



PUBLIC

Document Version: 2608 – 2026-08

Surrogate Bid API

SAP Ariba Sourcing

SAP Ariba APIs

SAP Ariba developer portal



Content

Surrogate Bid API.	3
Surrogate Bid API Endpoint.	3
Workflow for the Surrogate Bid API.	4
Developing an Application to Submit a Surrogate Bid.	5
Version History for the Surrogate Bid API.	16
API-Specific Disclaimers and Legal Information.	17

Surrogate Bid API

This guide is for developers creating applications to submit surrogate bids on behalf of participants (supplier users) in SAP Ariba Sourcing events.

This guide applies to:

- SAP Ariba Strategic Sourcing Suite
- SAP Ariba Sourcing

Note

For details on the maximum number of line items, questions, and other supported elements, refer to [Event Content Limits](#).

Restriction

Do not use this API and the interactive user interface to submit bids at the same time. Concurrent operations for the same event and participant using this API and the interactive user interface can create a condition that requires bids to be submitted multiple times before they are accepted.

Related guides

[Managing Projects, Teams, Documents, and Tasks](#)

[Event Management Guide](#)

Surrogate Bid API Endpoint

The Surrogate Bid API supports the `jobs` endpoint.

Use the `jobs` endpoint to post jobs that generate, download, and submit bid response sheets and to get job status information.

Workflow for the Surrogate Bid API

Use the Surrogate Bid API to submit bids on behalf of participants.

See [Developing an Application to Submit a Surrogate Bid \[page 5\]](#) for parameters and additional information.

⚠ Restriction

Do not use this API and the interactive user interface to submit bids at the same time. Concurrent operations for the same event and participant using this API and the interactive user interface can create a condition that requires bids to be submitted multiple times before they are accepted.

1. Obtain information for the client: ID of the event for which the bid will be submitted, ID of the buyer submitting the surrogate bid, and the ID of the participant.
2. The client creates an `export` job to generate a bid response sheet for the event.
Endpoint: `jobs`.
The client retains the value of the `jobId` from the response.
3. The client retrieves the status of the `export` job until the status is `Success`.
Endpoint: `jobs/{jobId}`.
The client retains the value of the `fileId` field from the response.
4. The client downloads the bid response sheet.
Endpoint: `/jobs/{jobId}/files/{fileId}`.
5. The user enters responses (bids) in the bid response sheet using Microsoft Excel.
6. The client creates an `import` job to upload your bid response sheet.
Endpoint: `jobs`.
The client retains the value of the `jobId` from the response.
7. The client retrieves the status of the `import` job until the status is `Success`.
Endpoint: `jobs/{jobId}`.
8. If the event includes questions or line item terms that accept attachments for responses, the client uploads attachments on behalf of participants.
 1. The client creates an `add_attachment_response` job to upload a file for an attachment.
Endpoint: `jobs`.
The client retains the value of the `jobId` from the response.
 2. The client retrieves the status of the `add_attachment_response` job until the status is `Success`.
Endpoint: `jobs/{jobId}`.
9. The client creates a `submit` job to submit the imported bid response sheet and any uploaded attachments for the event.
Endpoint: `jobs`.
The client retains the value of the `jobId` from the response.
10. The client retrieves the status of the `submit` job until the status is `Success`.
Endpoint: `jobs/{jobId}`.

Developing an Application to Submit a Surrogate Bid

This procedure describes how to use the Surrogate Bid API to submit bids on behalf of participants.

Prerequisites

- To create an application and request access to use the SAP Ariba APIs relevant to this feature, you must have access to the SAP Ariba Developer Portal. For details refer to [Steps to Start Using the SAP Ariba APIs in Help for the SAP Ariba Developer Portal](#).
- All queries made to this API must be authenticated using OAuth authentication. For details refer to [SAP Ariba Developer Portal Authentication in Help for the SAP Ariba Developer Portal](#).

Context

⚠ Restriction

Do not use this API and the interactive user interface to submit bids at the same time. Concurrent operations for the same event and participant using this API and the interactive user interface can create a condition that requires bids to be submitted multiple times before they are accepted.

📌 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.

Procedure

1. Obtain the following information:
 - ID of the event for which the bid will be submitted (as shown next to the event name in the user interface), such as Doc12345.
 - ID of a buyer user authorized to submit a surrogate bid for the event (as entered in the **User Name** field on the SAP Ariba sign in page). The web service will submit the surrogate bid as this user.
 - ID (user ID) of the participant (supplier) for whom the surrogate bid will be submitted. This participant must be invited to the event and for the items that the responses will be submitted for.

