



PUBLIC

Document Version: 2402 – 2024-02

# Help for the SAP Ariba Developer Portal

## SAP Ariba developer portal

# Content

<b>Help for the SAP Ariba Developer Portal. . . . .</b>	<b>3</b>
SAP Ariba Developer Portal Guide for Administrators. . . . .	4
Registering Your Organization to Use the SAP Ariba Developer Portal . . . . .	5
Administrator Functions Related to SAP Ariba Developer Portal User Accounts. . . . .	5
Administrator Functions Related to Applications on the SAP Ariba Developer Portal . . . . .	10
Exporting Your Data from the SAP Ariba Developer Portal . . . . .	15
Browsing APIs on the SAP Ariba Developer Portal . . . . .	16
Management of Your Own APIs on the SAP Ariba Developer Portal . . . . .	19
Registering an RPA Bot in the SAP Ariba Developer Portal . . . . .	24
SAP Ariba Developer Portal Quick Start Guide for Developers. . . . .	26
Steps to Start Using the SAP Ariba APIs . . . . .	26
Progress of Your Application Tracker. . . . .	28
SAP Ariba Developer Portal Authentication. . . . .	30
Use of the API Gateway and OAuth to Authenticate Applications. . . . .	30
Finding Your Application's Application Key and OAuth Client ID. . . . .	31
Managing Security Certificates for Mutual Authentication. . . . .	31
Generating the <b>OAuth Secret</b> and <b>Base64 Encoded Client ID and Secret</b> . . . . .	33
OAuth Access Token Request and Receipt. . . . .	34
Making of REST API Calls with the OAuth Access Token and Application Key. . . . .	38
SAP Ariba Developer Portal Authentication Credentials Saving and Safeguarding. . . . .	39
API-Specific Disclaimers and Legal Information. . . . .	40

# Help for the SAP Ariba Developer Portal

The SAP Ariba developer portal gives your organization and its developers access to solution-extending APIs that can make your business more efficient and effective. This guide is for a customer seeking to extend the functionality of SAP Ariba solutions by calling the APIs published on the SAP Ariba developer portal .

The SAP Ariba developer portal provides details about SAP Ariba APIs, offers customers the ability to create and manage applications that use those APIs, register and manage the developer and administrator users who can execute those functions. It also provides a way to register outbound APIs for use in custom forms, and to register RPA bots that interact with SAP Ariba solutions.

This guide applies to:

- SAP Ariba developer portal
- SAP Ariba APIs

To use any of the APIs on the SAP Ariba developer portal, you need to create an application to make the web service calls. This document provides all the information you need to set up your organization to create applications that offer powerful extensions to your SAP Ariba solutions by calling on the APIs provided on the SAP Ariba developer portal.

This help is divided into three sections:

- An **administrator guide** that describes how to set up user accounts and register applications for use on the SAP Ariba developer portal: [SAP Ariba Developer Portal Guide for Administrators \[page 4\]](#)
- A **quick start guide for developers**, which provides step-by-step instructions explaining how to create an application that consumes the APIs provided via the SAP Ariba developer portal: [SAP Ariba Developer Portal Quick Start Guide for Developers \[page 26\]](#)
- An extensive chapter describing **how to incorporate the OAuth authentication protocol** into your applications. OAuth authentication is mandatory, so please pay special attention to this chapter: [SAP Ariba Developer Portal Authentication \[page 30\]](#)

## General Prerequisites

All use of the SAP Ariba developer portal requires the following prerequisites.

- Your organization must have a current license for one or more SAP Ariba solutions or an SAP Business Network solution component to use the APIs. Example solutions include SAP Ariba Buying, SAP Business Network Discovery, SAP Ariba Invoice Management, SAP Ariba Payables, and others.
- Your organization must be in the United States of America or another supported country/region.
- If your organization works with the public sector, you may need to fulfill specific prerequisites prior to using certain APIs. Details are outlined in the documentation associated with each API.
- An SAP Ariba APIs administrator account is required. Your organization's SAP Ariba administrator can request this account at the SAP Ariba developer portal: <https://developer.ariba.com/api>. Once your SAP Ariba APIs administrator access to the SAP Ariba developer portal has been established, additional developer accounts can be added to your organization.

### Note

There are instances of the SAP Ariba developer portal in various regions. Your administrator should request an account in the region where your site is located and your applications will be hosted. For access to APIs related to the SAP Business Network, request an account in the US data center regardless of where your site is located and your applications will be hosted. This may mean requesting separate accounts if your applications are hosted outside the US region, one in the US data center for APIs related to the SAP Business Network and one in your data center for APIs related to SAP Ariba applications.

- You must use a compatible browser. SAP Ariba APIs supports the following browsers:
  - Firefox 47.0.1
  - Chrome 63.0.3
  - Safari 11.0.2
  - IE 11.0.9600

### Note

Some of the individual APIs presented on the SAP Ariba developer portal require additional prerequisites. See the documentation for each API for specifics.

## SAP Ariba Developer Portal Guide for Administrators

This chapter describes the various functions that a user with the **Organization Admin** role can perform in the SAP Ariba developer portal .

[Registering Your Organization to Use the SAP Ariba Developer Portal \[page 5\]](#)

[Administrator Functions Related to SAP Ariba Developer Portal User Accounts \[page 5\]](#)

[Administrator Functions Related to Applications on the SAP Ariba Developer Portal \[page 10\]](#)

[Exporting Your Data from the SAP Ariba Developer Portal \[page 15\]](#)

[Browsing APIs on the SAP Ariba Developer Portal \[page 16\]](#)

[Management of Your Own APIs on the SAP Ariba Developer Portal \[page 19\]](#)

[Registering an RPA Bot in the SAP Ariba Developer Portal \[page 24\]](#)

# Registering Your Organization to Use the SAP Ariba Developer Portal

Use this procedure to register your organization to use the SAP Ariba developer portal.

## Context

Follow these steps to register your organization to use the SAP Ariba developer portal.

### Note

This procedure is not required for FedRamp users, for whom accounts are created automatically upon initial Single Sign-On authentication.

## Procedure

1. At <https://developer.ariba.com/api>, choose your region from the **Portals** dropdown.
2. In response to **Don't have an account?** click **Sign Up**
3. Choose **Request an account** and fill out the form, then click **Submit**.
4. You will receive a confirmation email in response to the form.
5. At <https://developer.ariba.com/api>, choose your region from the **Portals** menu and log in using single sign-on.
6. Agree to the Terms of Service for your region.

## Administrator Functions Related to SAP Ariba Developer Portal User Accounts

From directly within the SAP Ariba developer portal, you can browse the users associated with your organization. You can assign them roles and applications here as well. However, most SAP Ariba developer portal user management functions are executed in SAP Ariba launchpad or the SAP ID service. You can set up multi-factor authentication (MFA) using the SAP ID service.

The SAP Ariba developer portal uses the time-based one-time password (TOTP) algorithm to implement MFA. For detailed information about TOTP and where to seek support, see <https://support.sap.com/en/my-support/mfa.html>.

## Setting Up TOTP Two-Factor Authentication

The SAP Ariba developer portal uses TOTP two-factor authentication. An existing user can set this up at <https://accounts.sap.com/ui/protected/profilemanagement>. In the **Multi-Factor Authentication** section, find **TOTP Two-Factor Authentication** and click **Activate**.

## Registering a New User

To create a new user, do one of the following:

### Note

This procedure is not required for FedRamp users, for whom accounts are created automatically upon initial Single Sign-On authentication.

- Click the **+** sign to the right of the search field. This redirects you to SAP Ariba Connect.
- From the **Actions** menu, choose **Manage your users**. This redirects you to SAP Ariba Connect.
- Open a browser window directly to the New User Registration page at <https://support.sap.com/en/my-support/users.html>.

Choose **New User**.

## Managing Users

Most user management functionality is executed using SAP Launchpad. You can access user management functionality on SAP Launchpad by choosing the **Actions > Manage your users** menu option or by clicking the plus sign.

To view and set a user's role, see [Setting User Roles on the SAP Ariba Developer Portal \[page 8\]](#)

To deactivate a user, use the **Actions > Manage your users** menu option.

