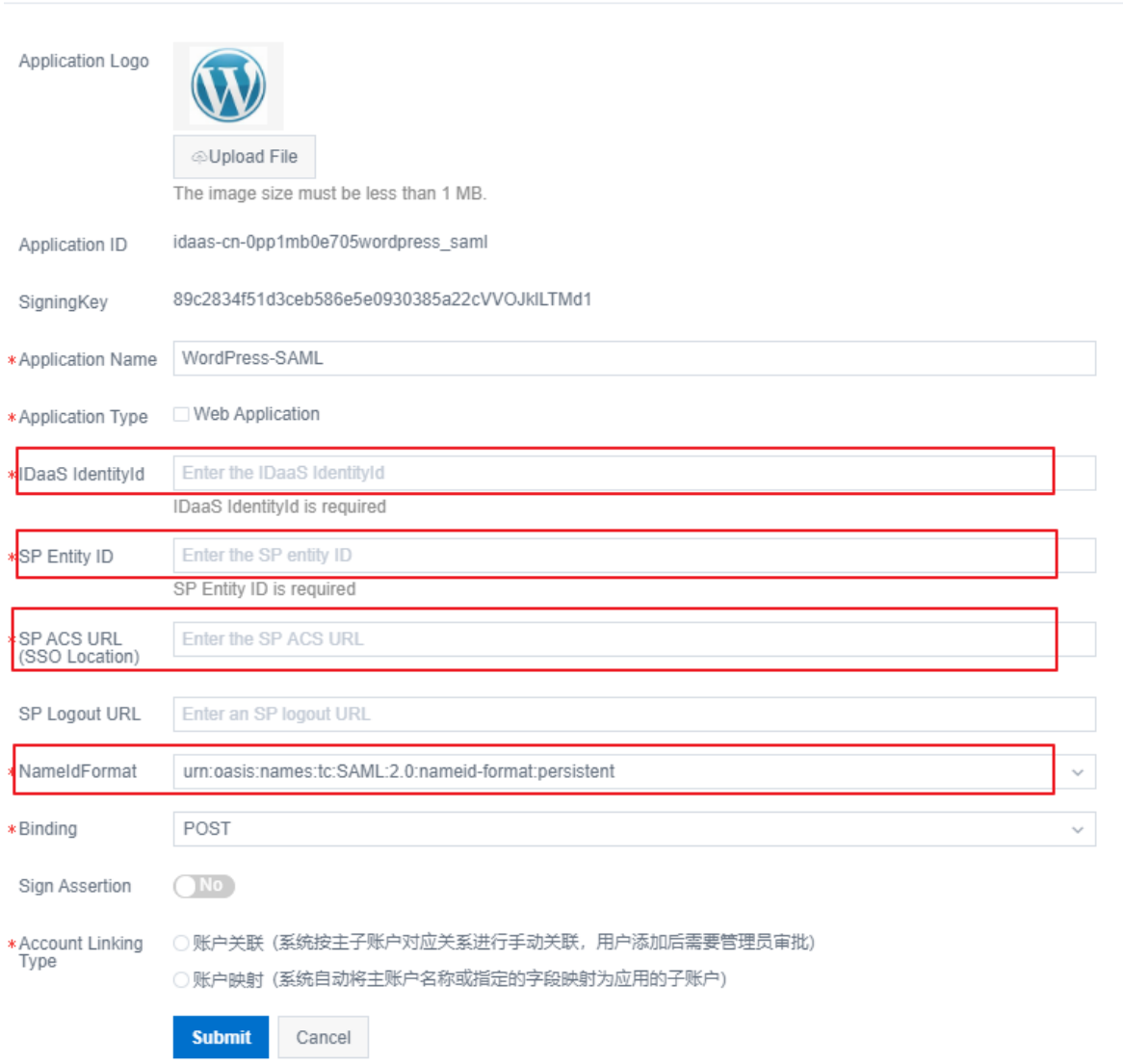
WordPress-SAML applications are used to implement single sign-on for WordPress.

Procedure

1. Log on to the IDaaS console as an IT administrator. For more information, see Logon in Administrator Guide.
2. In the left-side navigation pane, choose **Applications > Add Applications**. Find the SAML application and click **Add Application** in the Actions column.
3. Click **Add SigningKey**. Configure the parameters and click Submit.



4. Find the new SigningKey in the SigningKey list and click **Select** in the Actions column. Configure the parameters and click Submit.

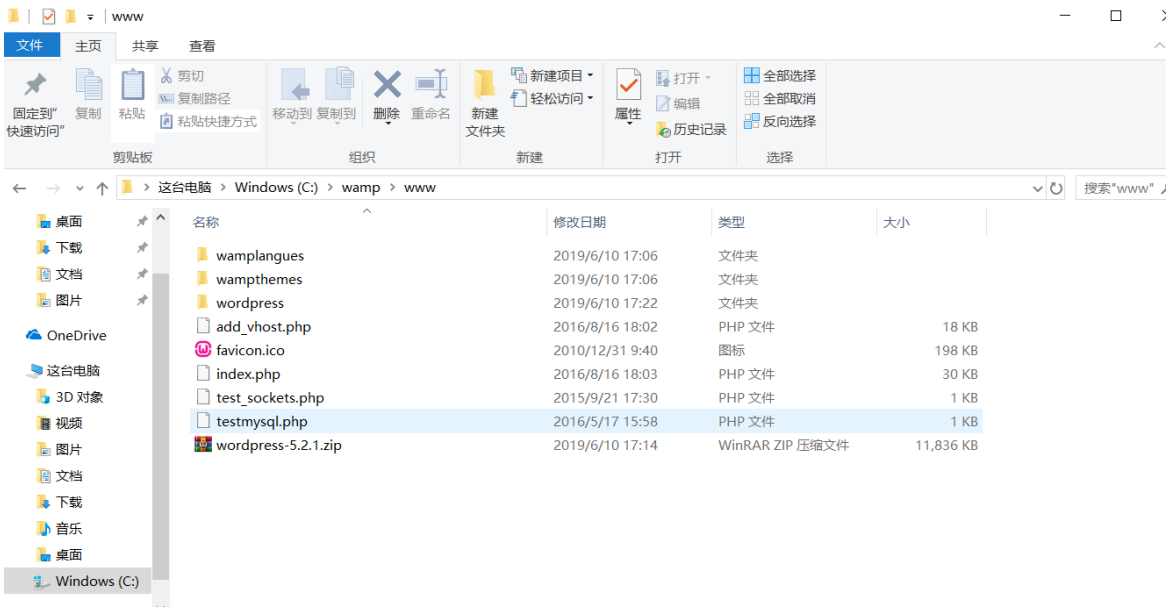


Note Set IDaaS IdentityId to a value as needed and this value must be consistent with that in WordPress. Set SP Entity ID, SP ACS URL, and NameIdFormat to the values obtained from WordPress.

Configure WordPress-SAML

WordPress-SAML runs in a PHP environment and you must set up a PHP environment. The procedure is as follows:

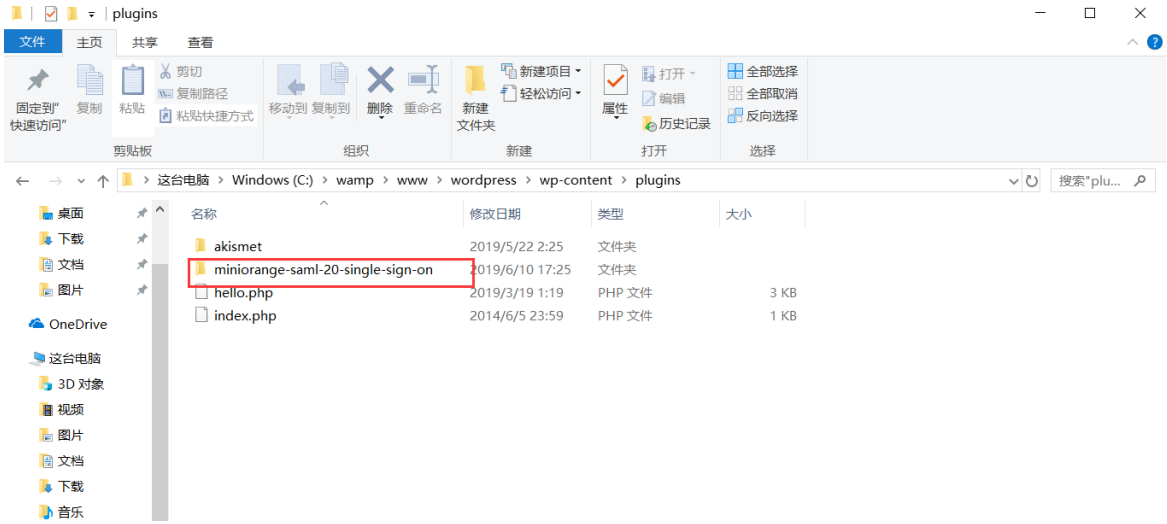
1. Download WampServer from the official website and decompress it.
2. Download WordPress from the official website and decompress it to the www directory of WampServe.



3. Create a WordPress account in MySQL.
4. On the WordPress welcome page, set the username and password.

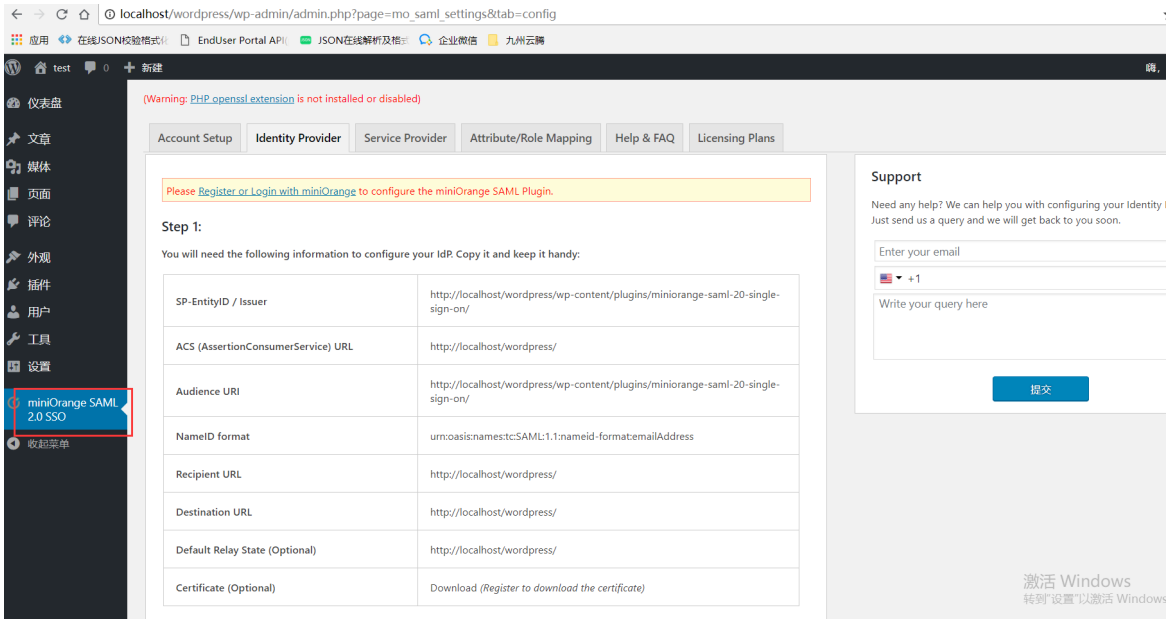
5. Click Install WordPress in the lower-left corner.
6. Download miniorange-saml-20-single-sign-on.4.8.23 and decompress it to the

C:\wamp\www\WordPress\wp-content\plugins directory.