To find the user ID of a participant, a member of at least one of the following groups can view participant user information by clicking ► **Supplier and Customer Manager** ► **Users** ► from the Ariba Administrator page:

Category Manager
Commodity Manager
Event Administrator
Limited Event Administrator
Sourcing Project Administrator
Supplier/Customer Agent
Supplier/Customer Agent
Supplier/Customer Manager
Supplier Project Administrator

2. Create an export job to generate a bid response sheet for the event.

Use the Sourcing Surrogate Bid API jobs endpoint to post a request that creates an export job:

```
POST {{runtime_URL}}/jobs?user=user&passwordAdapter=passwordAdapter&realm=realm
```

Consumes: multipart/form-data.

Produces: application/json.

Body:

```
{
  "operation": "export",
  "eventId": "eventId",
  "supplierId": "user"
}
```

Table 1: Query String Parameters (Used in All Queries for This API)

Query String Parameter	Description
user	User ID of the enterprise (buyer) who will submit the surrogate bid. This user must be authorized to submit a surrogate bid for the event.
passwordAdapter	Password adapter used to authenticate the enterprise user, such as PasswordAdapter1 .
realm	Your site name.

Table 2: Request Body Parameters

Name	Required/Optional	Type	Description
operation	Required	string	The operation for which to create the job. Valid values for bid response sheets: export: Create a job to generate a bid response sheet. When the job completes successfully, you can download the bid response sheet using the <code>/jobs/{jobId}/files/{fileId}</code> endpoint. import: Create a job to import (upload) a bid response sheet. Valid values for attachments: add_attachment_response: Create a job to upload and submit an attachment on behalf of a participant as a response to a question or a term in a line item. delete_attachment_response: Create a job to delete an attachment from a participant's response. You can use this operation for an attachment that was submitted via the interactive user interface or this API. download_attachment_response: Create a job to get an attachment from a participant's response. You can use this operation for an attachment that was submitted via the inter-

Name	Required/Optional	Type	Description
			active user interface or this API. When the job completes successfully, you can download the attachment using the <code>/jobs/{jobId}/files/{fileId}</code> endpoint. replace_attachment_response: Create a job to replace an attachment in a participant's response. You can use this operation for an attachment that was submitted via the interactive user interface or this API. Valid value for bid response sheet and attachments: submit: Create a job to submit a previously uploaded bid response sheet and attachments.
eventId	Required	string	ID of the event, such as Doc12345 .
supplierId	Required	string	User ID of the participant (supplier) for whom the bids will be submitted.

Name	Required/Optional	Type	Description
useBuyerLocale	Optional	Boolean	Use this parameter to configure the buyer's locale for the bid response sheet. Valid values for the export and import operations are: true: Enables you to export and import the bid response sheet in either the buyer's or the supplier's locale by using the Surrogate Bid API. false: Disables the capability to export and import the bid response sheet in either the buyer's or the supplier's locale by using the Surrogate Bid API.
file	Optional	file with "x-mimetype": "application/vnd.ms-excel"	This parameter is needed for import, add_attachment_response and replace_attachment_response operations; omit it for all other operations. The bid response sheet to import.
itemNumber	Optional	string	This parameter is needed for add_attachment_response, delete_attachment_response, download_attachment_response, and replace_attachment_response operations; omit it for all other operations. The number identifying the question or line item for the attachment. The value must match an entry in the Number column of the bid response sheet.

Name	Required/Optional	Type	Description
termName	Optional	string	This parameter is needed for <code>add_attachment_response</code>, <code>delete_attachment_response</code>, <code>download_attachment_response</code>, and <code>replace_attachment_response</code> operations when <code>itemNumber</code> refers to a line item; omit it for all other operations and when <code>itemNumber</code> refers to a question. The name identifying the term in the line item for the attachment. The value must match the name for a term as specified in the first row of the bid response sheet.

Table 3: Response Fields

Field	Type	Description
operation	string	Type of operation for the job. Possible values: • <code>ADD_ATTACHMENT_RESPONSE</code> • <code>DELETE_ATTACHMENT_RESPONSE</code> • <code>DOWNLOAD_ATTACHMENT_RESPONSE</code> • <code>EXPORT</code> • <code>IMPORT</code> • <code>REPLACE_ATTACHMENT_RESPONSE</code> • <code>SUBMIT</code>
jobId	string	ID of the job created.
submittedBy	string	Enterprise (buyer) user who will submit the surrogate bid.
submittedFor	string	Participant (supplier) user for whom the surrogate bid will be submitted.