## Browsing Users

You can browse user details directly in the SAP Ariba developer portal, or you can download certain user details as a CSV file:

- Browse your current user accounts by clicking the desired user on the left. [Browsing User Accounts in the SAP Ariba Developer Portal \[page 7\]](#)
- Download the first name, last name, and email address of the displayed user as a CSV file. [Downloading User Information from the SAP Ariba Developer Portal \[page 8\]](#)
- Generate and download a report in CSV format, containing all actions taken by the displayed user since the user was created. [Generating a User Audit Log \[page 9\]](#)

## Related Information

[User Roles on the SAP Ariba Developer Portal \[page 7\]](#)

# User Roles on the SAP Ariba Developer Portal

The SAP Ariba developer portal has two user roles:

- **Organization Admin**
- **Developer**

**Organization Admin** users can manage all user accounts and applications associated with your organization. **Developer** users can manage only those applications that are assigned to them.

## Browsing User Accounts in the SAP Ariba Developer Portal

Use this procedure to browse your current user accounts by clicking the desired user on the left.

### Context

To browse the user accounts currently in your organization, follow these steps:

### Procedure

1. Log in to the SAP Ariba developer portal as a user with the **Organization Admin** role.
2. In the left-hand navigation area, click **Manage**, then navigate to the **Users** tab. A list of users appears on the left.
3. To view the desired user, do one of the following:
  - If the desired users is listed, click it in the list to view details.
  - Scroll up or down using the scrollbar until the desired user is visible in the list, then click it in the list to view details.
  - Enter the name of the desired user in the **Search** field, then click the magnifying glass icon to filter the list. When the desired user is visible in the list, click it to view details.

## Results

The desired user is highlighted in the list, and details are displayed on the right.

## Downloading User Information from the SAP Ariba Developer Portal

Use this procedure to download the first name, last name, and email address of the displayed user as a CSV file.

### Context

To download the first name, last name, and email address of the displayed user as a CSV file, follow these steps:

### Procedure

1. Log in to the SAP Ariba developer portal as a user with the **Organization Admin** role.
2. In the left-hand navigation area, click **Manage**, then navigate to the **Users** tab. A list of users appears on the left.
3. Click the desired user in the list on the left. User details appear on the right.
4. Click **Actions** > **Download Personal Information**.

## Results

Your browser downloads the data in CSV format.

## Setting User Roles on the SAP Ariba Developer Portal

Use this procedure to view and set a user's roles using the toggles in the **Roles** section of the user profile.

### Context

A user with the **Organization Admin** role can assign a user the role of **Organization Admin** or **Developer**.



## Procedure

1. Log in to the SAP Ariba developer portal as a user with the **Organization Admin** role.
2. In the left-hand navigation area, click **Manage**, then navigate to the **Users** tab. A list of users appears on the left.
3. Click the desired user in the list on the left. User details appear on the right.
4. The **Roles** section of the user profile includes toggles for **Organization Admin** and **Developer**. If the toggle for a role is blue and the switch is on the right side, the user has that role. If the toggle for a role is gray and the switch is on the left side, the user does not have that role. Turn on the toggles for the roles you wish this user to have.

## Related Information

[User Roles on the SAP Ariba Developer Portal \[page 7\]](#)




## Generating a User Audit Log

Use this procedure to generate and download a report in CSV format, containing all actions taken by the displayed user since the user was created.

## Context

To generate and download a user's audit log, follow these instructions:

## Procedure

1. Log in to the SAP Ariba developer portal as a user with the **Organization Admin** role.
2. In the left-hand navigation area, click **Manage**, then navigate to the **Users** tab. A list of users appears on the left.
3. Click the desired user in the list on the left. User details appear on the right.
4. Click  **Actions**  **Download audit log** 

## Results

Your browser downloads the audit log in CSV format.

# Administrator Functions Related to Applications on the SAP Ariba Developer Portal

To access **Organization Admin** functions related to applications, log in to the SAP Ariba developer portal as a user with the **Organization Admin** role and choose **Manage** from the left-hand navigation area. From the **Manage Applications** tab, you can perform the following actions related to your organization's applications:

- Create an application. [Creating an Application on the SAP Ariba Developer Portal \[page 27\]](#)

## Note

Users with the **Developer** role can also perform this action.

- View detailed information about an application. [Viewing Application Details on the SAP Ariba Developer Portal \[page 10\]](#)

## Note

Users with the **Developer** role can also perform this action.

- Delete the displayed application. [Deleting an Application from the SAP Ariba Developer Portal \[page 11\]](#)
- Reassign the displayed application to a different developer within your organization. [Reassigning an Application on the SAP Ariba Developer Portal \[page 12\]](#)
- Begin the process of publishing the displayed application. [Requesting API Access for an Application on the SAP Ariba Developer Portal \[page 13\]](#)
- Cancel a API access for an application. Users with the **Organization admin** role can cancel the API access request for a client application. Display the relevant application on the **Applications** list, then click **Cancel**.
- Generate an OAuth secret for the displayed application. All applications must authenticate to the production server using OAuth. This option is available only after the request for API access has been granted and approved. [How to generate the OAuth Secret and Base64 Encoded Client and Secret \[page 33\]](#)

## Viewing Application Details on the SAP Ariba Developer Portal

Use this procedure to view detailed information including the name and description of the application, the developer who created it, the most recent change date, and the application's API Key.

### Context

To view the details of an existing application on the SAP Ariba developer portal, follow these steps:

### Procedure

1. Log in to the SAP Ariba developer portal.

2. Navigate to the application list by doing one of the following:
  - If you are a **Developer** user, click **Manage** in the left-hand navigation area.
  - If you are an **Organization admin** user, navigate to ► **Manage** ► **Applications** ▾.
3. Locate the desired application in the application list on the left. You may need to search or scroll through the list using the scrollbar or the search and filter controls.
4. Click the desired application in the application list on the left.

## Results

Application details are displayed on the right, including the name and description of the application, the developer who created it, the most recent change date, and the application's API Key. The **What's Next?** area displays a graphical progress tracker for the application. See [Progress of Your Application Tracker \[page 28\]](#).

## Deleting an Application from the SAP Ariba Developer Portal

Use this procedure to delete the displayed application.

### Context

To delete an application from the SAP Ariba developer portal, follow these steps:

### Procedure

1. Log in to the SAP Ariba developer portal as a user with the **Organization admin** role.
2. Navigate to ► **Manage** ► **Applications** ▾.
3. Locate the desired application in the application list on the left. You may need to search or scroll through the list using the scrollbar or the search and filter controls.
4. Delete the displayed application by clicking ► **Actions** ► **Delete application** ▾.

## Results

The application is deleted, and no longer appears in the list of applications.

# Reassigning an Application on the SAP Ariba Developer Portal

Use this procedure to reassign the displayed application to a different developer within your organization.

## Context

To reassign an application, follow these steps:

## Procedure

1. Log in to the SAP Ariba developer portal as a user with the **Organization admin** role.
2. Navigate to ► **Manage** ► **Applications** ►.
3. Locate the desired application in the application list on the left. You may need to search or scroll through the list using the scrollbar or the search and filter controls.
4. Click ► **Actions** ► **Assign this application to another developer** ►.
5. Choose the desired new developer from the list.

## Results

The new developer's name appears in the application details. The application will now appear in the **My applications** list of the new developer. It will no longer appear in the **My applications** list of the previous developer.

# Requesting API Access for an Application on the SAP Ariba Developer Portal

Begin the process of publishing the displayed application by using this procedure to request API access.

## Context

Customers requesting API access for applications may have only one application per realm/API combination. To request API access for an application, follow these steps:

## Procedure

1. Log in to the SAP Ariba developer portal as a user with the **Organization admin** role.
2. Navigate to **Manage > Applications**.
3. Locate the desired application in the application list on the left. You may need to search or scroll through the list using the scrollbar or the search and filter controls.
4. Click **Actions > Request API access**.
5. Fill out the **Application Details** dialog as follows:

- From the **Select an API** dropdown, choose the name of the API you wish to use.
- Choose your solution or solutions from the checkboxes. If a solution is not selectable, that means it has not been configured for use with the API you chose. Solutions that may be available are as follows:

<b>Procurement</b>	When you check the box for <b>Procurement</b> , a dropdown list of procurement realms opens. Choose the desired procurement realms from this list.
<b>Sourcing</b>	When you check the box for <b>Sourcing</b> , a dropdown list of sourcing realms opens. Choose the desired sourcing realms from this list.
<b>SAP Business Network (formerly Ariba Network)</b>	When you check the box for <b>SAP Business Network (formerly Ariba Network)</b> , the <b>AN-ID</b> field appears. Enter your ANID in this field.
<b>Customer ID</b>	When you check the box for <b>Customer ID</b> , a field appears where you can enter your customer ID. This triggers a support request for your API access.