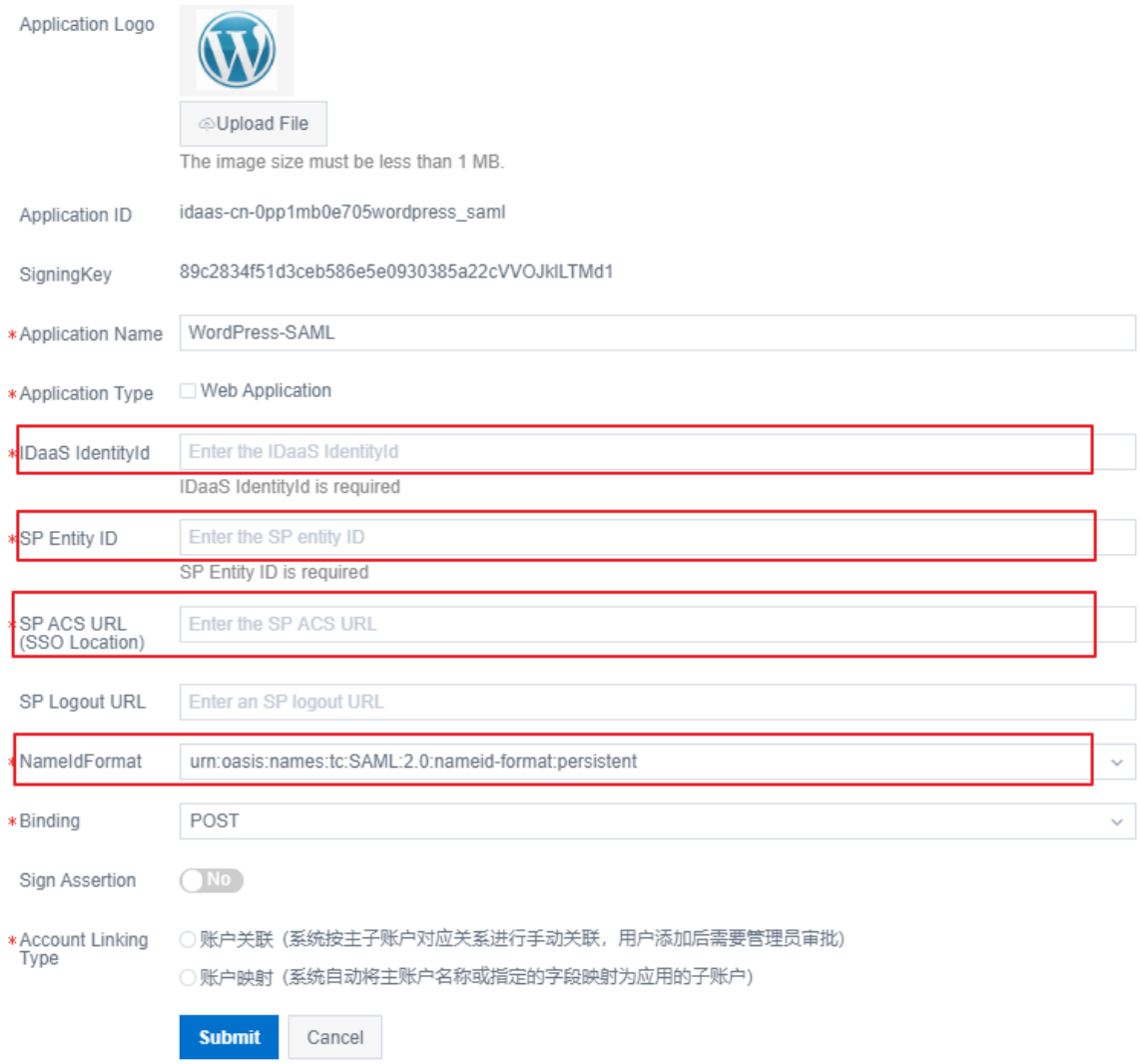


7. Restart the php environment and choose Restart All Services. Then install and enable miniOrange: Log on to WordPress and click Plug-ins. Click Enable for miniOrange and then refresh the page.

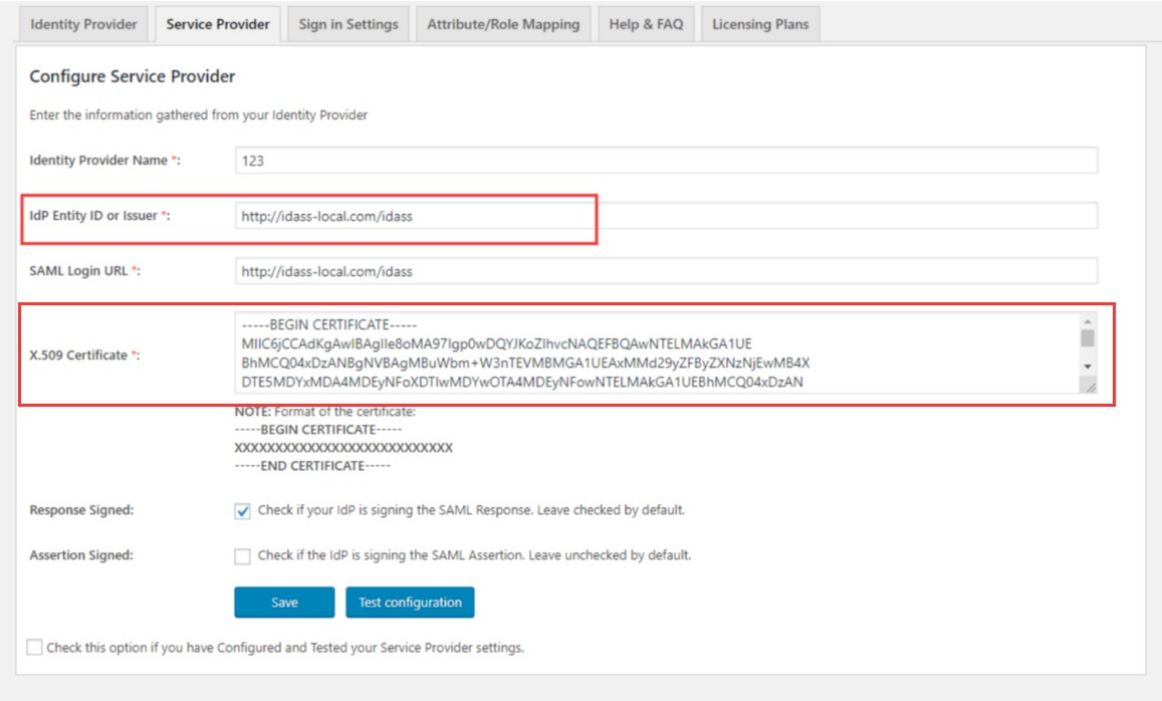
8. The miniOrange SAML item is displayed on the page.



9. Record the values of SP-EntityID/Issuer, ACS (Assertion Consumer Service) URL, and NameID format on the Identity Provider tab. Enter them on the Add Application (WordPress) page.



10. Set IDaaS IdentityId to a value as needed and this value must be consistent with that on the Service Provider tab in WordPress. Download the .cer file from the IDaaS console and copy the file information to the X.509 Certificate field on the Service Provider tab in WordPress.



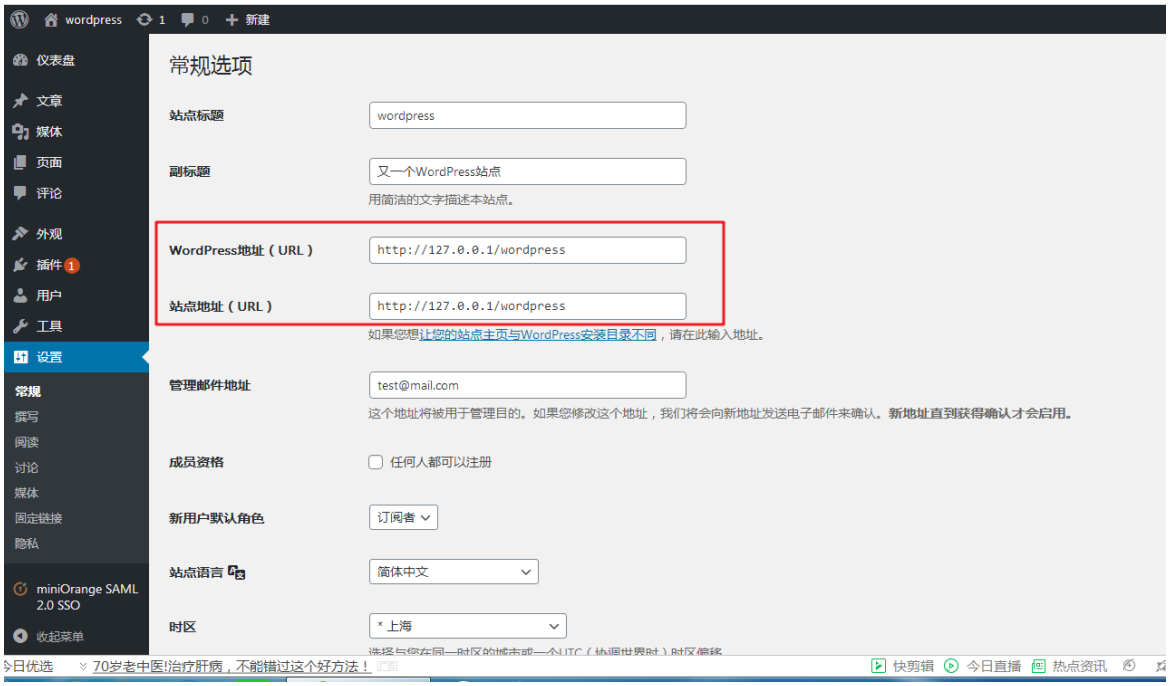
11. After the configuration is complete, add an application account for the WordPress-SAML application. This application account is the email address for the account in WordPress. Then you can log on to WordPress from the IDaaS console in a single sign-on manner.

Notes

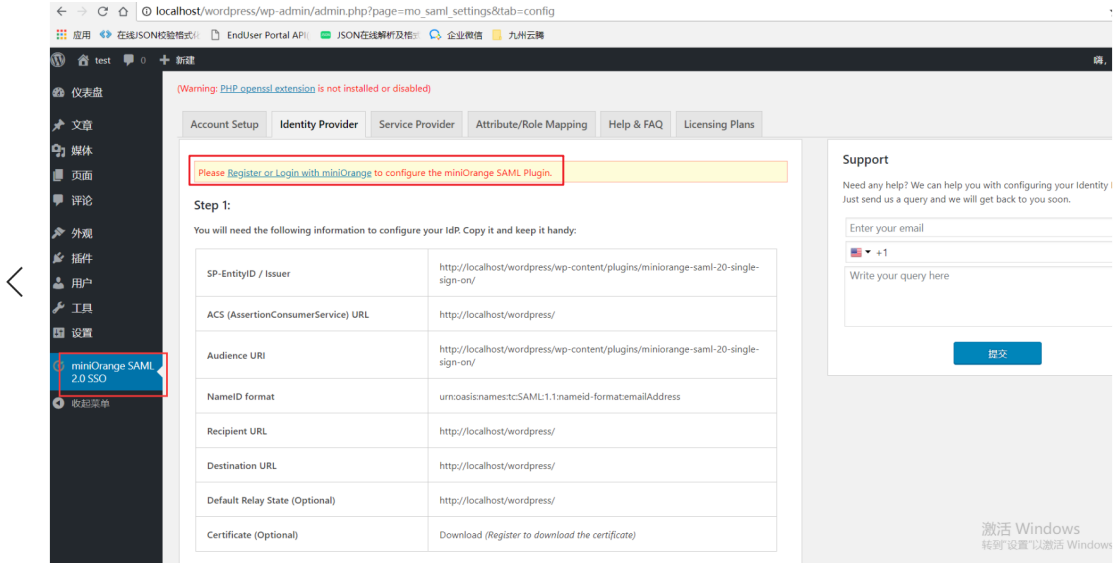
1. The URL of the local WordPress application is 127.0.0.1 or localhost. You can modify the URL on **Settings** page.

You will need the following information to configure your IdP. Copy it and keep it handy:

SP-EntityID / Issuer	http://127.0.0.1/wordpress/wp-content/plugins/miniorange-saml-20-single-sign-on/
ACS (AssertionConsumerService) URL	http://127.0.0.1/wordpress/
Audience URI	http://127.0.0.1/wordpress/wp-content/plugins/miniorange-saml-20-single-sign-on/
NameID format	urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
Recipient URL	http://127.0.0.1/wordpress/
Destination URL	http://127.0.0.1/wordpress/
Default Relay State (Optional)	Available in the **premium** version
Certificate (Optional)	Available in the **premium** version



2. If you cannot modify the miniOrange settings, click the link in the red rectangle in the following figure to log on to miniOrange.