Retain the value of the `jobId`; you will need it to check the status of the job and to download the bid response sheet.

Request example:

```
POST {{runtime_URL}}/jobs?
user=aBuyer&passwordAdapter=PasswordAdapter1&realm=mySite
```

Body:

```
{
  "operation": "EXPORT",
  "eventId": "Doc6841882",
  "supplierId": "aSupplier"
}
```

Response example:

```
{
  "operation": "EXPORT",
  "jobId": "Doc6841882-~EXPORT-~aSupplier-~1571880836213",
  "submittedBy": "aBuyer",
  "submittedFor": "aSupplier"
}
```

3. Retrieve the status of the export job to determine if the job completed successfully and the bid response sheet is available for you to download.

Use the Sourcing Surrogate Bid API `/jobs/{jobId}` endpoint to retrieve the status of the export job:

```
GET {{runtime_URL}}/jobs/{jobId}?
user=user&passwordAdapter=passwordAdapter&realm=realm
```

Produces: application/json.

Consumes: multipart/form-data.

Table 4: Path Parameters

Path Parameter	Description
<code>jobId</code>	ID of the job for which you want status.

Table 5: Response Fields

Field	Type	Description
<code>jobId</code>	string	ID of the job.

Field	Type	Description
operation	string	Type of operation for the job. Possible values: • ADD_ATTACHMENT_RESPONSE • DELETE_ATTACHMENT_RESPONSE • DOWNLOAD_ATTACHMENT_RESPONSE • EXPORT • IMPORT • REPLACE_ATTACHMENT_RESPONSE • SUBMIT
jobStatus	string	Job status. Possible values: • New • Pending • Success • Failure • Fatal
submittedBy	string	Enterprise (buyer) user who will submit the surrogate bid.
submittedFor	string	Participant (supplier) user for whom the surrogate bid will be submitted.
fileId	string	ID of the bid response sheet for the operation.
messages	string	Description or additional information for the job status.

Retain the value of the `fileId` field; you will need it to download the bid response sheet.

If `jobStatus` is `New` or `Pending`, repeat this step until the status is `Success`, `Failure`, or `Fatal`. If the status is `Success`, continue to the next step. If the status is `Failure` or `Fatal`, examine the `messages` for more information.

Request example:

```
GET {{runtime_URL}}/jobs/Doc6841882-~EXPORT-~aSupplier-~1571880836213?
user=aBuyer&passwordAdapter=PasswordAdapter1&realm=mySite
```

Response example:

```
{
  "jobId": "Doc6841882-~EXPORT-~aSupplier-~1571880836213",
  "operation": "EXPORT",
  "jobStatus": "Success",
  "submittedBy": "aBuyer",
  "submittedFor": "aSupplier",
  "fileId": "Doc6841882-~EXPORT-~aSupplier-~1571880836213",
  "messages": [
    "Successfully completed export job."
  ]
}
```

```
} ]
```

4. Download the bid response sheet.

When the `jobStatus` from step 3 is Success, use the Sourcing Surrogate Bid API `/jobs/{jobId}/files/{fileId}` endpoint to download the bid response sheet.

```
GET {{runtime_URL}}/jobs/{jobId}/files/{fileId}?
user=user&passwordAdapter=passwordAdapter&realm=realm
```

Consumes: `multipart/form-data`.

Produces: `application/octet-stream`.

Table 6: Path Parameters

Path Parameter	Description
<code>jobId</code>	ID of the job used to export the bid response sheet.
<code>fileId</code>	ID of the bid response sheet, as returned in the response for step 3.

Request example:

```
GET {{runtime_URL}}/jobs/Doc6841882-
__EXPORT-__aSupplier-__1571880836213/files/Doc6841882-
__EXPORT-__aSupplier-
__1571880836213?user=aBuyer&passwordAdapter=PasswordAdapter1&realm=myrealm
```

5. Save the response as a Microsoft Excel worksheet file, which is the bid response sheet. Enter responses (bids) in the bid response sheet.
6. Create an `import` job to upload your bid response sheet.

Use the Sourcing Surrogate Bid API `/jobs` endpoint and to post a request that creates an `import` job.

```
POST {{runtime_URL}}/jobs?user=user&passwordAdapter=passwordAdapter&realm=realm
```

Consumes: `multipart/form-data`.

Produces: `application/json`.

