CONFIDENTIAL

2018-12

The SAP Ariba developer portal

SAP Ariba developer portal

THE BEST RUN SAP

Content

The SAP Ariba developer portal

The developer portal gives your organization and its developers access to solution-extending APIs that can make your business more efficient and effective.

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 extentions 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 developer portal: 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 developer portal: Developer portal quick start guide for developers [page 9]
- 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: Developer portal authentication [page 11]

General prerequisites

All use of the developer portal requires the following prerequisites.

- Your organization must have a current license for one or more SAP Ariba solutions or an Ariba Network solution component to use the APIs. Example solutions include SAP Ariba Buying, SAP Ariba Discovery, SAP Ariba Invoice Management, SAP Ariba Payables, and others.
- Your organization must be in the United States of America or another supported country.
- 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.
- 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

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

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.

In this section:

How to register your organization to use the developer portal

Procedure

1. At https://developer.ariba.com/api, and choose your region from the **Portals** dropdown.
2. Choose **Request an account** and fill out the form.
3. You will receive email in response to the form. Follow the instructions in the email to receive login credentials for a user with the **Organization Admin** role.
4. At https://developer.ariba.com/api, choose your region from the **Portals** menu, and log in using the credentials you received.
5. Agree to the Terms of Service for your region.

Administrator functions related to developer portal user accounts

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

- **Create new user accounts with the Developer role**, by clicking the **+** sign to the right of the search window. Fill out the form as follows:
  - Enter the user's name as you want it to appear.
  - Enter the user's email address. This email address serves as the user's login name.
  - Enter a temporary password for the user.

  When you send the login credentials to the user, remind them to change their password upon initial login.

- **Browse your current user accounts** by clicking the desired user on the left.
- **Download the first name, last name, and email address of the displayed user as a CSV file** by clicking ▶ **Actions** ▶ **Download Personal Information** ▶.
- **Change the first name, last name, and email address of the displayed user** by clicking ▶ **Actions** ▶ **Edit User** ▶.
- **Delete the displayed user account** by clicking ▶ **Actions** ▶ **Delete** ▶.
- **Help the displayed user set up a new password** by choosing ▶ **Actions** ▶ **Reset user password** ▶. This action sends a standard password-reset email to the user, including instructions.
- **View and set the displayed user's role** using the toggles in the **Roles** section. You can assign a user the role of **Organization Admin** or **Developer**. While **Organization Admin** users can manage all accounts and applications associated with your organization, **Developer** users can manage only those applications that are assigned to them.
- Generate and download a report in CSV format, containing all actions taken by the displayed user since the user was created. To generate and dowload this report, click ▶ **Actions** ▶ **Download audit log** ▶.

Administrator functions related to applications on the developer portal

To access **Organization Admin** functions related to applications, log in to the 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:

- In order to use any API on the SAP Ariba developer portal, you need to create an application. Begin the development process by clicking **Create application** from the home page or clicking the **+** symbol near the search bar in **My applications**. 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 to identify it as part of a registered application. The new application will appear in the **My applications** list.

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

- Click on an application 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. All web service calls made by this application must include its API Key.
- Delete the displayed application by choosing ▶ **Actions** ▶ **Delete application** ▶.
- Reassign the displayed application to a different developer within your organization by choosing ▶ **Actions** ▶ **Assign this application to another developer** ▶.
- Begin the process of publishing the displayed application by choosing ▶ **Actions** ▶ **Request production access** ▶ and filling out the popup form as follows:
  - In the **API Names** field, enter the names of the APIs you wish to use. Use the titles on the **Discover** tab. For example, **Custom Forms API** or **Flow Extension API**.
  - From the **Ariba cloud application** dropdown, choose the application you with to extend using APIs.
  - In the **Realm name** field, enter the name of your customer realm.

    - Optional. In the **AN ID** field, enter your Ariba Network ID.
    - To register your application for use in the test realm, click the **Test** radio button in the **Realm type** section. To register yoru application in the production realm, click **Production** instead.
    - Optional. If you have additional comments, you an type them in the **Additional comments** box.
    - When you are ready, click **Submit** to begin the process of requesting production access for your application. To cancel, click **Cancel**.

  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.
- 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 production access has been granted and approved. See How to generate the OAuth Secret and Base64 Encoded Client and Secret [page 13] for details.

How to browse APIs on the 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:

   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 on the developer portal

When a new version of an API is released, a single developer portal discovery page hosts both versions.