1.3. Implement single sign-on for Salesforce

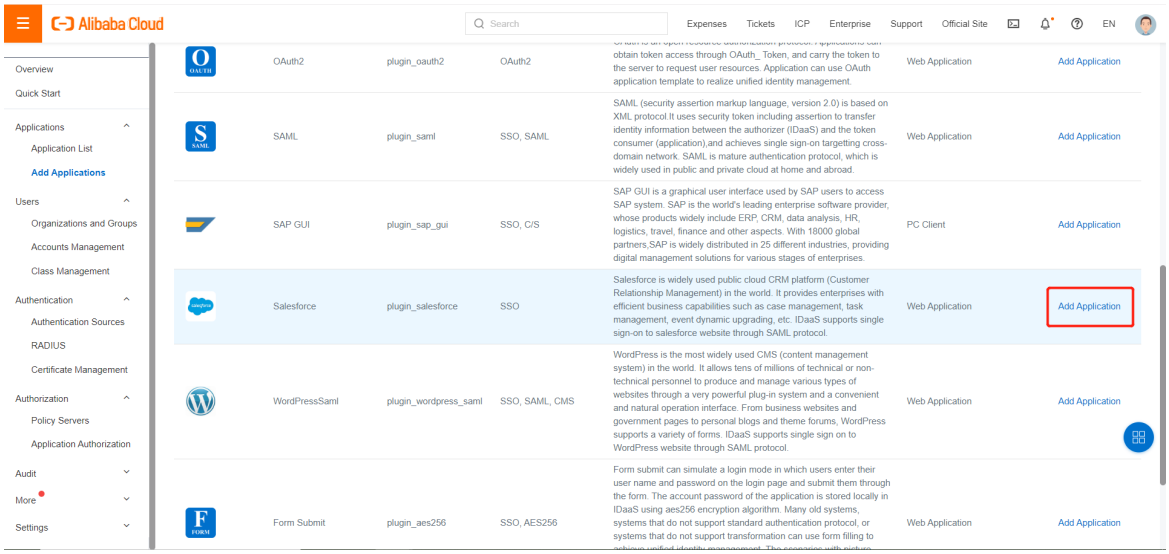
This topic describes how to use the SAML protocol to implement single sign-on for Salesforce in the IDaaS console.

Background

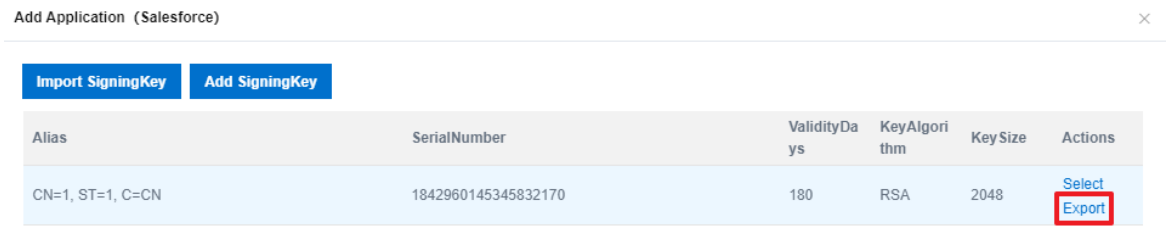
Salesforce is a customer relationship management (CRM) software service provider based in San Francisco, USA. It was founded in 1999 and provides a customer relationship management platform for on-demand applications. Salesforce supports single sign-on with the SAML protocol.

Procedure

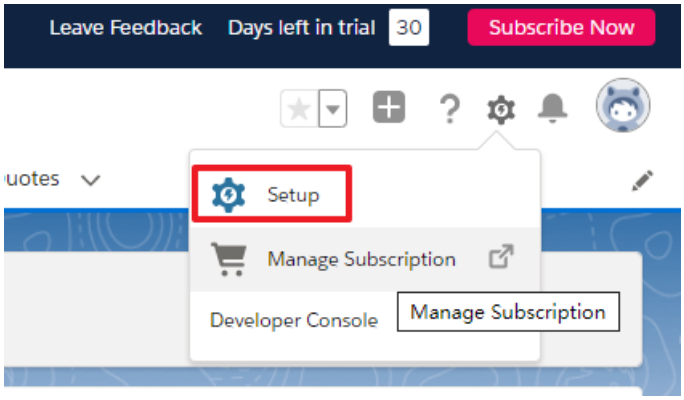
1. Log on to the IDaaS console as an IT administrator. For more information, see Logon in Administrator Guide.
2. In the left-side navigation pane, choose **Applications > Add Application**. Find the Salesforce application and click Add Application in the Actions column.



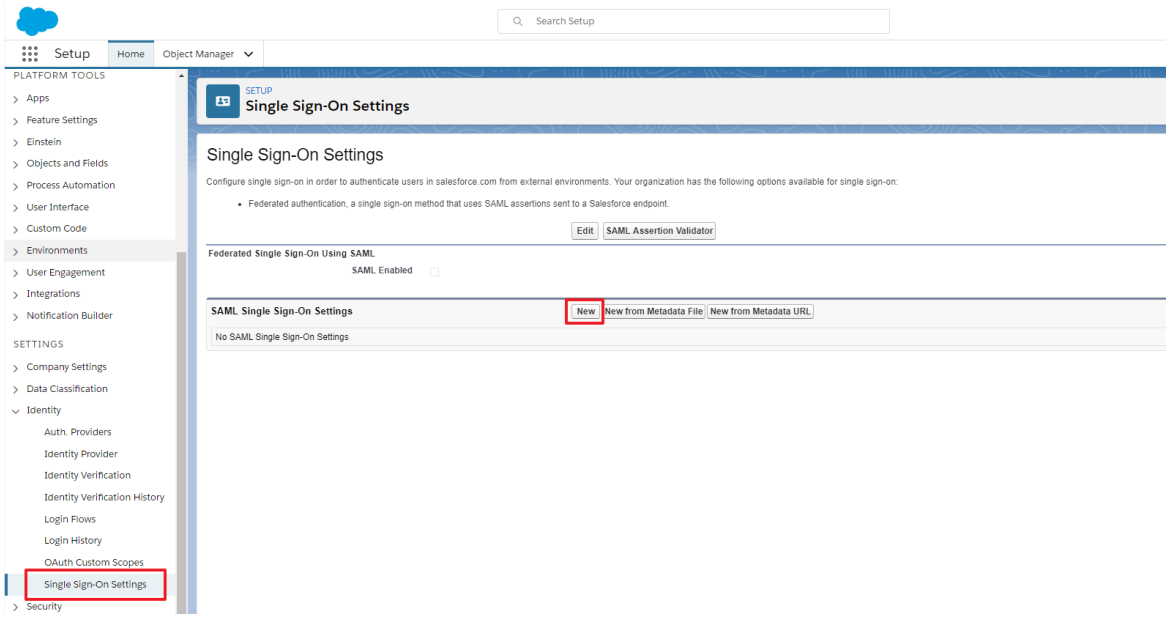
3. Find an existing SigningKey. You can add a SigningKey first if there are no existing SigningKeys. Click Export in the Actions column. Export a .cer certificate locally.



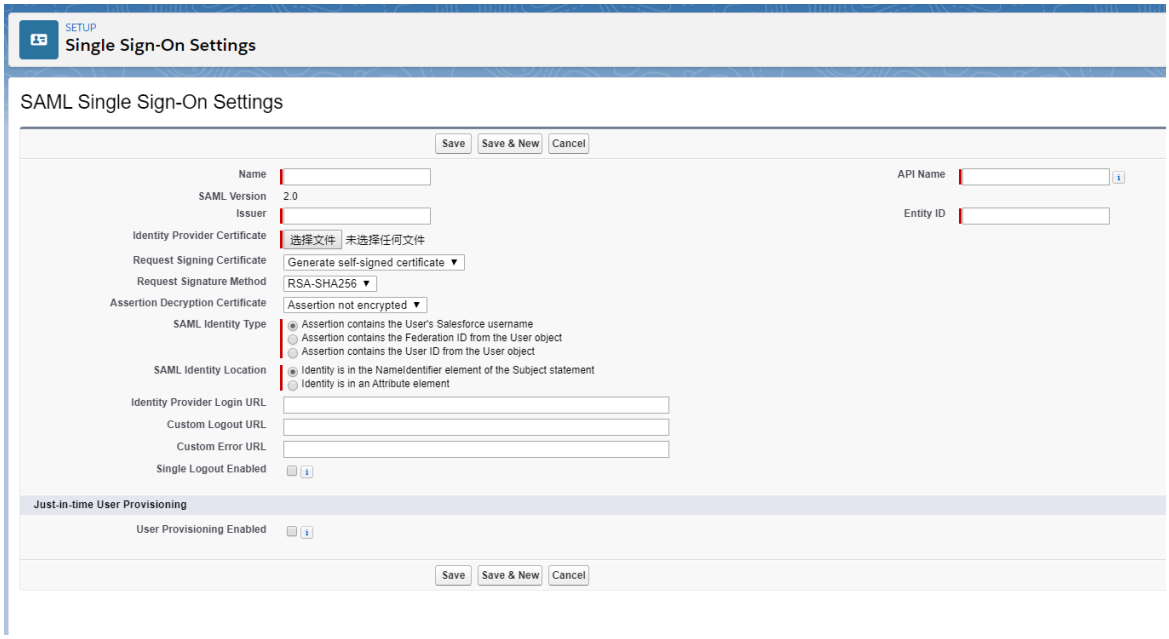
4. Log on to Salesforce as an administrator. Click Settings in the upper-right corner.



5. In the left-side navigation pane, choose Identity > Single Sign-On Settings. Find SAML Single Sign-On Settings and click New.



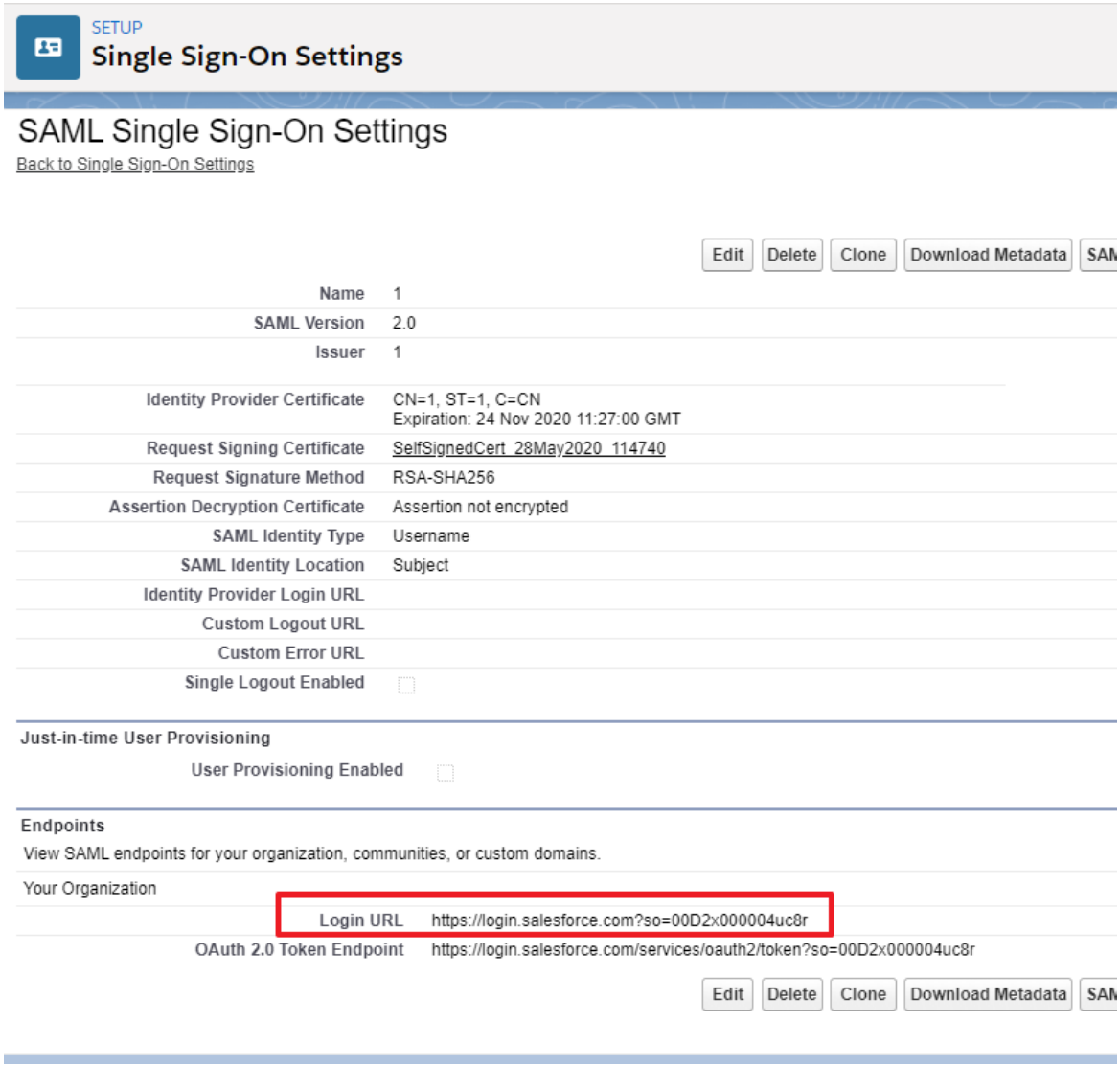
6. Go to the SAML Single Sign-On Settings page.