### Note

All solutions are present, but only those for which the selected API has been configured can be chosen.

- Optional. If you have additional comments, you can type them in the provided text entry field.
6. When you are ready, click **Submit** to begin the process of requesting API access for your application. To cancel, click **Cancel**.

## Results

When the request is approved, you will receive email with further instructions. The application cannot be used in the production environment until this request is approved. Additionally, email is sent to all **Organization admin** users who have logged in and accepted the most recent terms of use. This provides accountability and increases the security of your data.

## Applying a List of Allowed IPs to an Application on the SAP Ariba Developer Portal

Use this procedure to apply a list of allowed IPs to an application on the SAP Ariba developer portal. Only requests that originate from IP addresses within the approved CIDR ranges on an application's list of allowed IPs are granted access to your application. If an application's list of allowed IPs contains no CIDR ranges, requests may originate from any IP address.

### Prerequisites

- You must be an **Organization admin** user to work with list of allowed IPs on the SAP Ariba developer portal.
- CIDR ranges must be valid, and IP addresses must be IPv4.
- Applications that originate from an IP address within one of the CIDR ranges on the list of allowed IPs must still present a valid API key and fulfill OAuth authentication processes to call SAP Ariba APIs.
- At most 30 CIDR ranges may be added to any single list of allowed IPs.

### Context

Follow these steps to add, edit, or delete CIDR ranges from an application's list of allowed IPs.

### Procedure

1. Log in to the SAP Ariba developer portal as a user in the **Organization Admin** group.
2. Click **Manage** > **Applications** and then open the desired application.
3. From the **Actions** menu, choose **Allowed IPs**.
4. You can perform the following functions in the **Allowed CIDR Ranges (IPv4)** dialog:
  - To add a CIDR range to the allowlist, type it in the **+ new range** field. When you type `Enter` or `space`, the new CIDR range appears as a button within the **CIDR Range(s)** box.

### Note

For more information about how CIDR ranges represent multiple IPs, you can read online about CIDR Notation.

- To delete a CIDR range from the allowlist, click the **X** icon on its button.
  - To edit a CIDR range that is already on the allowlist, click on its button (but not on the **X** icon), then make your changes. When you type `Enter` or `space`, the edited CIDR range becomes a button again.
  - To remove all CIDR ranges from the allowlist, click **Clear all**.
5. Optional. To exit the **Allowed CIDR Ranges (IPv4)** dialog without making any changes, click **Cancel**.
  6. When you are finished making changes, click **Save**.

The **Allowed CIDR Ranges (IPv4)** dialog is dismissed, and the current application is displayed. The new list of allowed IPs applies to the displayed application.

API calls that originate from an IP address that is not included in one of the allowed CIDR ranges will receive the following error message in the response: `Access is Denied. Please contact your Organization admin.`

## Exporting Your Data from the SAP Ariba Developer Portal

Use this procedure to generate and export a JSON file containing information about your users, outbound APIs, applications, and registered BOTs.

### Context

The ability to download customer data is a useful data privacy tool. For example, if you want to delete sensitive information from the SAP Ariba developer portal you can download it first, thereby preserving both data privacy and your own access to the information.

Sample fields extracted for each category include the following:

Category	Sample Fields
Users	<i>firstname</i> The user's given or first name
	<i>lastname</i> The user's surname
Applications	<i>uniqueName</i> The unique name of the application
	<i>description</i> The description entered for the application in the SAP Ariba developer portal
Outbound APIs	<i>name</i> The name of the outbound API
	<i>upstreamUrl</i> The URL used to execute queries to the outbound API

Category	Sample Fields	
Bots	<i>botName</i>	The name used to register the bot on the SAP Ariba developer portal
	<i>username</i>	The user name for the account the bot uses

The extracted data includes many more fields than the samples listed here. You can see the full set of fields by downloading your data and reading the JSON file.

Follow these steps to generate and export a JSON file containing information about the users, applications, outbound APIs, and BOTs you have registered on the SAP Ariba developer portal.

## Procedure

1. From the gear-shaped settings menu near the top of the developer portal, choose **Download My Data**.
2. If your browser is configured to prompt you to choose a download destination, do so. Otherwise, the file is saved in the destination folder configured in your browser settings.

# Browsing APIs on the SAP Ariba Developer Portal

Use this procedure to browse APIs on the SAP Ariba developer portal.

## Procedure

1. Click **Discover** in the left-hand navigation pane.
2. The tabs along the top of the screen organize available APIs into functional categories. Click the tab for a category.
3. The list of APIs for the category is displayed on the left. Click the name of the desired API to view its discovery page.
4. Each API discovery page includes a brief description of the API's functionality, a link to more detailed help, and the following sections:

**Description** Includes a brief description of the API and a link to documentation specific to the API. Also includes rate limit information. Best practice is to track your API usage to ensure you don't exceed these rate limits. See [Tracking Your API Usage to Avoid Exceeding Rate Limits \[page 28\]](#).

**Environment Details** Displays the public URI prefixes for the testing and production environments for this API.

**Download API spec** Click to download request and response schemas in JSON format.

**Detailed Documentation** Displays the URL endpoints for use when making web service calls. Click on any method in this section for syntax and parameter information.



### Try it out

You can click **Try it out** within any expanded method in the **Detailed Documentation** section to investigate the method for yourself by providing inputs and viewing the resulting output. For the `Realm` parameter, enter `mytestrealm`.

### Models

Drill down by clicking within this section to see details including implementation notes and schemas for the response class and the response message.

## API Versioning and Deprecation on the SAP Ariba Developer Portal

When a new version of an API is released, the previous version is deprecated; a single SAP Ariba developer portal discovery page hosts both versions.

This topic describes the the policies and best practices that apply when a new version on an API is published on the SAP Ariba developer portal, triggering the deprecation and eventual decommissioning of the previous version.

You can view a list of deprecated and decommissioned APIs by choosing **Deprecated & Decommissioned APIs** from the gear-shaped settings menu near the top of the SAP Ariba developer portal. This list displays each API that has one or more versions in the deprecated or decommissioned states. Users can drill down within a listed API to see the deprecated and decommissioned versions. Clicking **View active version** opens the discovery page for the active version of the selected API.

### Versioning

A version indicator located directly below the API name identifies which version of the API is currently displayed on the discovery page for an API. When only one version is available, this version indicator is a simple label. When multiple versions are available, it is a dropdown labeled **Version X** where **X** is the displayed version number. Users can opt to switch to a different version by choosing from this dropdown. The most recent version is displayed by default.

Information such as runtime URLs, endpoints, and models presented on the discovery page relate to the displayed version.

To write an application using a particular version of an API, use the runtime URLs specific to that version. Different versions distinguish their runtime URLs by modifying the version number. For example:

Version 1

```
https://openapi.ariba.com/api/hypothetical_api/v1/prod
```

Version 2

```
https://openapi.ariba.com/api/hypothetical_api/v2/prod
```

## Deprecation

When a new version of an API is published on the SAP Ariba developer portal, the old version is deprecated, with the following effects:

- The deprecated version is marked **Deprecated** in the version control dropdown on its SAP Ariba developer portal discovery page.
- If you navigate to the discovery page for the deprecated API by choosing the deprecated version from the version control dropdown, a message indicating the date of deprecation appears at the top of the page.
- Existing client applications that call the deprecated API will be supported for 12 months and will work until the API is decommissioned. If you call a deprecated API, the response includes an `X-API-WARN` header indicating that the version is deprecated and identifying the active version.
- However, no new client applications can request access to the deprecated version of the API; all new client applications must request the new version.

## Decommissioning

At the discretion of SAP Ariba, the API version is eligible to be decommissioned after being in deprecated or active state for a minimum period of 24 months which must include at least 12 months in a deprecated state. Decommissioning an API version has the following effects:

- The decommissioned version is marked **Decommissioned** in the version control dropdown on its SAP Ariba developer portal discovery page.
- The decommissioned API should not be called at all, and if called might not provide expected responses. Existing client applications that use the decommissioned API may no longer function. If you call a decommissioned API, the response includes an `X-API-WARN` header indicating that the version is decommissioned and identifying the active version.
- No new client applications can request access to the decommissioned version of the API; all new client applications must request the new version.

## Accessibility Features of the SAP Ariba Developer Portal

You can navigate the SAP Ariba developer portal in various ways, depending on your accessibility preferences.