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`

Managing your own APIs on the 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 API, either by clicking **Get Started** for your first API or by clicking the plus sign (near the **Search** bar on the left) for subsequent APIs. Fill out the form as follows:
  - Required. Enter a name for the API in the **API name** field.
  - Required. Enter the target URL in the **Target URL** field. This is the URL for the API you want to publish, such as `https://www.someapipurveyor.com/api/v2`. This URL must provide a complete path to the API. It may not include parameters. Parameters should be entered in the **URL query parameters** section or added at runtime. An internet protocol such as `https` is required.
  - Required. Enter a brief description in the **Description** field.
  - Required. In the **Whitelisted Tenants** dropdown, select the realms from which you wish this API to be accessible.
  - 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 **Tags** field and then hit the `Space` bar. To remove a tag, click the **x** in the tag you want to remove.

    > ℹ Note
    >
    > Tags may not contain spaces.

  - Optional. You can enter static headers and values as desired in the **Header parameters** table. To add additional rows, click **Add a row** 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.
  - Optional. You can enter static URL query parameters and values in the **URL query parameters** table. To add additional rows, click **Add a row** 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.

- The **Authentication type** dropdown allows you to instruct SAP Ariba to perform authentication tasks. Possible values are **None**, **Basic authentication**, and **OAuth 2.0**. Default is **None**.

  None Choose this if you do not needSAP Ariba to perform authentication tasks.

  Basic authentication Choose this if your API requires HTTP basic authentication. Enter username and password in the form.

  OAuth 2.0. 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:

    - **Token endpoint** (required): Enter the endpoint used to retrieve the access token for your API.
    - **Client Id** (required): Enter client ID for your API. This ID should be provided by the administrator of the external API.
    - **Client secret** (required): Enter the client secret for your API. This secret should be provided by the administrator of the external API.
    - **Grant type** (required): Enter the grant type for your API.
    - **HTTP Request Method**: Choose **POST** or **GET** from the dropdown. Default is **POST**.
    - **Body Content Type**: Choose **FORM_URL_ENCODED** or **JSON** from the dropdown. Default is **FORM_URL_ENCODED**.
    - **OAuth header parameters** (optional): Header parameters and their assigned values entered in this table will be sent with the access token.
    - **OAuth URL query parameters** (optional): URL query parameters and their assigned values entered in this table will be sent with the access token.
    - **OAuth form parameters** (optional): Form parameters and their assigned values entered in this table will be sent with the access token.

  > i Note
  >
  > If your API does not conform to the OAuth 2.0 standard, this authentication option will not succeed.

- Required. Upload a swagger file for the API by clicking **Browse file...**. This file 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 your custom form setup.

- 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.
- Filter your current external APIs using their assigned tags. To find all of your APIs that have been assigned a particular tab, type the desired tag into the **Search** field, then hit `Enter` 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 `Enter` 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.

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 developer portal to extend the functionality of your solutions.

In this section:

Steps to start using the APIs

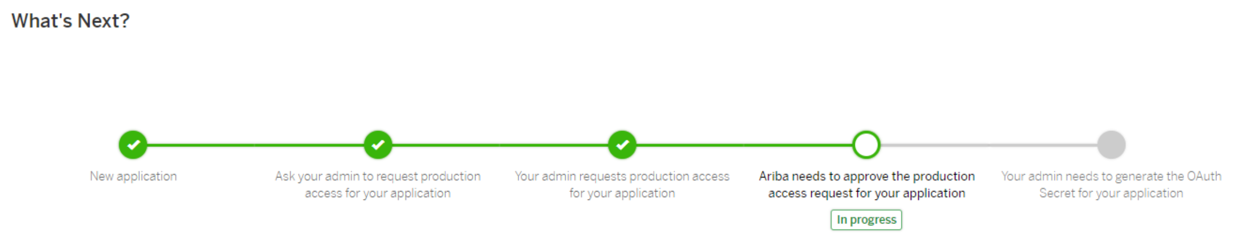
Once your organization is registered to use the SAP Ariba APIs and your **Organization Admin** user has set up **Developer** user accounts, follow these steps to create applications that extend the functionality of SAP Ariba solutions.

1. Choose one or more APIs to use in your application. See How to browse APIs on the developer portal [page 6].
2. In order to use any of the APIs on the developer portal, you need to create an application. Begin the development process by clicking **Create application** from the home page or clicking the **+** symbol near the search bar in **My applications**. This generates an **Application key** that identifies your application within the system. Every API request your application makes must include this key. See Administrator functions related to developer portal user accounts [page 4]
3. Ask your organization admin to request production access by displaying the application in **My applications** and clicking ▶ **Actions** ▶ **Ask your admin to request production access** ◀. See "Requesting production access" in Tracking the progress of your application [page 10] for details.
4. A user with the **Organization Admin** role requests approval for production access by displaying the application in **My applications** and clicking ▶ **Actions** ▶ **Request production access** ◀. See Administrator functions related to applications on the developer portal [page 5].
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 How to generate the OAuth Secret and Base64 Encoded Client and secret [page 13]
7. A user requests OAuth access tokens for the application. SeeRequesting and receiving OAuth access tokens [page 13].
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 13].