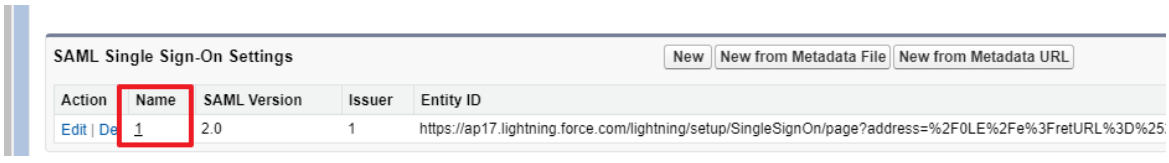
- **Name**: the name of the SAML single sign-on configuration. You can enter a name as needed.
- **Issuer**: Note that this value must be the same as that of **IDaaS IdentityId** in IDaaS.
- **Entity ID**: Set it to https://SAML.Salesforce.com.
- **Identity Provider Certificate**: Select the certificate file exported from IDaaS.
- **Request Signing Certificate**: Use the default value.
- **Request Signature Method**: Set it to RSA-SHA1.
- **Assertion Decryption Certificate**: Select Assertion unencrypted.
- **SAML Identity Type**: Select Assertion contains User's Salesforce username.
- **SAML Identity Location**: Select Identity is in the NameIdentifier element of the Subject statement.
- Leave **Identity Provider Login URL**, **Customer Logout URL**, and **Custom Error URL** empty.

Click Save.

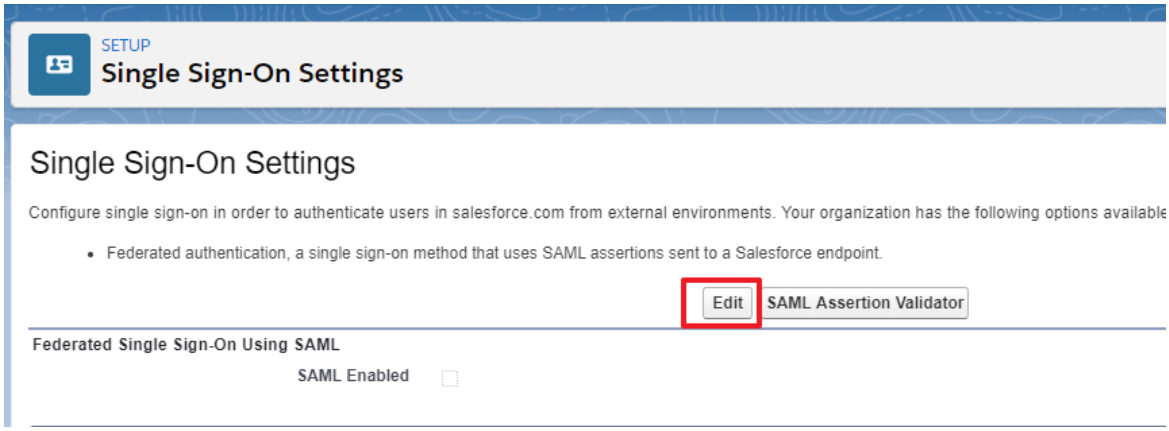
7. After you configure the SAML settings, the SAML details will be displayed with the name that you specified. You must the value of Salesforce Login URL for later use.



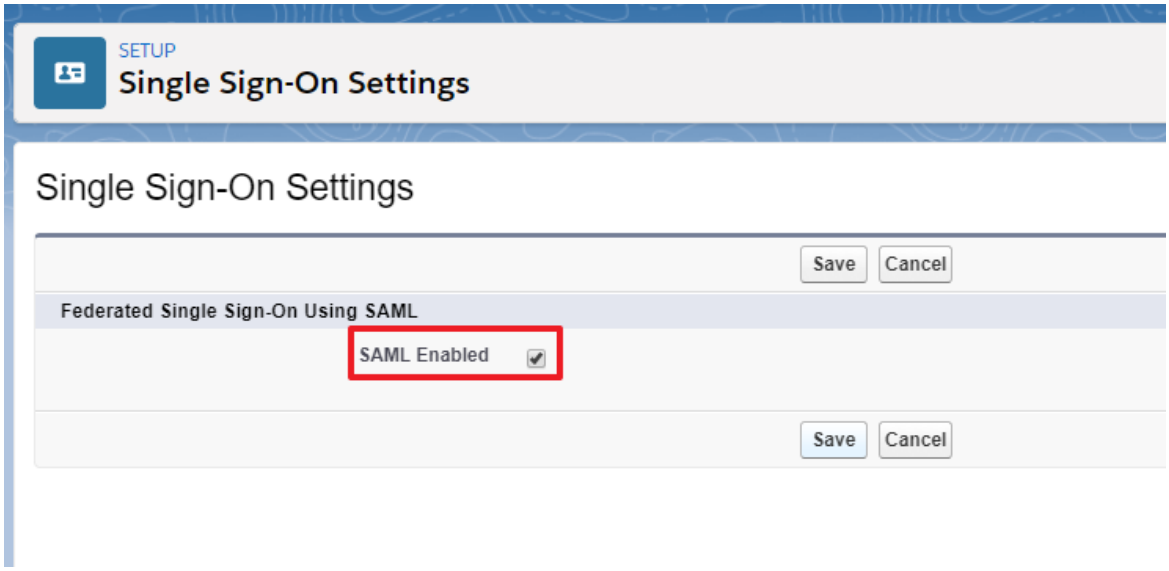
Note: You also can click the SAML name to view the value of Salesforce Login URL on the preceding page.



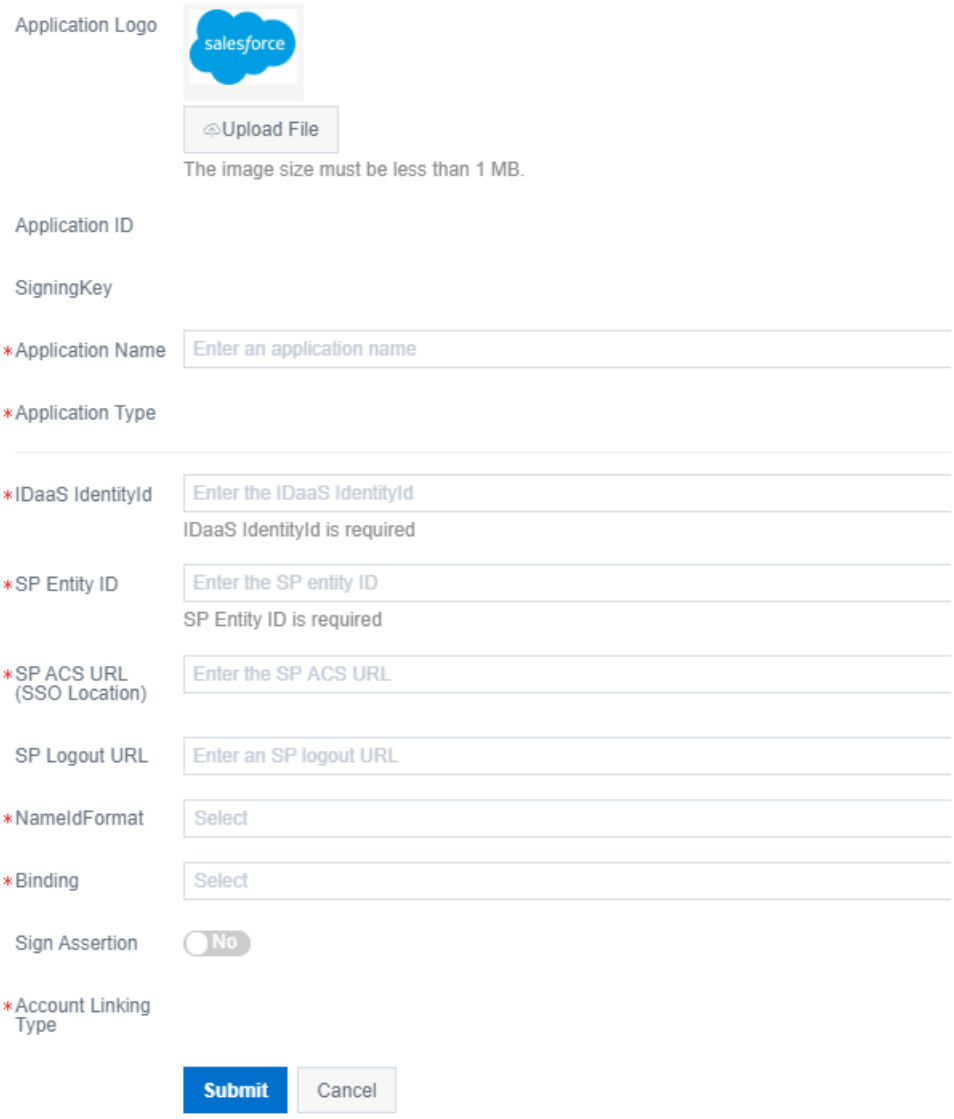
8. Find Single Sign-On Settings and click Edit.



9. Select the SAML Enabled check box and click Save.



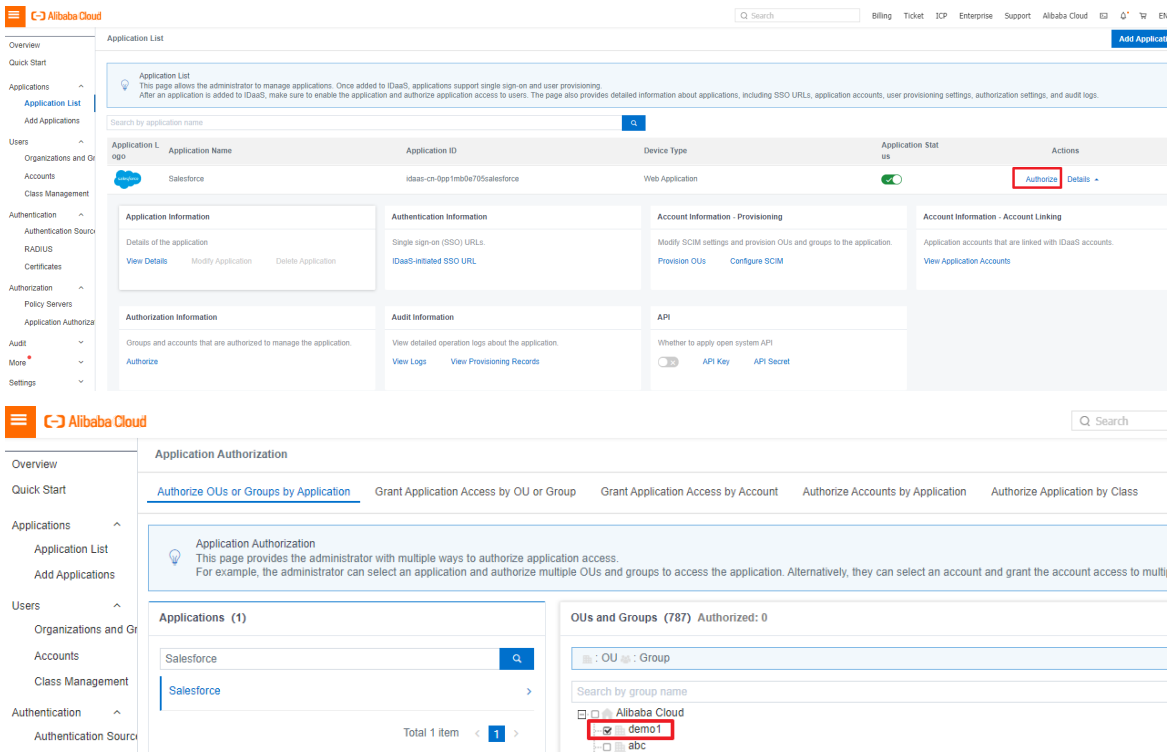
10. Return to the Add Application (Salesforce) page of the IDaaS console. Find the target SingingKey and click Select in the Actions column to configure the SAML parameters.