In addition to point-and-click mouse pointer navigation and visual screen display, the SAP Ariba developer portal supports keystroke navigation and screen readers.

### Keyboard Navigation

- Use the `Enter` key to choose the control that currently has focus.
- Use the `Tab` key to move focus to the next control on the screen.
- Use `Shift` + `Tab` to move to the previous control.

- If the focus is on a control within a list group (menu list, API list, etc), use `[Shift] + [Right Arrow]` to escape the current list and skip to the next focusable element after the list group. For example, on the **Applications** screen, you can use this key sequence to exit the list of applications and move focus to the **Actions** menu dropdown.
- Similarly, you can use `[Shift] + [Up Arrow]` to escape the current list group and move focus to the focusable element above the list group. For example, on the **Applications** screen, you can use this key sequence to exit the list of applications and move focus to the **+** (add) button.

## Screen Reader Compatibility

- In addition to text content, screen readers can read titles, dialog headers, image descriptions, and group labels.
- In particular, screen readers can step through the graphical progress tracker in the **What's Next** section when you are tracking the creation status of an application. For each step on the graphical progress tracker, the screen reader can read the step name and announce its status (**Done**, **Pending**, and so on).
- Screen readers can announce element types when appropriate. For example, menu items are called out as menu items, list items as list items, and navigation tabs as navigation tabs.
- Screen readers can announce which item within a list group (menu list, API list, etc) has been selected, and the position of the selected item within its list group (for example 2 of 7).

## Additional Accessibility Features

- You can enable or disable captions while viewing videos.
- On the signup page, you can choose between audio captcha and video captcha challenges.

## Related Information

[Progress of Your Application Tracker \[page 28\]](#)

# Management of Your Own APIs on the SAP Ariba Developer Portal

Customers in an SAP Ariba buyer organization might need access, within their custom forms, to their own APIs. This feature allows you to publish your own RESTful APIs that call your internal systems, to the SAP Ariba developer portal for use in your custom form applications.

To access **Organization Admin** functions related to your APIs, log in to the SAP Ariba developer portal as a user with the **Organization Admin** role and choose **Manage** from the left-hand navigation area. From the **My APIs** tab, you can perform the following actions related to your APIs:

- Publish a new outbound API. For detailed instructions, see [Publishing an Outbound API to Your My APIs List \[page 21\]](#).
- Browse your current external APIs by clicking the desired API on the left. You can see authentication details and target URL in this area. To see the schemas associated with the API, click the desired method in the **Detailed documentation** area.

#### Note

For troubleshooting purposes, SAP Ariba Support can create and process applications on behalf of a customer organization. These may appear on the SAP Ariba developer portal on the customer's **Applications** list. The name of an SAP Ariba Support application will always begin with the prefix **Ariba Support -**. Customers should ignore these applications, which will generally be deleted after troubleshooting is complete.

- Filter your current external APIs using their assigned tags. To find all of your APIs that have been assigned a particular tag, type the desired tag into the **Search** field, then hit  or click the magnifying glass icon. The page now displays only those APIs that have been assigned the tag you specified.
- Search for a specific external API by name. To find a specific API by name, type the desired API name into the **Search** field, then hit  or click the magnifying glass icon.
- Edit the displayed API by choosing **Edit API** from the **Actions** menu.
- Deactivate the displayed API, so that it remains visible on the **My APIs** tab but cannot be called during runtime by SAP Ariba applications. Choose **Deactivate API** from the **Actions** menu.
- Delete the displayed API by choosing **Delete API** from the **Actions** menu. If you delete an API, any applications that rely on the deleted API will no longer be functional.
- Download the swagger document associated with the displayed API by choosing **Download document** from the **Actions** menu.

## Prerequisites

To use this feature, you must fulfill the following prerequisites:

- You must be registered to use the SAP Ariba developer portal.
- You must have the **Organization Admin** role to use this feature.
- To use this feature with Custom Forms, Custom Forms must be configured to call an external API.
- Before the first time you use this feature you will be required to accept the new Terms of Use.

## Limitations

APIs published on the **My APIs** tab can be called only from Custom Forms.

These APIs must use the REST protocol.

These APIs may not duplicate the functionality of an SAP Ariba API in any way.

SAP Ariba recommends against the exchange of personal data using these APIs. If such information is passed or received using calls to these API, the customer takes full responsibility for that data.

Upon request by SAP Ariba, Customer will provide SAP Ariba with documentation as to any External API call and confirmation as to license rights obtained for connection to the SAP Ariba Cloud Service. SAP Ariba may implement new requirements or a certification process regarding external API calls with advance notice to Customer. SAP Ariba may disable or reject use of any external API call in SAP's discretion or to protect the SAP Ariba system operations or security.

## Publishing an Outbound API to Your My APIs List

Use this procedure to publish an outbound API to your **My APIs** list. You can use your outbound APIs within your custom forms.

### Context

To publish your own RESTful APIs that call your internal systems, to your **My APIs** list on the SAP Ariba developer portal for use in your custom form applications, follow these instructions:

### Procedure

1. Log in to the SAP Ariba developer portal as a user with the **Organization Admin** group.
2. Choose **Manage** from the left-hand navigation area.
3. On the **My APIs** tab, do one of the following to open the **Publish API** form:
  - If this is your first outbound API, click **Get Started**
  - Otherwise, click the plus sign icon located near the **Search** bar on the left.
4. Fill out the **Publish API** form as follows:

Field	Description
<b>API Name</b>	Required. Enter a name for the API.
<b>Target URL</b>	Required. This is the URL for the API you want to publish, such as <code>https://www.someapipurveyor.com/api/v2</code> . This URL must provide a complete path to the API. It may not include parameters. Parameters should be entered in the <b>URL query parameters</b> section or added at runtime. An internet protocol such as <code>https</code> is required.
<b>Description</b>	Required. A brief description of the API.
<b>Allowed Tenants</b>	Required. Select the realms from which you wish this API to be accessible.

Field	Description
Tags	<p>Optional. You can add tags to the API. Later, you can filter your APIs using these tags. To add a tag, type it in the <b>Tags</b> field and then hit the <code>Space</code> bar. To remove a tag, click the <b>x</b> in the tag you want to remove.</p> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>Note</b></p> <p>Tags may not contain spaces.</p> </div>
Header parameters	<p>Optional. You can enter static headers and values as desired in the <b>Header parameters</b> table. To add additional rows, click <b>Add a row</b> for each. Do not add headers with variable values to this table. Headers entered in this table will be submitted with every request to the API. Additional headers can also be sent at runtime.</p>
URL query parameters	<p>Optional. You can enter static URL query parameters and values in the <b>URL query parameters</b> table. To add additional rows, click <b>Add a row</b> for each. Do not add parameters with variable values to this table. Parameters and their values entered in this table will be submitted with every request to the API. Additional parameters can also be specified at runtime.</p>
Mutual Authentication	<p>Optional. If you wish to enable mutual authentication for this API, choose one from this list of public signed certificates.</p> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>• If the certificate you want is not in the dropdown list, contact the certificate team via email or Slack channel.</li> <li>• If you opt for mutual authentication, set it up here first, before setting it up in your API.</li> <li>• If you do not enable mutual authentication by choosing a certificate, the authentication type <b>None</b> will not be available in the <b>Authentication type</b> section.</li> </ul> </div>