Tracking the progress of your application

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

When you create a new application on the 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?
New application
Ask your admin to request production access for your application
Your admin requests production access for your application
Ariba needs to approve the production access request for your application
In progress
Your admin needs to generate the OAuth Secret for your application

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.

Requesting production access

Some of the steps in the process refer to the need for production access. Production access enables your application to use a specific API in the test or production realm. The developer can trigger the request for production access right in the 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 production access**. This alerts the organization admin of the need to request production access for the current application.

Developer portal authentication

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

In this section:

Using the API Gateway and OAuth to authenticate applications

The APIs on the developer portal are protected by the API Gateway and OAuth authentication. API Gateway authenication 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 needs 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.

How 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.
3. Click **Manage** and select an application from the list.
4. 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.

How to generate the OAuth Secret and Base64 Encoded Client and secret

To execute OAuth authentication, you will need the OAuth Client ID and OAuth secret for your application.

Context

> i 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 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.

Requesting and receiving OAuth access tokens

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.

> i 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

In this section:

How to refresh an expired access token [page 15]

How to request the initial access token

Request an access token by sending your **OAuth Client ID** and **Client Secret** via HTTP Basic Authentication, using an `HTTP POST` request.

Use the following `CURL` example as a model when constructing the initial request your application will send 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=openapi_2lo'
```

> **i Note**
>
> You can find the value of `oauth_server_url_prefix` for your region on the SAP Ariba developer portal on the discovery page for any API, in the **Environment details** table.

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

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

How to refresh an expired access token

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 need to refresh the access token.

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

To request a new access token, make a request to the OAuth server using the refresh token you received with the access token you wish to refresh.

Sample request

Use the following `CURL` example as a model when constructing the request your application will make to the OAuth server to refresh an 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=refresh_token&refresh_token=<refresh_token>'
```

> i Note
>
> You can find the value of `oauth_server_url_prefix` for your region on the SAP Ariba developer portal on the discovery page for any API, in the **Environment details** table.

Query parameters

`refresh-token`	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

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

How to make 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.

> i 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

> i 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>'
```

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:

`401 Unauthorized - Token is expired`	The `Authorization` header `bearer` token has expired. Follow the steps in How to refresh an expired access token [page 15]
`401 Unauthorized - This token is not authorized to access this API`	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.
`401 Unauthorized - No API key found in request`	The message header is missing the `apikey` value
`403 Forbidden - Invalid authentication credentials`	The message header has an invalid value for `apikey`

Saving and safeguarding developer portal authentication credentials

- 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.
- SAP Single Sign-On (SAP SSO) includes Password Manager which will manage the use and security of your SAP Open APIs OAuth security tokens and secret words.

Revision history

The following table provides a brief history of the updates to this guide. SAP Ariba updates the technical documentation for its cloud solutions if:

- Software changes delivered in service packs or hot fixes require a documentation update to correctly reflect the new or changed functionality.
- The existing content is incorrect or user feedback indicated that important content is missing.

SAP Ariba reserves the right to update its technical documentation without prior notification. Most documentation updates will be made available in the same week as the software service packs are released, but critical documentation updates may be released at any time.

Month/Year of update	Updated topic	Short description of change
November	Managing your own APIs on the developer portal	Added filtering external APIs by tags.
	API versioning on the developer portal	Added topic.
August	Steps to start using the APIs	Added step for developer to ask admin to request production access
	Tracking the progress of your application	Added topic
	Managing your own APIs on the developer portal	Added topic
April	Administrator functions related to developer portal user accounts	Added ability to download user activity report.

Month/Year of update	Updated topic	Short description of change
March 2018	• The SAP Ariba developer portal • API application development workflow • Administrator functions related to applications on the developer portal	Emphasized that to use any of the APIs you must create an application.
	Administrator functions related to applications on the developer portal	Detailed exactly how to fill out the form when requesting production access for an application.
	API application development workflow	Changed title of topic to Steps to start using SAP Ariba APIs.
	Administrator functions related to developer portal user accounts	Added new functions: edit user name and email; download user name and email as CSV file.
March 2018	• Requesting and receiving OAuth access tokens • How to request the initial access token • How to refresh an expired access token • How to make REST API calls with the OAuth access token and application key	Updated CURL examples.
February 2018	How to register your organization to use the developer portal	Registration process is location-specific.
	• The SAP Ariba developer portal	Updated supported browser list
	Developer portal quick start guide for developers	Removed *API application development walk-through* and moved section before *Developer portal authentication*
January 2018	n/a	Initial publication.

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 web site at any time. Any production use of or commercialization of applications containing any APIs provided on this web site 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 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 a 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.

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.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

www.ariba.com

THE BEST RUN SAP