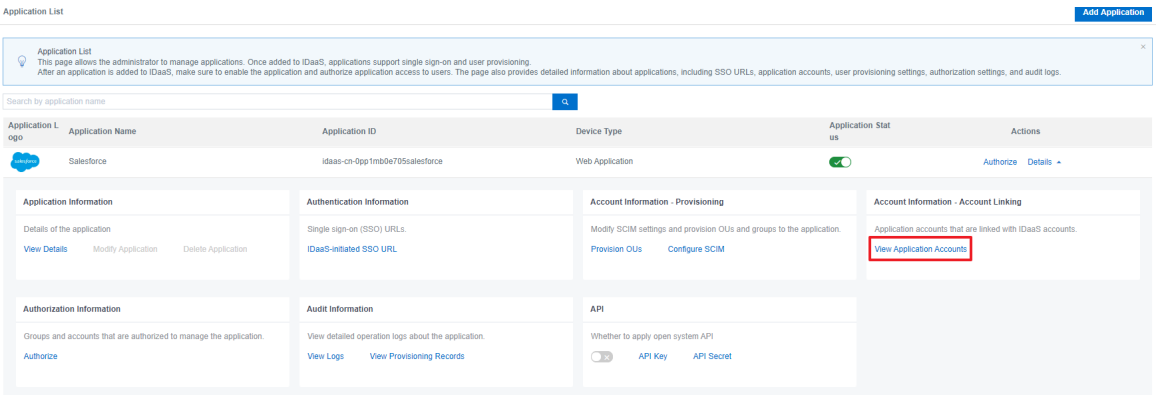
- Set IDaaS IdentityId to the value of Issue specified in Salesforce.
- Set SP ACS URL(SSO Location) to the Salesforce logon URL.

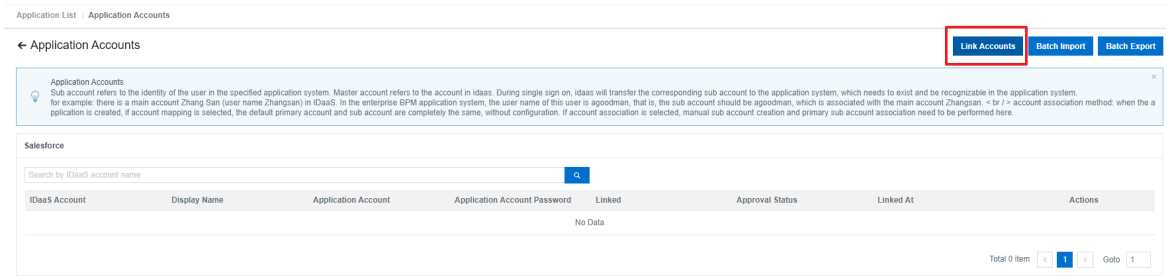
> **Note** The URL format is https://login.Salesforce.com?so=<Your organization ID>. If you are not sure about your organization ID in Salesforce, go to the Company Profile > Company Information page of Salesforce.

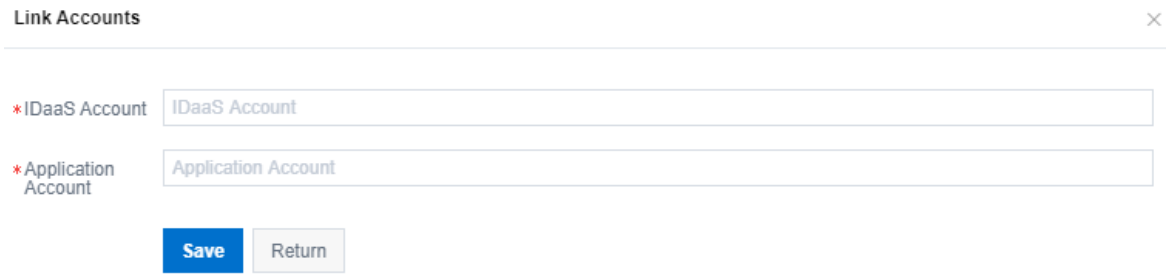
11. Enable and authorize the application.

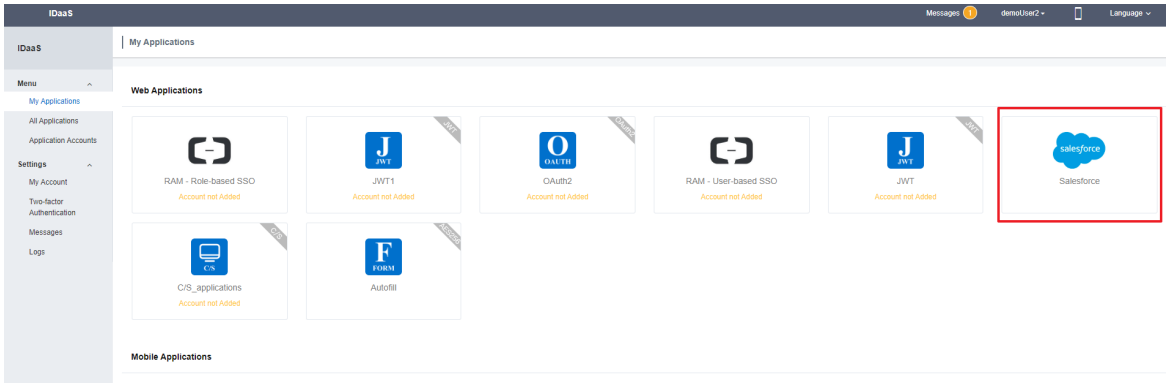


12. Add an application account and log on to Salesforce in a single sign-on manner.









If all the preceding steps are successful, you have logged on to Salesforce in a single sign-on manner.

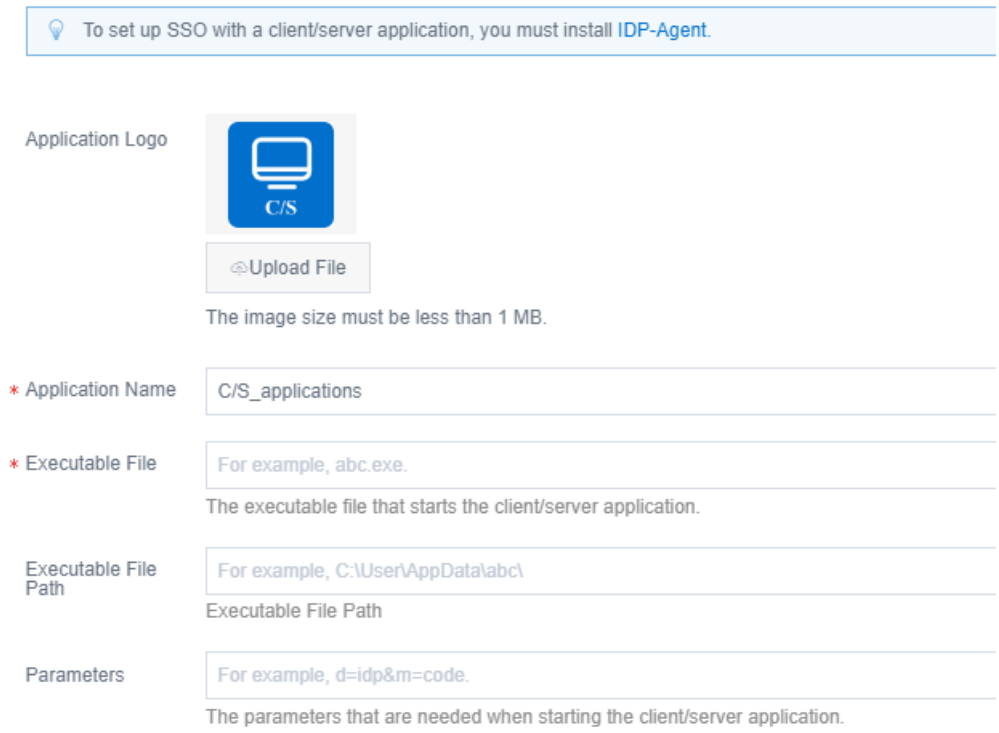
2.Standard Protocol Template Usage Guide

2.1. C/S Applications User Manua

The OIDC protocol can be used to pass in parameters for logon after the program has been enabled. This method only applies to applications that can receive and parse OIDC protocol parameters.

Procedure

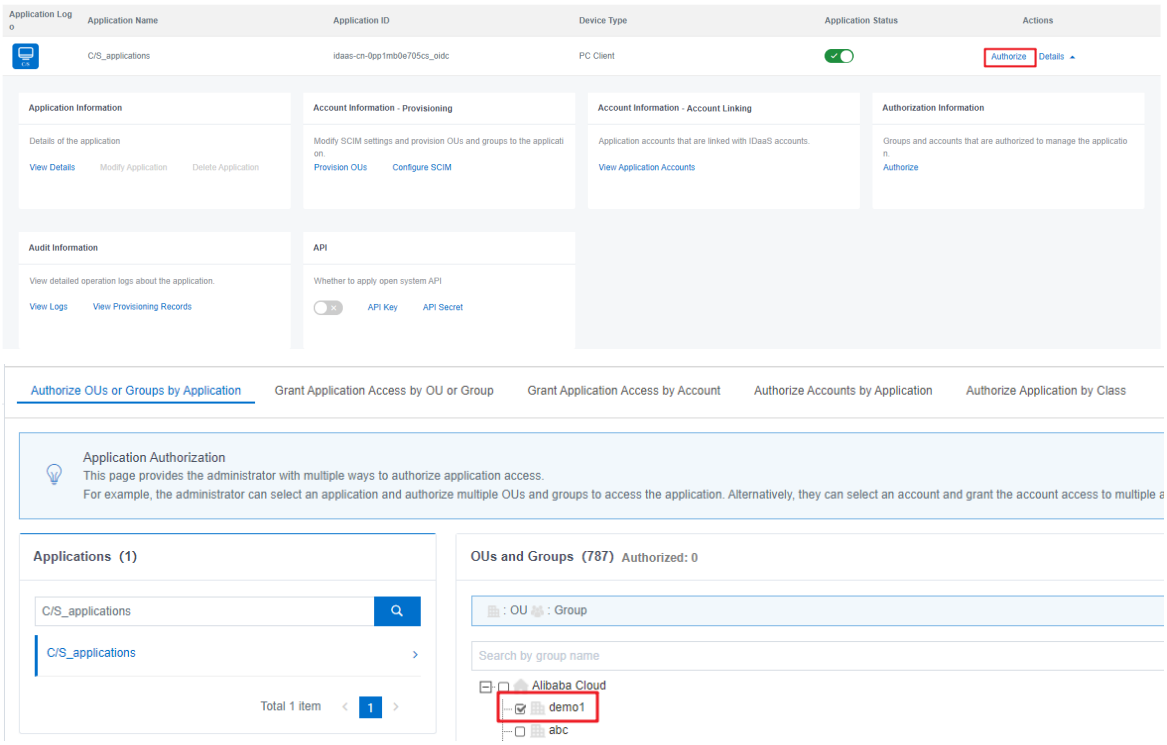
1. In the left-side navigation pane, choose **Applications > Add Applications**. Find the C/S program and click **Add Application** in the Actions column.



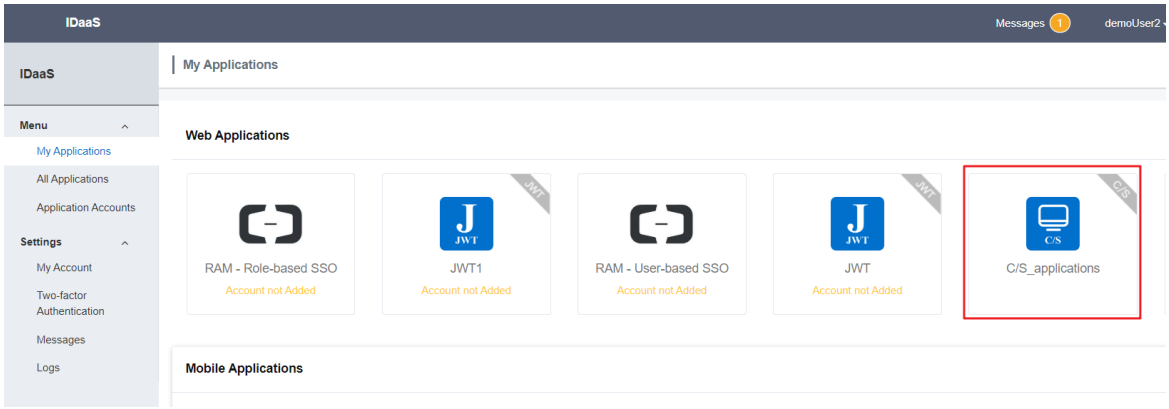
> **Note** To implement single sign-on for C/S applications, you must install the IDP-Agent plug-in locally.

- Application Name: required. Specify a name as needed.
- Executable File: the name of the executable file for startup.
- Executable File Path: the path of the C/S application file on the local computer.