Field	Description
<b>Authentication type</b>	This dropdown allows you to instruct SAP Ariba to perform authentication tasks. Possible values are:
<b>None</b>	Choose this if you do not need SAP Ariba to perform authentication tasks.
	<div data-bbox="1019 415 1419 617" style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px;"> <p data-bbox="1040 426 1138 457"><b>Note</b></p> <p data-bbox="1040 474 1386 590">This option is not available unless you enabled mutual authentication by choosing a certificate in the <b>Mutual Authentication</b> section.</p> </div>
<b>Basic authentication</b>	Choose this if your API requires HTTP basic authentication. Enter username and password in the form.
<b>OAuth 2.0</b>	<p data-bbox="1019 745 1419 892">Choose this if your API requires OAuth 2.0 authentication. Complete the fields with the data required to authenticate to the external APIs authentication server, as follows:</p> <ul data-bbox="1029 903 1419 1885" style="list-style-type: none"> <li data-bbox="1029 903 1419 997">• <b>Token endpoint</b> (required): Enter the endpoint used to retrieve the access token for your API.</li> <li data-bbox="1029 1003 1419 1119">• <b>Client Id</b> (required): Enter client ID for your API. This ID should be provided by the administrator of the external API.</li> <li data-bbox="1029 1125 1419 1241">• <b>Client secret</b> (required): Enter the client secret for your API. This secret should be provided by the administrator of the external API.</li> <li data-bbox="1029 1247 1419 1310">• <b>Grant type</b> (required): Enter the grant type for your API.</li> <li data-bbox="1029 1316 1419 1411">• <b>HTTP Request Method</b>: Choose <b>POST</b> or <b>GET</b> from the dropdown. Default is <b>POST</b>.</li> <li data-bbox="1029 1417 1419 1533">• <b>Body Content Type</b>: Choose <b>FORM_URL_ENCODED</b> or <b>JSON</b> from the dropdown. Default is <b>FORM_URL_ENCODED</b>.</li> <li data-bbox="1029 1539 1419 1654">• <b>OAuth header parameters</b> (optional): Header parameters and their assigned values entered in this table will be sent with the access token.</li> <li data-bbox="1029 1661 1419 1818">• <b>OAuth URL query parameters</b> (optional): URL query parameters and their assigned values entered in this table will be sent with the access token.</li> <li data-bbox="1029 1824 1419 1885">• <b>OAuth form parameters</b> (optional): Form parameters and their assigned</li> </ul>

Field	Description
	values entered in this table will be sent with the access token.
	<div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px;"> <p><b>Note</b></p> <p>If your API does not conform to the OAuth 2.0 standard, this authentication option will not succeed.</p> </div>
<b>Browse file...</b>	Required. Upload a swagger file that provides the methods and schema for the API

- To publish, click **Publish** at the bottom of the dialog. To cancel, click **Cancel**. Once the API is published, it can be used by SAP Ariba in your custom form setup.

## Results

Your API appears on your **My APIs** list. If you enabled mutual authentication and you wish to import the certificate into your upstream server truststore, the certificate download link can be found on the details page for the API. The certificate expiration date can be found on the same page.

# Registering an RPA Bot in the SAP Ariba Developer Portal

Use this procedure to register your RPA bot.

## Context

Follow these steps to register your RPA bot on the SAP Ariba developer portal:

## Procedure

- Log in to the SAP Ariba developer portal as a user in the **Organization Admin** group.
- If you have not already created a user for your RPA bot, do so now. See [Administrator Functions Related to SAP Ariba Developer Portal User Accounts \[page 5\]](#) for instructions.
- Navigate to the RPA Bot registration page by clicking **Manage** in the left-hand navigation area, then clicking the **Bots for RPA** tab.
- Open the bot registration form by clicking the plus sign icon.
- Fill out the form, as follows:



<b>RPA bot name</b>	Enter the unique user name you used when creating the user your bot will use to authenticate on the SAP Ariba developer portal.
<b>Which SAP Ariba application are you automating for?</b>	Enter the name of the SAP Ariba application you intend your bot to act on. For example, SAP Ariba Sourcing or SAP Ariba Buying and Invoicing.
<b>Realm name</b>	Select the name of the realm in which your bot will operate.
<b>What is the unique user name used to log in as the RPA bot?</b>	Enter the username you created for this bot.
<b>What RPA technology will your bot use?</b>	Specify the RPA technology your bot will use. Example technologies include but are not limited to SAP iRPA, BluePrism, and UIPath.

#### ⓘ Note

Although various RPA technologies are possible, SAP can provide technical support only for SAP iRPA (via component CA-ML-IPA). See [Best Practices for Using SAP IRPA Bots in SAP Ariba Solutions \[page 25\]](#) for details about using SAP iRPA. If you opt to use another RPA technology, you do so at your own risk. You will not be able to report defects or seek support from SAP Ariba Support; instead seek support from the makers of the RPA technology you have chosen.

<b>What business process will your RPA bot automate?</b>	Describe the task your bot will perform. What business problem are you trying to solve and automate by using a bot?
<b>What is the expected business value?</b>	Explain what makes it useful to automate this task using an RPA bot.

#### ⓘ Note

All fields are required.

- To complete your registration, click **Submit**.

## Results

On success, your new RPA bot appears in the list of RPA bots registered for your organization. Remember to use only one RPA bot at a time, as using multiple bots could adversely affect system performance.

## Best Practices for Using SAP IRPA Bots in SAP Ariba Solutions

You may use SAP's RPA technology, iRPA, to build RPA bots for SAP Ariba solutions.

#### ⓘ Note

Although various RPA technologies are possible, SAP can provide technical support only for SAP iRPA. If you opt to use another RPA technology, you do so at your own risk. You will not be able to report defects or seek

support from SAP Ariba Support; instead seek support from the makers of the RPA technology you have chosen.

Keep the following information in mind if you choose to use SAP iRPA bot technology in SAP Ariba solutions:

- When using SAP iRPA bots, SAP Ariba user interface screens should be captured at the recommended 100% resolution.
- Bots using SAP iRPA technology have successfully performed test operations only in SAP Ariba Buying and Invoicing.
- Training for iRPA technology is performed by the SAP iRPA team, not by SAP Ariba.
- Supported browsers are Internet Explorer and Google Chrome.

Supported functions include:

- Entering information in form fields.
- Submitting forms by clicking on buttons.
- Clicking on links.
- Calling SAP Ariba APIs from within SAP iRPA.

## Defect Reporting and Support

- Defects should be reported to SAP iRPA, not to SAP Ariba. In case of technical issues with iRPA software, report an incident on component **CA-ML-IPA**.
- In your defect report, include a full description of the defect.
- For more information about using SAP iRPA bot technology, see [the SAP iRPA product documentation](#) and [the SAP iRPA community page](#).

# SAP Ariba Developer Portal Quick Start Guide for Developers

The topics in this section provides a general workflow to show how these functions fit together, and how to create applications that use the APIs on the SAP Ariba developer portal to extend the functionality of your solutions.

[Steps to Start Using the SAP Ariba APIs \[page 26\]](#)

[Progress of Your Application Tracker \[page 28\]](#)

## Steps to Start Using the SAP Ariba APIs

Once your organization is registered to use the SAP Ariba APIs and your **Organization Admin** user has set up **Developer** user accounts, follow these steps to create applications that extend the functionality of SAP Ariba solutions.

1. Choose one or more APIs to use in your application. See [How to browse APIs on the developer portal \[page 16\]](#).
2. Create an application. This generates an **Application key** that identifies your application within the system. Every API request your application makes must include this key as the value of the `apiKey` parameter. See [Creating an Application on the SAP Ariba Developer Portal \[page 27\]](#)

#### Note

SAP Ariba limits customers to one application per realm/api combination. Duplicate applications require additional resources from SAP Ariba and are seldom permitted. If you require a duplicate application, you must contact your SAP Ariba account representative and provide a compelling business case explaining why the duplicate application is necessary. Your request will be reviewed and approved or denied based on cloud resource impact.

3. Ask your organization admin to request API access for your application by displaying the application in **My applications** and clicking **Actions > Ask your admin to request API access**. See "Requesting API access" in [Progress of Your Application Tracker \[page 28\]](#) for details.
4. A user with the **Organization Admin** role requests approval for API access. See [Requesting API Access for an Application on the SAP Ariba Developer Portal \[page 13\]](#).
5. SAP Ariba assesses the request, and once processed and approved, the **Organization Admin** user receives email with an OAuth client ID for the application.
6. A user with the **Organization Admin** role generates the OAuth secret and base64-encoded client and secret. See [Generating the OAuth Secret and Base64 Encoded Client ID and Secret \[page 33\]](#)
7. A user requests OAuth access tokens for the application. See [Requesting and receiving OAuth access tokens \[page 34\]](#).
8. A user with the **Developer** role codes a client application, presenting the application key and OAuth credentials with each web service call made. See [How to make REST API calls with the OAuth access token and application key \[page 33\]](#).

## Creating an Application on the SAP Ariba Developer Portal

Use this procedure to create an application that generates the API key that must be passed with all API requests.

### Context

In order to use any API on the SAP Ariba developer portal, you must create an application. To begin the development process, create an application by following these steps:

### Procedure

1. Log in to the SAP Ariba developer portal.
2. Do one of the following:
  - Click **Create application** from the home page.

- If you have the **Developer** role, click **Manage** in the left-hand navigation area, then click the + symbol near the search bar.
  - If you have the **Organization admin** role, navigate to ► **Manage** ► **Applications** ▾, then click the + symbol near the search bar.
3. Fill out the **Create a new application** form by entering an application name and description, then click **Submit**.