Body:

```
{
  "operation": "import",
  "eventId": "eventId",
  "supplierId": "user",
  "file": "{file}"
}
```

Retain the value of the `jobId` from the response; you will need it to check the status of the job.

```
POST {{runtime_URL}}/jobs?
user=aBuyer&passwordAdapter=PasswordAdapter1&realm=mySite
```

Body:

```
{
  "operation": "import",
  "eventId": "Doc6841882",
  "supplierId": "aSupplier",
  "file" : {file}
}
```

7. Retrieve the status of the `import` job to determine if the job completed successfully.

Use the Sourcing Surrogate Bid API `/jobs/{jobId}` endpoint to retrieve the status of the `import` job. Use the procedure described in step 3 with the job ID for the `import` job.

If `jobStatus` is `New` or `Pending`, repeat this step until the status is `Success`, `Failure`, or `Fatal`. If the status is `Success`, continue to the next step. If the status is `Failure` or `Fatal`, examine the `messages` for more information.

8. If the event includes questions or line item terms that accept attachments for responses, upload attachments on behalf of participants.
 - a. Use the Sourcing Surrogate Bid API `/jobs` endpoint to post a request that creates an `add_attachment_response` job. This uploads a file for an attachment.

```
POST {{runtime_URL}}/jobs?
user=user&passwordAdapter=passwordAdapter&realm=realm
```

Consumes: `multipart/form-data`.

Produces: `application/json`.

Body:

```
{
  "operation": "add_attachment_response",
  "eventId": "eventId",
  "supplierId": "user",
  "file": "file",
  "itemNumber": "itemNumber",
  "termName": "termName"
}
```

- If the attachment is a response to a question, include the `itemNumber` body parameter.
- If the attachment is a response to a line item term, include the `itemNumber` and `termName` body parameters.
- The `itemNumber` must match a value in the `Number` column of the bid response sheet.
- The `termName` parameter must match the name for a term as specified in the first row of the bid response sheet.

Retain the value of the `jobId` from the response; you will need it to check the status of the job.

Request example:

```
POST {{runtime_URL}}/jobs?
user=aBuyer&passwordAdapter=PasswordAdapter1&realm=mySite
```

Body:

```
{
```

```

    "operation": "add_attachment_response",
    "eventId": "Doc6841882",
    "supplierId": "aSupplier",
    "file": "{file}",
    "itemNumber": "3.2",
    "termName": "Additional Information"
  }

```

- b. Use the Sourcing Surrogate Bid API `/jobs/{jobId}` endpoint to retrieve the status of the `add_attachment_response` job.

Use the procedure described in step 3 with the job ID for the `add_attachment_response` job.

If `jobStatus` is `New` or `Pending`, repeat this step until the status is `Success`, `Failure`, or `Fatal`. If the status is `Success` and you have an additional attachment to upload, repeat this step and create another `add_attachment_response` job; otherwise, continue to the next step. If the status is `Failure` or `Fatal`, examine the messages for more information.

9. Create a `submit` job to submit the imported bid response sheet and any uploaded attachments.

When the `jobStatus` from the `import` job and all `add_attachment_response` jobs is `Success`, use the Sourcing Surrogate Bid API `jobs` endpoint to post a request that creates a `submit` job. This job submits a previously imported bid response sheet and any attachments on behalf of the participant.

```
POST {{runtime_URL}}/jobs?user=user&passwordAdapter=passwordAdapter&realm=realm
```

Consumes: `multipart/form-data`.

Produces: `application/json`.

Body:

```

{
  "operation": "submit",
  "eventId": "eventId",
  "supplierId": "user"
}

```

Retain the value of the job ID from the response; you will use it to check the job status.

```
POST {{runtime_URL}}/jobs?
user=aBuyer&passwordAdapter>PasswordAdapter1&realm=mySite
```

Body:

```

{
  "operation": "submit",
  "eventId": "Doc6841882",
  "supplierId": "aSupplier"
}

```

10. Use the Sourcing Surrogate Bid API `/jobs/{jobId}` endpoint to retrieve the status of the `submit` job.

Use the procedure described in step 3 with the job ID for the `submit` job.

Version History for the Surrogate Bid API

The Surrogate Bid API may have multiple versions, only one of which is active at this time. Previous versions are deprecated and might be decommissioned at a later date.

API Version	Status	As of Release
1	Active	2005 (Q2 2020)

Related Information

[API versioning and deprecation on the developer portal](#)

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 © 2026 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>.