2. Enable and **Authorize** the application. By default, Authorize Groups by Application is selected.



3. Log on to the IDaaS console as a common user. For more information, see Logon in User Guide.
4. Click the C/S application icon on the My Application page.



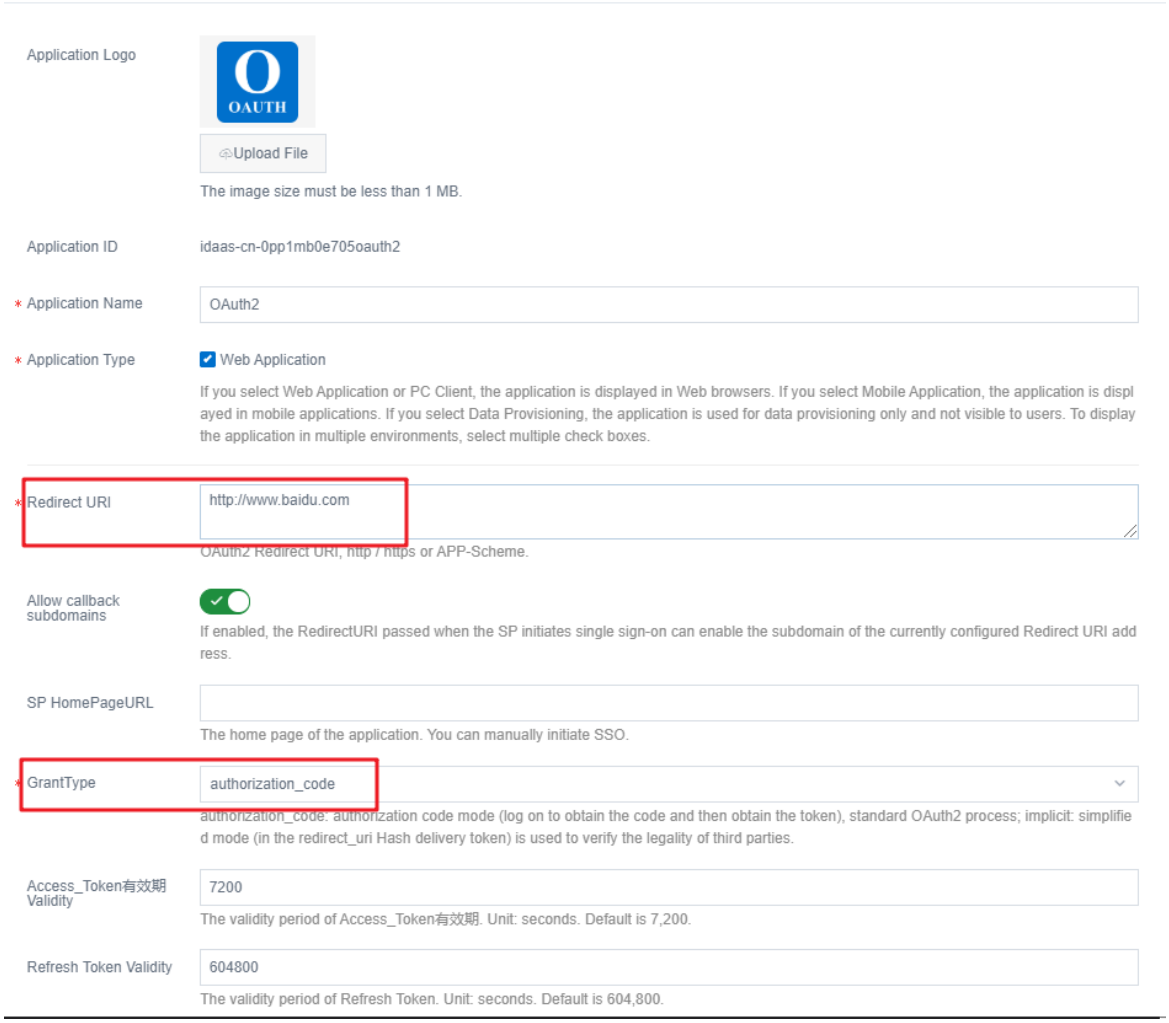
After the application account has been added, you can click the C/S application icon on the My Application page to perform single sign-on.

2.2. OAuth2.0 Application User Manual

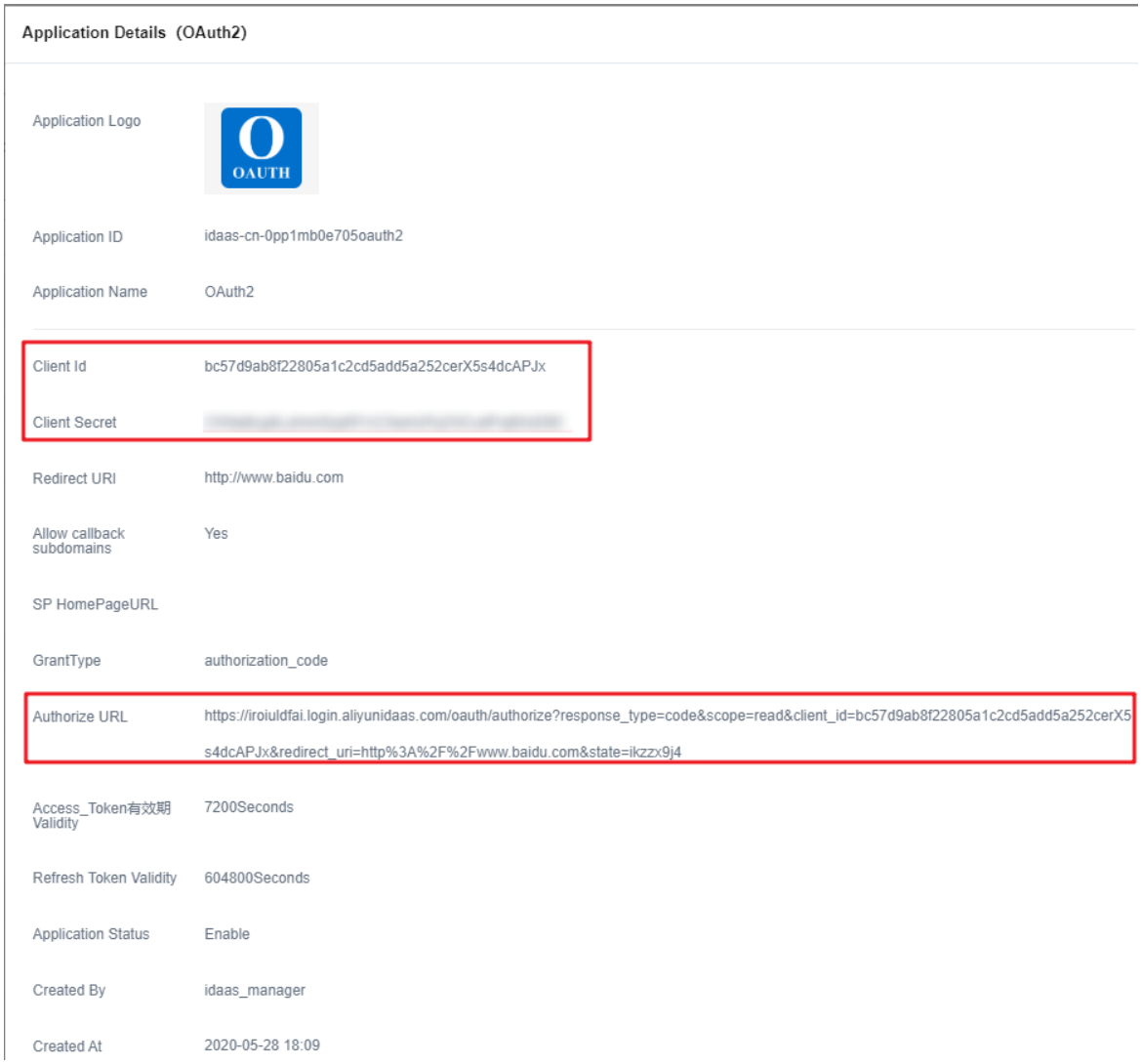
OAuth2 is an open protocol for resource authorization. Applications can obtain access tokens through OAuth and use the tokens to request user resources from the server. Applications can use the OAuth application for centralized authentication.

Procedure

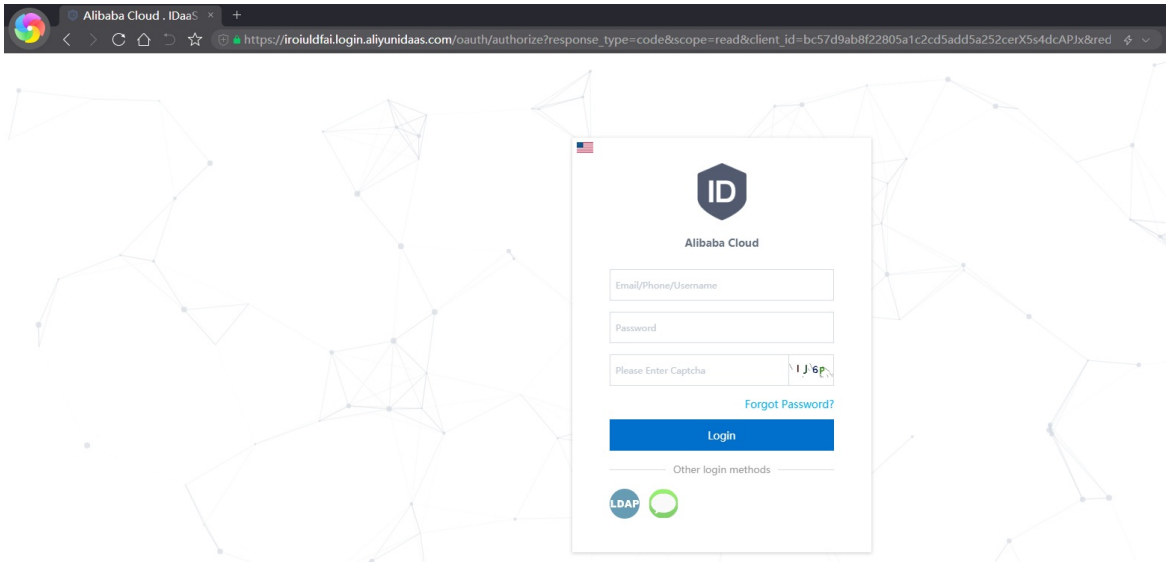
1. Log on to the IDaaS console as an IT administrator. For more information, see Logon in Administrator Guide.
2. In the left-side navigation pane, choose **Applications > Add Application**.
3. Find the OAuth2 application and click **Add Application** in the Actions column.

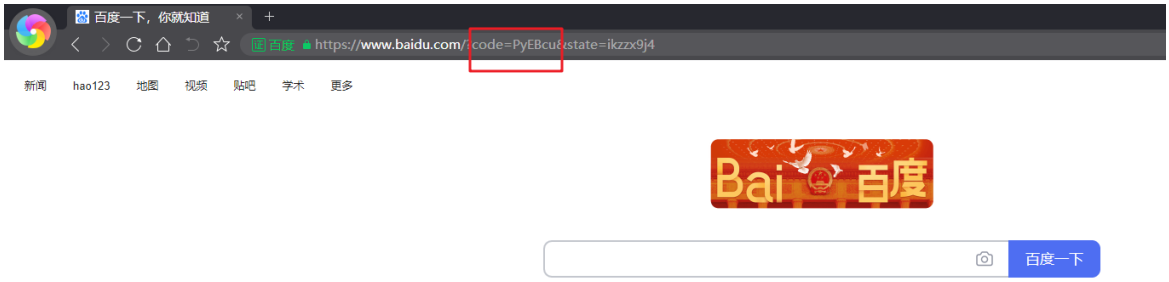


4. View the details of the OAuth2 application and obtain the values of AppKey, AppSecret, and Authorize URL.



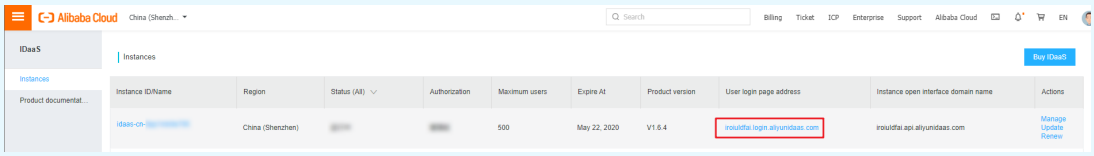
5. Open the Authorize URL in your browser and use the authorized account to log on. After successful logon, you will be redirected to the webhook address. Extract the value of the code parameter from the address bar of the browser.