#### ⓘ Note

To exit the form without creating an application, click **Cancel**.

## Results

This creates a data object to represent your application in the system, and generates an **Application key** that identifies your application within the system. Every API request your application makes must include this key as the value of the `apiKey` parameter to identify it as part of a registered application. The new application will appear in the list of applications on the left. For **Developer** users, this list is labeled **My Applications**; for **Organization admin** users, this list is labeled **Applications**.

## Tracking Your API Usage to Avoid Exceeding Rate Limits

Use this procedure to track your API usage. Your API usage is listed in the output headers.

You can track your current API usage by looking at the following headers in the JSON response to your API queries:

<b>X-RateLimit-Limit-day</b>	Indicates how many queries are permitted per day
<b>X-RateLimit-Remaining-day</b>	Indicates how many queries you have left in the current day
<b>X-RateLimit-Limit-hour</b>	Indicates how many queries are permitted per hour
<b>X-RateLimit-Remaining-hour</b>	Indicates how many queries you have left in the current hour
<b>X-RateLimit-Limit-minute</b>	Indicates how many queries are permitted per minute
<b>X-RateLimit-Remaining-minute</b>	Indicates how many queries you have left in the current minute
<b>X-RateLimit-Limit-second</b>	Indicates how many queries are permitted per second
<b>X-RateLimit-Remaining-second</b>	Indicates how many queries you have left in the current second

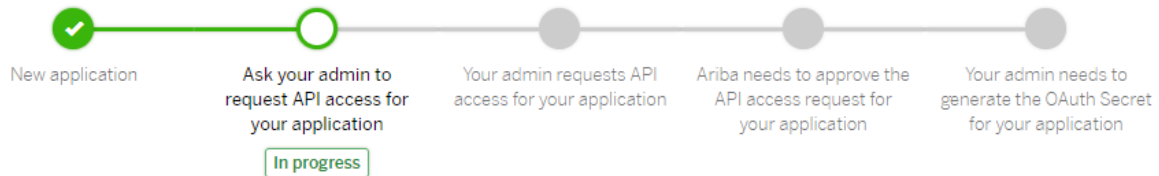
## Progress of Your Application Tracker

This topic describes how to use the **What's Next** progress tracker to organize the creation of your API application in the SAP Ariba developer portal.

When you create a new application on the SAP Ariba developer portal, its application page displays a graphical progress tracker in the **What's Next** section. This tracker

shows the steps required to create and publish an application using SAP Ariba APIs.

## What's Next?



## How to Interpret the Progress Tracker

Each node on the tracker represents a step in the process. The label on each node provides a brief description of the represented step, and you can hover the pointer over a node for a more detailed explanation of the step itself and how to execute it.

When a step is completed, the node representing that step changes color, and its hover text changes to reflect completion, allowing you to track your progress. Completed steps are green and contain a check mark. The next step in the process is a green circle with no check mark. Subsequent incomplete steps are gray circles with no check mark.

## Progress Tracker Accessibility Features

You can manipulate the **What's Next** graphical progress tracker on the SAP Ariba developer portal using the mouse or keyboard controls.

Hovering the mouse over a node on the **What's Next** graphical progress tracker opens a dialog. This dialog can be dismissed by clicking the **x** in its upper-right corner with the mouse, and you can click **Show more** to view more information. You can also use keyboard controls, as follows:

- Use the **TAB** key to put focus on the **Show more** in the tooltip or **x** in the dialog that appears when you choose **Show more**.
- Use the **ENTER** or **RETURN** key with focus on **Show more** to view a dialog containing more information.
- Use the **ENTER** or **RETURN** key with focus on the **x** to close the dialog.
- Use the **ESC** key to close the dialog.

## Requesting API Access

Some of the steps in the process refer to the need for API access. API access enables your application to use a specific API in the test or production realm. The developer can trigger the request for API access right in the SAP Ariba developer portal UI by following these steps:

1. On the **Manage Applications** tab, open the page for your application.
2. From the **Actions** menu, choose **Ask your admin to request API access**.
3. Fill out the **Application Details** dialog as follows:
  - From the **API Names** dropdown, choose the name of the API you wish to use in your application.
  - From the **Realm Name** dialog, choose your realm name.
  - Optional. If the **AN-ID** field does not populate automatically based on your **Realm Name** selection, enter your ANID.
  - In the **Realm Type** section, choose the radio button for either **Test** or **Production**.
  - Optional. If you want to submit a duplicate combination of an auto-approved API and one or more realms, enter a reason in the text box.
4. Click **Submit**. This alerts the organization admin of the need to request API access for the current application.

## SAP Ariba Developer Portal Authentication

This chapter describes how to authenticate your API calls using OAuth.

[Use of the API Gateway and OAuth to Authenticate Applications \[page 30\]](#)

[Finding Your Application's Application Key and OAuth Client ID \[page 31\]](#)

[Managing Security Certificates for Mutual Authentication \[page 31\]](#)

[Generating the OAuth Secret and Base64 Encoded Client ID and Secret \[page 33\]](#)

[OAuth Access Token Request and Receipt \[page 34\]](#)

[Making of REST API Calls with the OAuth Access Token and Application Key \[page 38\]](#)

[SAP Ariba Developer Portal Authentication Credentials Saving and Safeguarding \[page 39\]](#)

## Use of the API Gateway and OAuth to Authenticate Applications

The APIs on the SAP Ariba developer portal are protected by the API Gateway and OAuth authentication. API Gateway authentication is based on the application key (`apikey`) of your application. Only valid application keys enable requests to be accepted by the gateway. We further support the two-legged OAuth protocol or Client Credentials authorization flow in which a registered client application requests and receives an access token from the OAuth authentication server. All API requests need to have both a valid application key and a valid OAuth token unique to the client application making the request.

Your application must make REST API requests to access to protected resources as follows:

- The registered client makes a one time request to send an authorization request to the OAuth server, and receives an access token and a refresh token in return.
- Include the request header `apikey` with value of the application key in the headers of each API call your application makes.
- The registered client includes the access token in all requests for access to protected resources from the resource server.

- When the access token expires, the client should present the refresh token to request a new access token and updated refresh token, and then use the new access token to request protected resources.
- Optional. You can configure client applications on the SAP Ariba developer portal to require mutual authentication before an OAuth token can be generated or refreshed. For details, see [Managing Security Certificates for Mutual Authentication \[page 31\]](#).

## Finding Your Application's Application Key and OAuth Client ID

Use this procedure to find your application's application key and OAuth client ID.

### Context

To authenticate to the SAP Ariba APIs, you will need your application's application key. The application key was generated when the application was first created. It is used as the value of `apikey` during REST API calls.

To execute OAuth authentication, you need your application's OAuth client ID. This ID was generated when the application was approved for production.

To find your application's application key and OAuth client ID, follow these steps:

### Procedure

1. Log in to the SAP Ariba developer portal.
2. Click **Manage** and select an application from the list.
3. The application key for your application is the value in the **Application key** field. The OAuth client ID is the value in the **OAuth client ID** field.

## Managing Security Certificates for Mutual Authentication

Use this procedure to store certificates in the repository for mutual authentication.

### Prerequisites

You must have X.509 certificates to authenticate your applications in Privacy Enhanced Mail (PEM) format. The certificates must be issued by a Certificate Authority (CA) trusted by SAP Ariba. To get information about the CAs currently trusted by SAP Ariba or to register a public CA to be trusted by SAP Ariba, contact SAP Ariba Support.

## Context

To configure an application to require mutual TLS authentication during the process of generating an OAuth token, upload one or more security certificates, as follows:

## Procedure

1. Log in to the SAP Ariba developer portal as an **Organization admin** user, then click **Manage**.
2. On the **Applications** tab, select the application for which you wish to require mutual TLS authentication.
3. From the **Actions** menu, choose **Edit Mutual TLS Authentication**.  
The **Mutual TLS Authentication** dialog appears.
4. Add certificates to the repository. For each desired certificate, follow these steps:
  - a. Click **Add a new certificate**  
The **Add a new certificate** dialog appears.
  - b. On the **Add a new certificate** dialog, do one of the following:
    - Copy and paste the desired certificate text into the **Add a certificate** text box.
    - Click **Upload certificate**, then browse to the desired certificate and open it. The certificate text appears in the **Add a certificate** text box.

### Note

The MTLS security certificate must be structured in the following order when adding a new certificate:

1. ServerCertificate
2. IntermediateCertificate
3. RootCertificate

When uploading the certificate, ensure that it is in a single plain text file and organized in the same order.