6. Use Postman to send a POST request to http://{IDaaS_server}/oauth/token?grant_type=authorization_code&code={code}&client_id={AppKey}&client_secret={AppSecret}&redirect_uri={redirect_uri}
   - Replace {IDaaS_server} with the IP address of the IDaaS server.

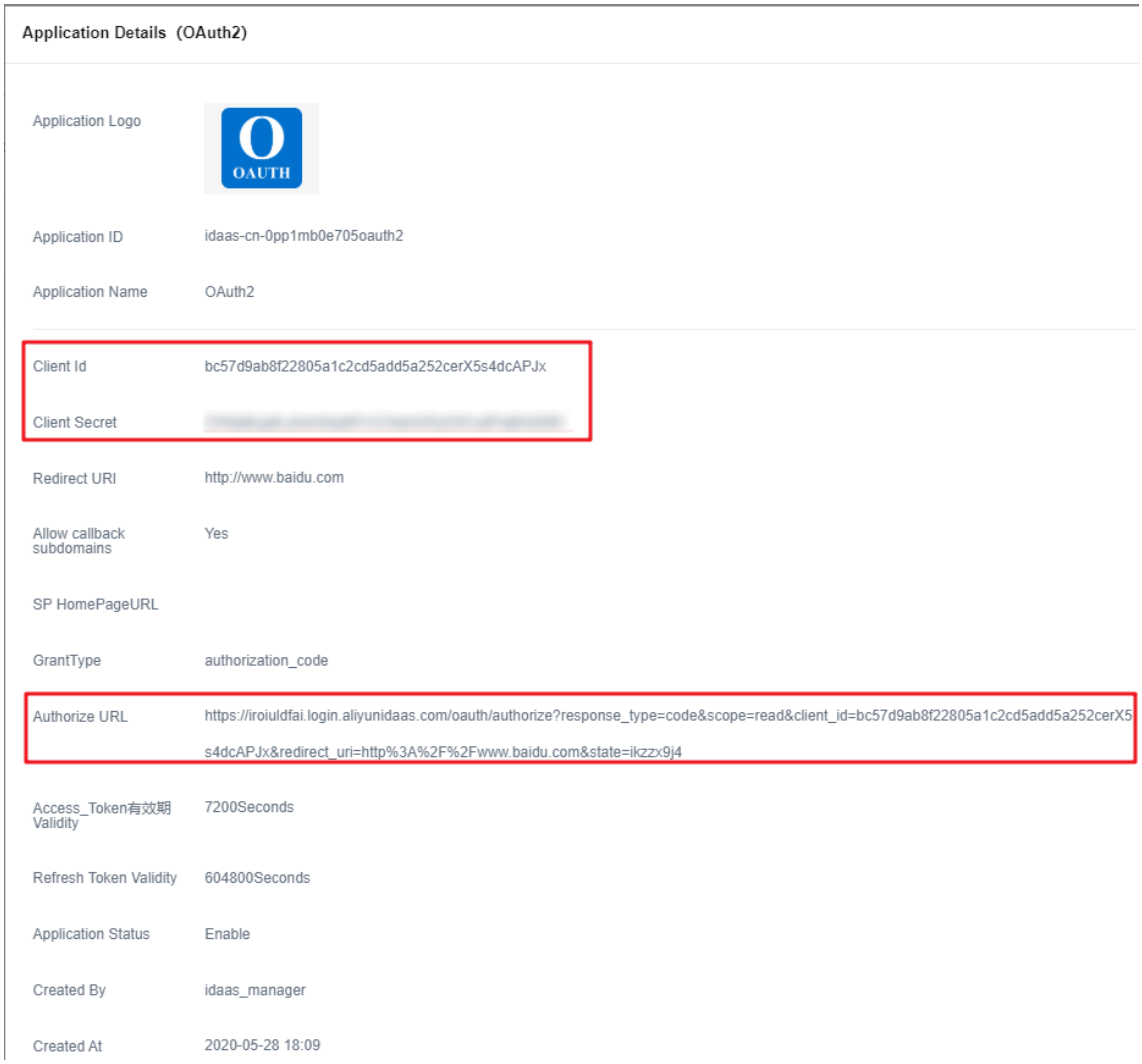
     Note To obtain IP address of the IDaaS server, log on to the IDaaS console and obtain the value in the **Portal API Address for User Access**.

   - Replace {code} with the value of the code parameter obtain in step 5.

     Notice The value of the code parameter can only be used once.

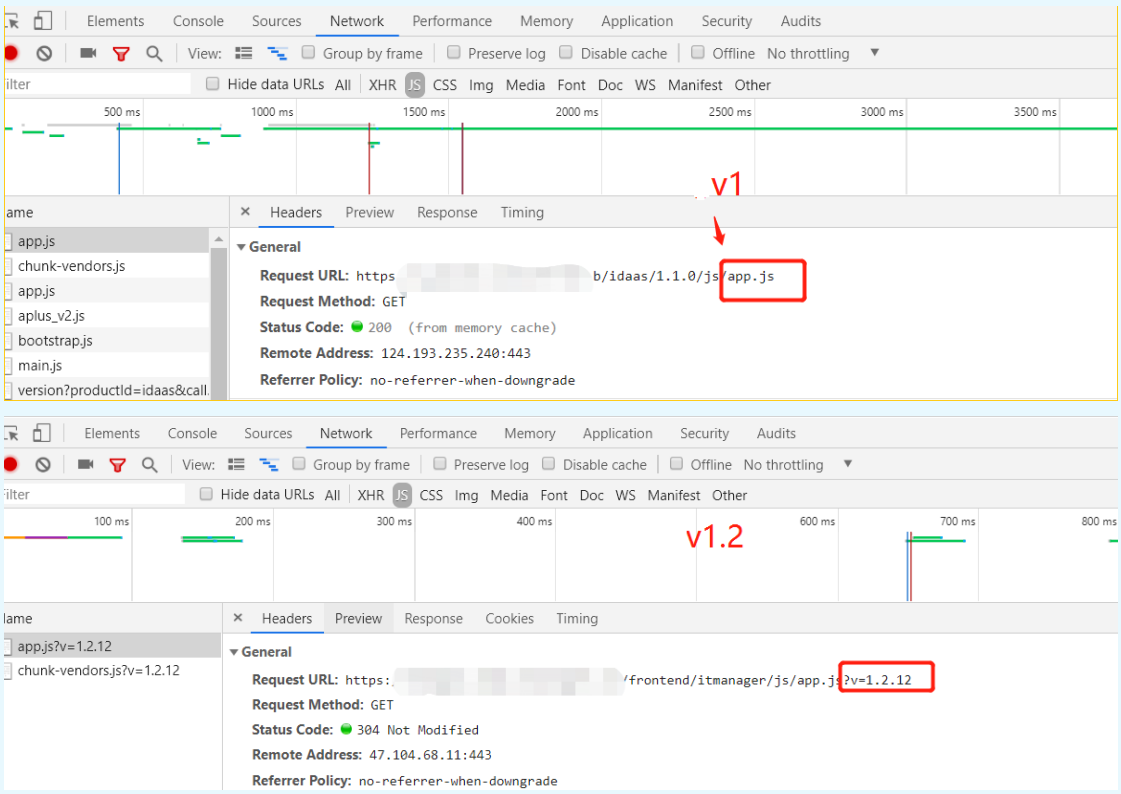
   - Replace {AppKey} and {AppSecret} with the values obtained in step 4.



- Replace {redirect_uri} with the value of Redirect URL that you specified in step 3.

7. The IDaaS server returns the access token, which can be used to access IDaaS server resources.
8. Use Postman to send a GET request to http://{IDaaS_server} /api/bff/v1.2/oauth2/userinfo?access_token={access_token}

Note v1.2 indicates the version. Replace it with the actual version. If no version number is written in js, enter v1 here. If a version number is written in js, enter that value here.



API operations

1. Request URI: /oauth/token
   - Description: You can call this operation to obtain the access token.
   - Request parameters

Parameter	Type	Required	Example	Description
code	String	Yes	vuQ3n6	The value of the code parameter in the callback after a successful logon.
client_id	String	Yes	oauth2 client_id	OAuth2 client_id
client_secret	String	Yes	oauth2 client_secret	OAuth2 client_secret
redirect_uri	String	Yes	http://example.com	The redirect URL

   - Response parameters

Parameter	Type	Example	Description
access_token	String	333ab704-abc0-48b3-8af0-496eedd15383	The access token returned.
token_type	String	bearer	The type of the access token.
expires_in	String	7199	The expiration time of the access token.
scope	String	read	The granted permissions.

- Error codes

HTTP status code	Error code	Error message	Description
400	invalid_grant	Invalid authorization code: "code".	The error message returned because the value of the code parameter is invalid.
400	invalid_grant	Redirect URI mismatch.	The error message returned because the value of Redirect URI is invalid.
401	Unauthorized	Unauthorized	The error message returned because your access is not authorized.
403	Forbidden	Forbidden	The error message returned because your access was denied.
404	ResourceNotFound	ResourceNotFound	The error message returned when the specified resource does not exist.
415	UnsupportedMediaType	UnsupportedMediaType	The error message returned because the media type is not supported.
500	InternalError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.

2. Request URI: /api/bff/v1.2/oauth2/userinfo

- Description: You can call this operation to obtain user details.
- Request parameters

Parameter	Type	Required	Example	Description
access_token	String	Yes	333ab704-abc0-48b3-8af0-496eedd15383	The access token.

- Response parameters

  Sample responses

```
{
"success": true,
"code": "200",
"message": null,
"requestId": "59C5766B-C7F9-4DF6-B5E4-0F2A89942749",
"data": {
    "sub": "4982789226325725762",
    "ou_id": "5920417439492153461",
    "nickname": "admin",
    "phone_number": null,
    "ou_name": "PG China",
    "email": "sz@xxxx.com",
    "username": "admin_wli"
  }
}
```

  Parameters for running the Spark Structured Streaming program

Parameter	Type	Example	Description
sub	String	4982789226325725762	The external ID of the account.
username	String	admin_wli	The username of the account.
nickname	String	admin	The nickname of the account.
email	String	sz@xxxx.com	The email address of the account.
phone_number	String	null	The phone number of the account.
ou_name	String	PG China	The name of the organization to which the account belongs.

Parameter	Type	Example	Description
ou_id	String	592041743949215346 1	The external ID of the organization to which the account belongs.

- Error codes

HTTP status code	Error code	Error message	Description
401	Unauthorized	Unauthorized	The error message returned because your access is not authorized.
403	Forbidden	Forbidden	The error message returned because your access was denied.
404	ResourceNotFound	ResourceNotFound	The error message returned when the specified resource does not exist.
415	UnsupportedMediaType	UnsupportedMediaType	The error message returned because the media type is not supported.
500	InternalError	The request processing has failed due to some unknown error, exception or failure.	The error message returned because an internal error has occurred.

2.3. Form Autofill Template User Manual

This topic describes how to use Form Autofill to implement single sign on for an application.

Background

A company uses Application A as its website with high visits. The traditional access method is simple but poses security risks.

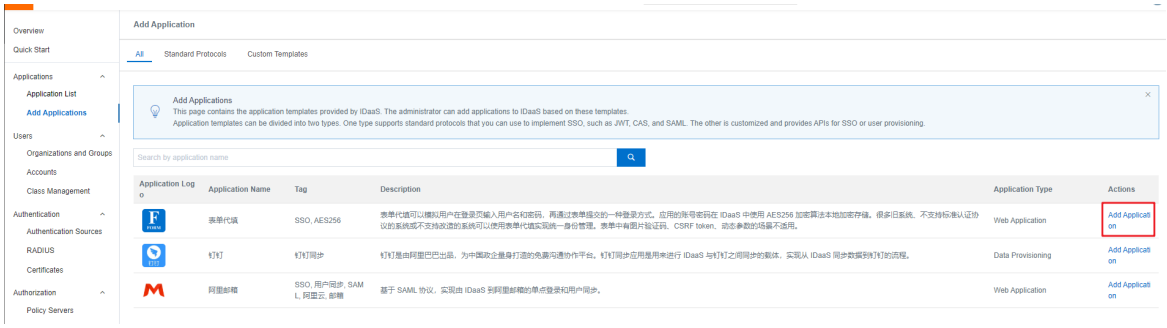
- Application A is frequently used in daily office work and repeated logons are time-consuming.
- There are security risks in Application A because verification codes are not used for logon authentication.