- c. Enter a unique, descriptive name for the certificate in the **Certificate name** field.
- d. Click **Add certificate**.  
Focus returns to the **Mutual TLS Authentication** dialog, and the new certificate appears in the **Manage certificates** table.

All certificates in the **Manage certificates** table are registered in the repository, and can be used to validate the certificates exchanged while generating an OAuth token.

5. Optional. To delete a certificate, click its **Delete** button in the **Manage certificates** table, then confirm in the the confirmation popup.
6. Enable MTLS by sliding the **Enable MTLS** slider into the right-hand position. By default this slider is in the left-hand position, indicating that mutual TLS authentication is not enabled.

### Note

You can turn MTLS off by sliding this slider back to the left-hand, off position - even if there are still certificates in the table.

7. When you are finished managing security certificates, click **Done**.



### Note

If you turn on Mutual TLS Authentication, the **Manage certificates** table must contain at least one certificate before you can proceed.

## Results

Mutual TLS authentication is enabled for this application, and all requests to generate or refresh an OAuth certificate for this application must authenticate using one of the certificates in the **Manage certificates** table.

### Note

To apply MTLS authentication, your OAuth queries must use the **MTLS OAuth URL Prefix** listed in the **Environment Details** section of the discovery page for the API your application will use, instead of using the usual **OAuth Server URL Prefix**.

## Generating the OAuth Secret and Base64 Encoded Client ID and Secret

If you do **not** have Mutual TLS authentication enabled for an application, you must generate an **OAuth secret** for your application and submit a **Base64 Encoded Client and Secret** (Base64-encoded **OAuth Client ID** and **OAuth Secret**) to the OAuth server to obtain an OAuth access token.

## Context

If Mutual TLS authentication is enabled for your application, skip this procedure. Instead, you must obtain and configure a security certificate for your application as described in [Managing Security Certificates for Mutual Authentication \[page 31\]](#).

### Note

- When you generate a new **OAuth Secret** for your application, the previous **OAuth Secret** and **Base64 Encoded Client and Secret** become invalid. You must retrieve a new access token and refresh token.
- **OAuth Secret** and **Base64 Encoded Client and Secret** can also be constructed by concatenating **OAuth Client ID** and **Client Secret** separated by a colon and encoding the result using Base 64 encoding with a tool such as <https://www.base64encode.org/>.

To generate the **OAuth Secret** and **Base64 Encoded Client and Secret**, follow these steps:

## Procedure

1. Log in to the SAP Ariba developer portal as a user with the **Organization Admin** role.
2. Click **Manage** in the left-hand navigation menu.
3. Select your application from the list of applications.
4. Choose **Actions > Generate OAuth Secret**.
5. Click **Submit**. The **OAuth Secret** and **Base64 Encoded Client and Secret** are displayed temporarily.
6. Copy the **OAuth Secret** and **Base64 Encoded Client and Secret** and save externally at a secured location.

## OAuth Access Token Request and Receipt

To gain access to protected resources, your registered application must present an access token to the OAuth server associated with your regional data center. See the SAP Ariba developer portal for the exact URL.

### Note

The topics in this section provide examples in CURL format. On Windows, you can

- either install CURL Command Line <https://curl.haxx.se/download.html>
- or copy the command line and import it into Postman <https://getpostman.com>

[Initial Access Token Requests \[page 34\]](#)

[Expired Access Token Refresh Requests \[page 36\]](#)

## Initial Access Token Requests

Request an access token by using HTTPS to post your request. If you do not have Mutual TLS (mTLS) authentication enabled, include the **OAuth Client ID** and **Client Secret** as HTTP Basic Authorization credentials. If you have mTLS enabled, include the **OAuth Client ID** in the request.

### Initial access token request Without mTLS

Use the following CURL example as a model when constructing the initial HTTPS request your application sends to the OAuth server for the initial access token:

```
curl -X POST
  {{oauth_server_url_prefix}}/v2/oauth/token \
  -H 'Authorization: Basic Base64_Encoded_Client_And_Secret' \
  -H 'Content-Type:application/x-www-form-urlencoded' \
```

```
-d 'grant_type=client_credentials'
```

Where:

*oauth\_server\_url\_prefix* is the URL prefix for the OAuth server for your region, as shown on the SAP Ariba developer portal on the discovery page for any API, in the **Environment details** table.

*Base64\_Encoded\_Client\_And\_Secret* is the Base64-encoded **OAuth Client ID** and **OAuth Secret** generated as described in [Generating the OAuth Secret and Base64 Encoded Client ID and Secret \[page 33\]](#).

#### Note

Legacy users may choose to use 'grant\_type=openapi\_2lo'

## Initial access token request with mTLS

Use the following CURL example as a model when constructing the initial HTTPS mTLS request your application sends to the OAuth server for the initial access token. The request specifies the **OAuth Client ID**. The request also includes the X.509 certificate file for the client and the corresponding private key file:

```
curl --request POST 'oauth_server_url_prefix/v2/oauth/token?
grant_type=openapi_2lo'\
--header 'client-id: clientId' \
--cert "myCertFile" --key "myPrivateKeyFile"
```

Where:

*oauth\_server\_url\_prefix* is the URL prefix for the OAuth server for your region, as shown on the SAP Ariba developer portal on the discovery page for any API, in the **Environment details** table.

*clientId* is the OAuth Client ID for the application. You can obtain the ID as described in [Finding Your Application's Application Key and OAuth Client ID \[page 31\]](#).

*myCertFile* is the name of the X.509 certificate file configured to authenticate your client as described in [Managing Security Certificates for Mutual Authentication \[page 31\]](#).

*myPrivateKeyFile* is the name of the private key file that corresponds to the X.509 certificate.

#### Note

Legacy users may choose to use 'grant\_type=openapi\_2lo'

## Sample Response

```
{
  "timeUpdated": 1462815524141,
  "access_token": "5b685b82-7f5a-42eb-b4a3-027004d317f5",
  "refresh_token": "6d6b2b9d-8264-46fd-9909-c870215d9b21",
  "token_type": "bearer",
  "expires_in": 1440
}
```

## Response Parameters

<code>timeUpdated</code>	the time when the access token was created
<code>access_token</code>	the token to be included in each request for access to protected resources
<code>refresh_token</code>	the token to be included in a request for a new access token when your current access token has expired
<code>token_type</code>	always bearer
<code>expires_in</code>	the duration in seconds before the token expires. The default is 1440 seconds, or 24 minutes

## Related Information

[Finding Your Application's Application Key and OAuth Client ID \[page 31\]](#)

## Expired Access Token Refresh Requests

Your access token expires after a number of seconds specified in the response element. The default lifespan for an access token is 1440 seconds, a total of 24 minutes.

If you get a 401 response code for a REST API call, you must acquire a new access token. To request a new access token, make a request to the OAuth server using the refresh token you received with the original access token.

You can refresh an access token either after it has expired, or no earlier than two minutes before it expires.

## Using a Refresh Token Without mTLS

If you do not have Mutual TLS (mTLS) authentication enabled, use the following `CURL` example as a model to request a new access token using a refresh token received with a previous access token.

```
curl -X POST
  {{oauth_server_url_prefix}}/v2/oauth/token \
  -H 'Authorization: BasicBase64_Encoded_Client_And_Secret' \
  -H 'Content-type:application/x-www-form-urlencoded' \
  -d 'grant_type=refresh_token&refresh_token=refresh_token'
```

Where:

`oauth_server_url_prefix` is the URL prefix for the OAuth server for your region, as shown on the SAP Ariba developer portal on the discovery page for any API, in the **Environment details** table.

*Base64\_Encoded\_Client\_And\_Secret* is the Base64-encoded **OAuth Client ID** and **OAuth Secret** generated as described in [Generating the OAuth Secret and Base64 Encoded Client ID and Secret \[page 33\]](#).

*refresh-token* is the refresh token you received with the access token you wish to refresh

## Using a Refresh Token with mTLS

If you have mTLS authentication enabled, use the following `CURL` example as a model to request a new access token using a refresh token received with a previous access token.

```
curl --request POST 'oauth_server_url_prefix/v2/oauth/token?
grant_type=openapi_2lo'\
--header 'client-id: clientId' \
--cert "myCertFile" --key "myPrivateKeyFile"
--form 'grant_type="openapi_2lo"' \
--form 'refresh_token="refreshToken"'
```

Where:

*oauth\_server\_url\_prefix* is the URL prefix for the OAuth server for your region, as shown on the SAP Ariba developer portal on the discovery page for any API, in the **Environment details** table.

*clientId* is the OAuth Client ID for the application. You can obtain the ID as described in [Finding Your Application's Application Key and OAuth Client ID \[page 31\]](#).

*myCertFile* is the name of the X.509 certificate file configured to authenticate your client as described in [Managing Security Certificates for Mutual Authentication \[page 31\]](#).

*myKeyFile* is the name of the private key file that corresponds to the X.509 certificate.

*refresh-token* is the refresh token you received with the access token you wish to refresh

## Sample Response

```
{
  "timeUpdated":1462818063261,
  "access_token":"f3e21aaf-218d-48b8-9195-b77bd88c8b82",
  "refresh_token":"da420531-ca3e-4eb0-9c2f-4f6584d1b91f",
  "token_type":"bearer",
  "expires_in":1440
}
```

Include the new access token in subsequent requests for protected resources. When this new access token expires, you can use the new refresh token to refresh it.

Future token refresh requests should use the newest refresh token. The old refresh token will be invalid.

## Response Parameters

<code>timeUpdated</code>	the time when the access token was created
<code>access_token</code>	the token to be included in each request for access to protected resources
<code>refresh_token</code>	the token to be included in a request for a new access token when your current access token has expired
<code>token_type</code>	always bearer
<code>expires_in</code>	the duration in seconds before the token expires. The default is 1440 seconds, or 24 minutes

## Making of REST API Calls with the OAuth Access Token and Application Key

Each request for protected resources must include a valid access token. This section provides information about how to request access to protected resources by including your access token in a request to the resource server.

### Note

This topic provides examples in CURL format. On Windows, you can

- either install CURL Command Line <https://curl.haxx.se/download.html>
- or copy the command line and import it into Postman <https://getpostman.com>

### Note

In the sample URLs in this topic, replace `{{runtime_url}}` with the desired runtime URL from the **Environment Details** table on the SAP Ariba developer portal discovery page for this API.

## Sample Request

Use the following CURL example as a model when constructing the request your application will send to the resource server for access to protected resources:

```
curl -X GET
  '{{runtime_url}}/{resource}?{service_query_parameter1=value1}
[...&{service_query_paramN=value}]' \
-H 'accept: application/json' \
-H 'apiKey: <application key>' \
-H 'Authorization: Bearer <access_token>'
```

## Constructing the Request URL

Construct the request URL by joining the API's public URL (found in the **Environment details** section of the API's discovery page), the resource, and any query parameters for the desired API, as follows: `{{runtime_url}}/{resource}?{parameters}`

For example, in the US data center, the URL for a GET request seeking a list of requisitions whose state has changed might look like this

```
curl -X GET
'https://openapi.ariba.com/api/approval/v1/prod/changes?
realm=myRealm&limit=5&offset=0&needTotal=false'
-H 'accept: application/json'
-H 'apiKey: <api_key>'
-H 'Authorization: Bearer <access_token>'
```

### Note

If the value of a parameter in the query URL includes a slash character (`/`), that slash character must be double URL encoded by replacing it with `%252f`. For example, if the supplier ID is `UmbrellaCorp/SmallCompany234`, in the URL this would be `UmbrellaCorp%252fSmallCompany234`.

## Response

The JSON response includes the data requested by your client application.

If there is something wrong with your request, you may receive one of the following error codes:

<b>401 Unauthorized - Token is expired</b>	The Authorization header bearer token has expired. Follow the steps in <a href="#">Expired Access Token Refresh Requests [page 36]</a>
<b>401 Unauthorized - This token is not authorized to access this API</b>	The API is not enabled for the OAuth client ID of the application. API-specific enablement steps must be configured first. See documentation for the specific API for details.
<b>401 Unauthorized - No API key found in request</b>	The message header is missing the apiKey value
<b>403 Forbidden - Invalid authentication credentials</b>	The message header has an invalid value for apiKey

## SAP Ariba Developer Portal Authentication Credentials Saving and Safeguarding

- Save and Store the OAuth Client ID, OAuth Client Secret, Base64 Encoded Client and Secret, and the Shared Secret.
- Do NOT store your OAuth Client Secret and/or Base64 Encoded Client and Secret word, token, or key in a database.

- Do NOT store or send your OAuth Client Secret and/or Base64 Encoded Client and Secret word, token, or key in an email.
- Do NOT store your OAuth Client Secret and/or Base64 Encoded Client and Secret word, token, or key in a code base that may use version control.
- Do NOT store your OAuth Client Secret and/or Base64 Encoded Client and Secret word, token, or key in a text file stored locally.
- Do NOT store your OAuth Client Secret and/or Base64 Encoded Client and Secret word, token, or key in session storage to be used as an authentication method.
- - OAuth2 Shared Secret access tokens should be securely saved and stored externally.
  - Use a commercial, web based, password manager application to tightly control access tokens and encrypt, store, share, and control access. Some SSO (Single Sign On) systems may have key or secret word secure storage features. Privileged accounts provide access to an organization's most sensitive data and critical systems, in addition to keys or secret words needing protection and control over who can access the keys that need to be secured.
- Regulatory Compliance for your organization may require strong OAuth token and key security storage.

## API-Specific Disclaimers and Legal Information

The SAP Ariba developer portal included in the SAP Ariba APIs product and the APIs made available on this site are provided solely at the discretion of SAP without warranty of any kind, and SAP may change, suspend, or cancel any or all features or functions of the SAP Ariba APIs product or revise the website at any time. Any production use of or commercialization of applications containing any APIs provided on this website is prohibited without a written agreement between your company and SAP governing such activities.

Access to this API is available to you as a subscriber to this solution as part of the SAP Cloud Service Level Agreement. However, it is not considered part of the solution. Use of this API is purely optional and is subject to restrictions stated in the documentation, including the Terms of Use and the documentation found at the SAP Ariba developer portal (see <https://developer.ariba.com/api>). If you wish to connect a third-party service using this API, first confirm that the company is participating in the SAP partner program and is authorized to provide a connection to this solution using this API. You will be required to submit written consent to SAP to authorize the exchange of data with the third-party service.





# Important Disclaimers and Legal Information

## Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

## Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

## Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

## Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

## Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

Copyright © 2024 Ariba, Inc. All rights reserved.

This documentation, as well as the Ariba solutions, software and/or services described in it, contain proprietary information. They are provided under a license or other agreement containing restrictions on use and disclosure and are also protected by copyright, patent and/or other intellectual property laws. Except as permitted by such agreement, no part of the document may be reproduced or transmitted in any form by any means, electronic, mechanical or otherwise, without the prior written permission of Ariba, Inc.

Ariba, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in the documentation. The information contained in the documentation is subject to change without notice.

Ariba and Ariba products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Ariba, Inc. in the United States and other countries. Please see <http://www.ariba.com/legal/trademarks> for additional trademark information and notices.

Ariba Sourcing solutions (On Demand and software) are protected by one or more of the following patents, including without limitation: U.S. Patent Nos. 6,199,050; 6,216,114; 6,223,167; 6,230,146; 6,230,147; 6,285,989; 6,408,283; 6,499,018; 6,564,192; 6,871,191; 6,952,682; 7,010,511; 7,072,061; 7,130,815; 7,146,331; 7,152,043; 7,225,152; 7,277,878; 7,249,085; 7,283,979; 7,283,980; 7,296,001; 7,346,574; 7,383,206; 7,395,238; 7,401,035; 7,407,035; 7,444,299; 7,483,852; 7,499,876; 7,536,362; 7,558,746; 7,558,752; 7,571,137; 7,599,878; 7,634,439; 7,657,461; 7,693,747; 8,364,577; and 8,392,317. Patents pending.

Other Ariba product solutions are protected by one or more of the following patents:

U.S. Patent Nos. 6,199,050, 6,216,114, 6,223,167, 6,230,146, 6,230,147, 6,285,989, 6,408,283, 6,499,018, 6,564,192, 6,584,451, 6,606,603, 6,714,939, 6,871,191, 6,952,682, 7,010,511, 7,047,318, 7,072,061, 7,084,998, 7,117,165; 7,225,145; 7,324,936; 7,536,362; 8,364,577; and 8,392,317. Patents pending.

Certain Ariba products may include third party software or other intellectual property licensed from a third party. For information regarding software or other intellectual property licensed from a third party, go to <http://www.ariba.com/copyrights.cfm>.