Solution

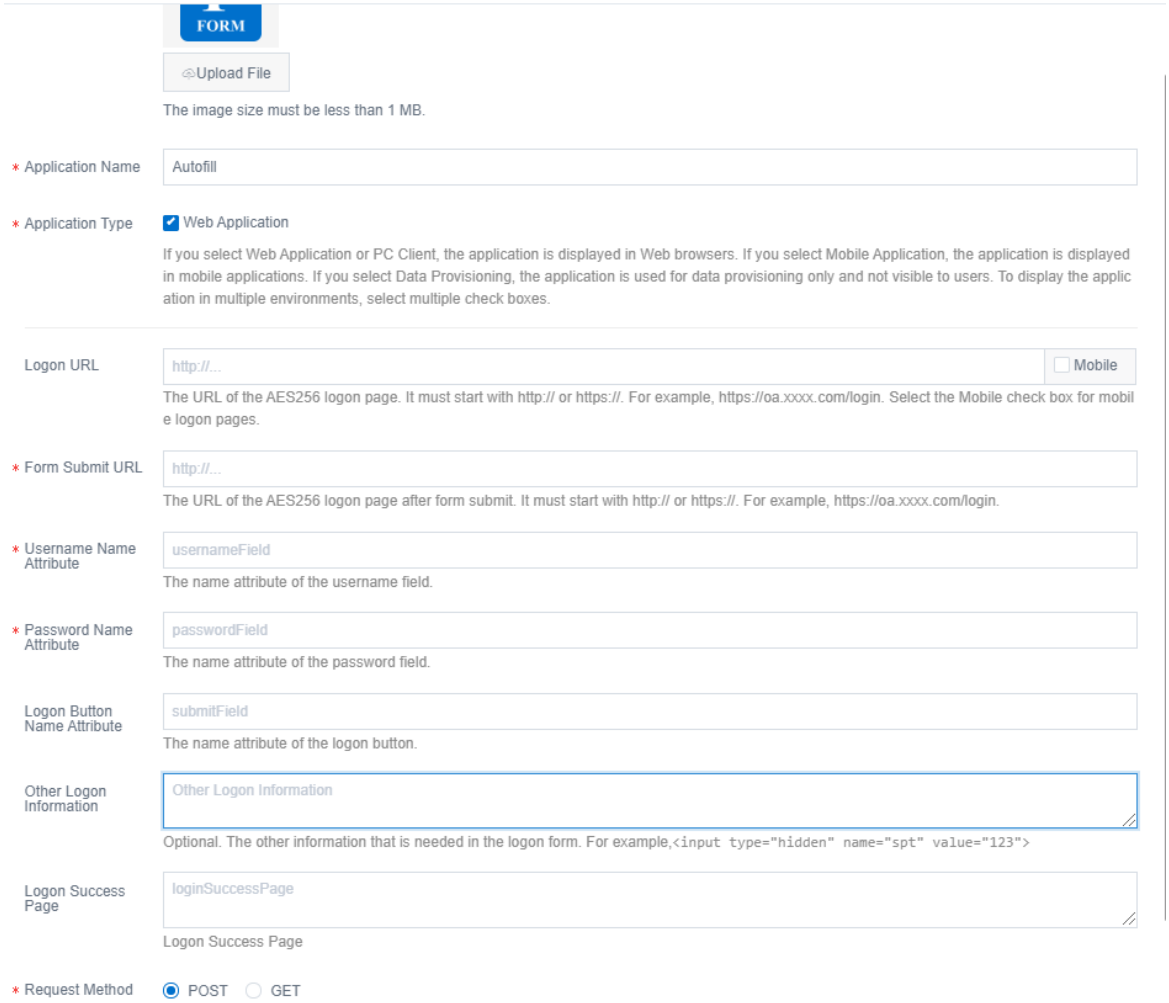
The Form Autofill application in IDaaS can be used to implement single sign-on and authentication for Application A.

Procedure

1. Log on to the IDaaS console as an IT administrator. For more information, see Logon in Administrator Guide.
2. In the left-side navigation pane, choose Applications > Add Application. Find the Form Autofill application and click Add Application in the Actions column.



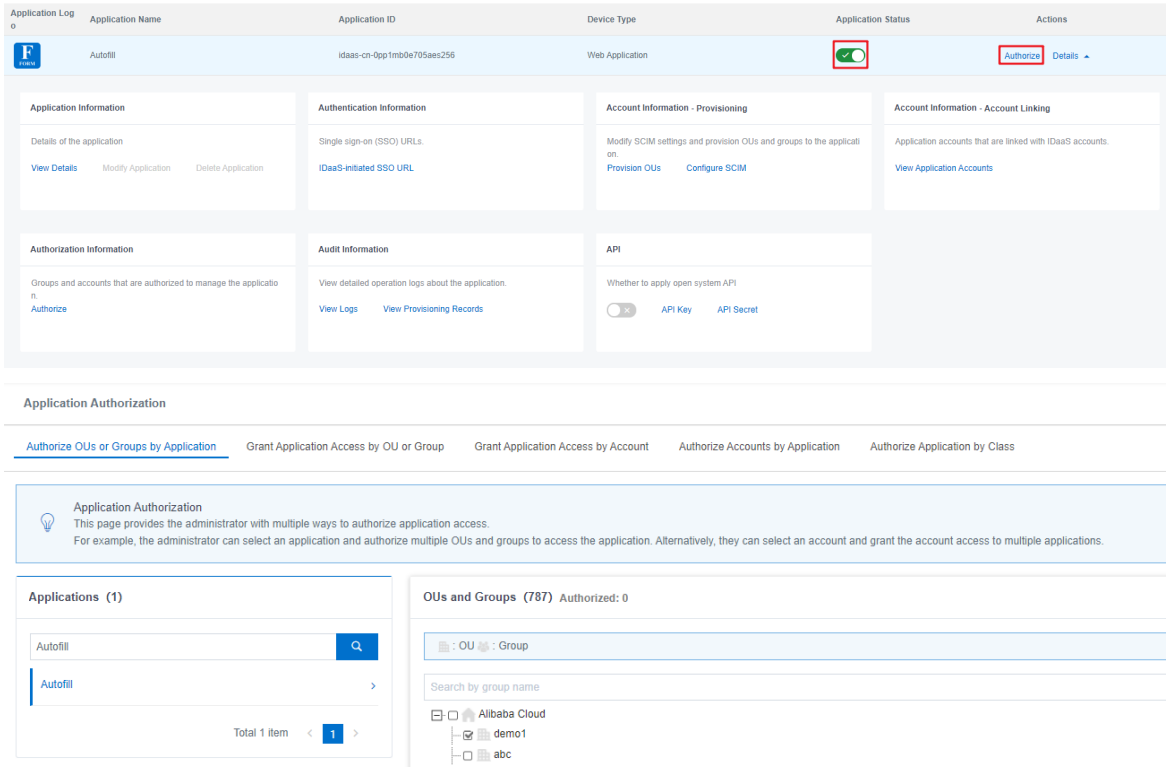
3. In the Add Application dialog box that appears, configure the following parameters:



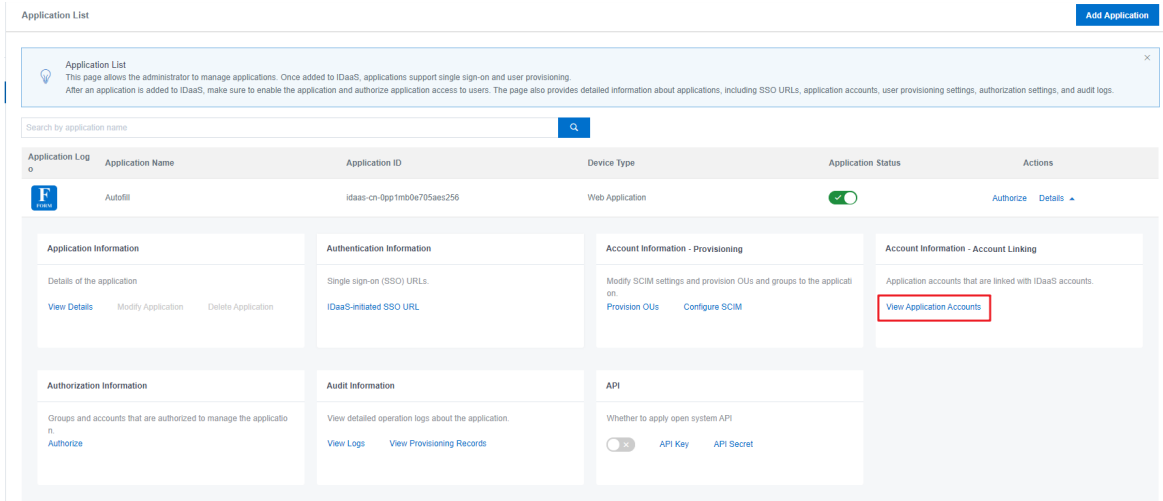
   - Form Submit URL: the logon URL of Application A.
   - Username Name Attribute: the username for the logon URL.
   - Password Name Attribute: the password for the logon URL.

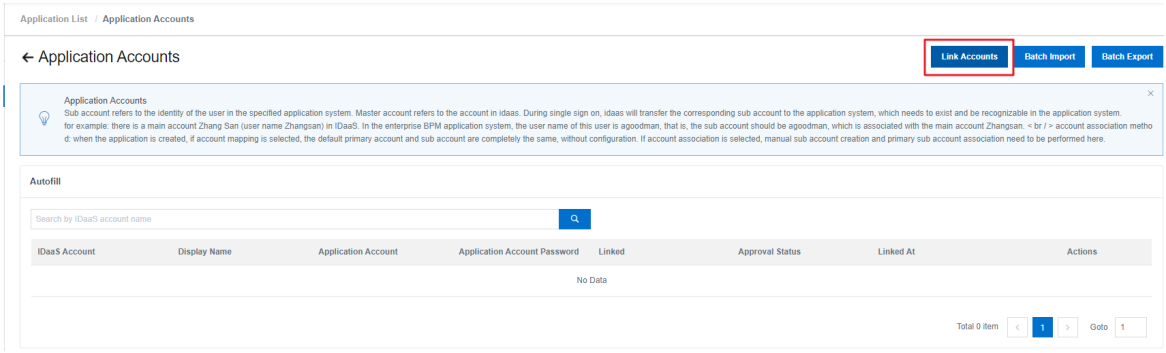
- Request Method: the request method for the logon URL.
- Account Linking Method: Select Account and Password.

4. Enable and authorize the application.

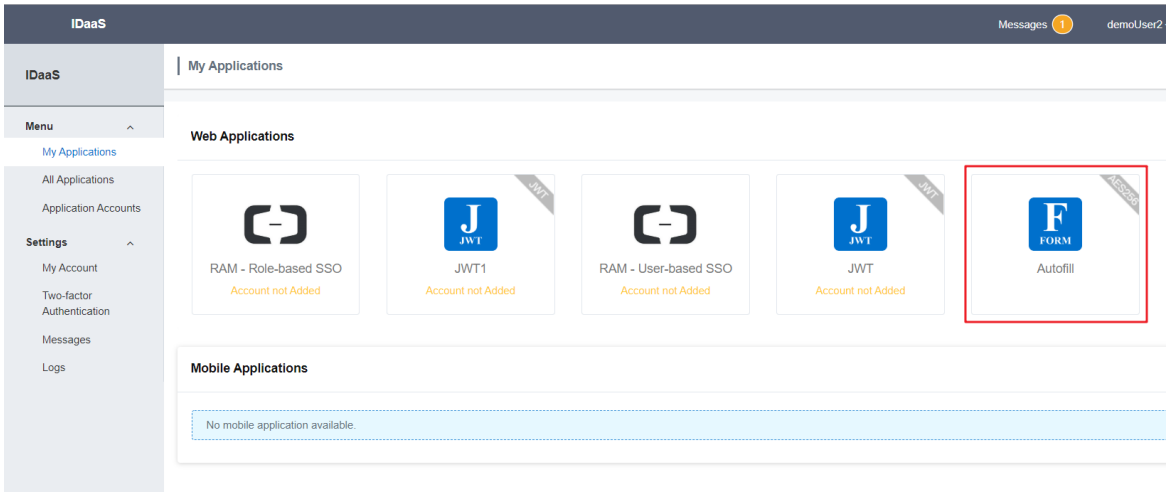


5. Bind the application account to the user account. The user account is the account used to access the IDaaS console, and the application account is the account for Application A.





6. Log on to IDaaS as the common user authorized to access the application and click the icon to log on to Application A in a single sign-on manner.



If all the preceding steps are successful, you have logged on to Application A in a single sign-on manner.