

PUBLIC

Document Version: 2611 – 2026-11

Project Document Management API

SAP Ariba APIs
SAP Ariba Developer Portal
SAP Ariba Sourcing
SAP Ariba Contracts
SAP Strategic Sourcing Suite

© 2026 SAP SE or an SAP affiliate company. All rights reserved.

Content

Project Document Management API. 3
Project Document Management API Endpoints. 5
Workflow for Project Document Management API. 6
Identifying and Downloading Project Documents by Using the Project Document Management API 7
Creating a Document or Folder by Using the Project Document Management Creation API. 10
Updating Documents by Using the Project Document Management Modification API. 18
Changing the State of a Document in a Contract Workspace. 29
Interpretation of Project Document Management API Responses. 31
Error Messages Returned by the Project Document Management API. 35
Document and Project Types Supported by the Project Document Management API. 37

API-Specific Disclaimers and Legal Information. 39

Project Document Management API

This guide is for developers who use the Project Document Management API to develop client applications to search for project documents, view the properties of a project document, download specific project documents, and create folders or project documents of type Standard Document, Main Agreement, and Contract Addendum. The API can also update folders and project documents of type Standard Document, Main Agreement, and Contract Addendum in a Contract Workspace (Procurement).

The Project Document Management API collection contains various endpoints that enable you to:

- Create folders and project documents of type Standard Document, Main Agreement, and Contract Addendum
- Update folders and project documents of type Standard Document, Main Agreement, and Contract Addendum in a Contract Workspace (Procurement)
- Change the state of documents such as standard documents, Main Agreement, and contract addendum in a Contract Workspace (Procurement)
- Perform certain actions on various documents. For more information, see Document and Project Types Supported by the Project Document Management API [page 37].

For a list of project types supported by the Project Document Management API, see Document and Project Types Supported by the Project Document Management API [page 37].

This guide applies to:

- SAP Ariba APIs
- SAP Ariba Developer Portal
- SAP Ariba Contracts
- SAP Ariba Strategic Sourcing Suite

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.
- User whose credentials are passed in the API call must be a project owner or a team member, or should have view permissions.

Restrictions

- You can use the Project Document Management Creation API to create Standard Document, Contract Addendum, Main Agreement, and folder only.

- You can use the Project Document Management Creation API to create project documents in a Contract Workspace (Procurement), but you cannot create project documents in any other workspace.
- Translations are not supported for the name and description fields. The base language of the project is set as the base language of the project document.
- You cannot create multiple folders or documents by making a single API call using the Project Document Management Creation API.
- When multiple users simultaneously try to create project documents (Standard Document, Main Agreement, and Contract Addendum) with the same name using the Project Document Management Creation API, multiple documents with the same name might get created for a project.
- The Project Document Management API does not support visibility, editability, and validity of fields at the document level.
- You can use the Project Document Management Modification API to upload a zip file as a Standard Document, but you cannot extract a zip file to a Contract Workspace (Procurement).
- You can use the Project Document Management Modification API to update project documents in a Contract Workspace (Procurement only), but you cannot update project documents in any other workspace.
- You cannot update multiple folders or documents by making a single API call using the Project Document Management Modification API.
- You cannot use the Project Document Management State Change API to modify the state of documents in contract workspaces that are in Published, On Hold, Closed, or Expired state.
- When multiple users simultaneously try to update project documents (Standard Document, Main Agreement, and Contract Addendum) with the same name, multiple documents with the same name might get updated for a project.
- In the Project Document Management Creation and Modification APIs, the file size limit for standard documents is 20 MB. When the `Application.ACM.IcertisIntegration` parameter is set to `false`, the file size limit for Main Agreements and Contract Addendums is 50 KB. When the `Application.ACM.IcertisIntegration` parameter is set to `true`, the file size limit for Main Agreements and Contract Addendums is 20 MB. For more information about the parameter, see Enable SAP Ariba Contracts integration and contract data sharing with Icertis.
- If you are using SAP Ariba Contracts integrated with Icertis Contracts Intelligence for SAP Ariba Solutions (that is the `Application.ACM.IcertisIntegration` parameter is set to `true`), the Main Agreement and Contract Addendum documents in Contract workspaces are processed in ICI and not in SAP Ariba Contracts. Therefore, if you set the `Application.ACM.IcertisIntegration` parameter to `false`, the Main Agreement and Contract Addendum documents that were not processed by SAP Ariba Contracts will not be shown in the outline view. For more information about the parameter, see Enable SAP Ariba Contracts integration and contract data sharing with Icertis.
- The filters specified in the `GET /projectDocuments` API endpoint act like search operations in SAP Ariba solutions with similar performance, which means there could be a slight delay before newly created or updated data appears. If the data is not available, run the API again after some time. The time it takes to run the API again is determined by the API rate limits that are configured for your application. For more information, see Tracking Your API Usage to Avoid Exceeding Rate Limits.

Related guides

Help for the SAP Ariba Developer Portal

Project Document Management API Endpoints

This topic describes the endpoints in the Project Document Management API.

API Name	Endpoint	Description
Project Document Management Retrieval API	`GET /projectDocuments`	Retrieves the list of documents that match the search criteria you specify. You can search for documents based on one or more of conditions such as the document title, document type, and the last modified date. For additional information, refer to Identifying and Downloading Project Documents by Using the Project Document Management API [page 7].
	`GET /projectDocuments/documentId`	Retrieves the list of documents, along with the metadata, associated with the specified project ID or folder ID. If a document ID is given in the query, the query retrieves the associated metadata. The metadata contains ID, name, size, folder, version, owner, type, dates the document was created and was last modified, signed document ID, and so on. For additional information, refer to Identifying and Downloading Project Documents by Using the Project Document Management API [page 7].
Project Document Management Download API	`GET /projectDocuments/documentId/files`	Downloads the document that matches the specified *documentId*. For additional information, refer to Identifying and Downloading Project Documents by Using the Project Document Management API [page 7].
Project Document Management Creation API	`POST /projectDocuments/{parentId}`	Creates folders or project documents of type Standard Document, Main Agreement, and Contract Addendum. For additional information, refer to Creating a Document or Folder by Using the Project Document Management Creation API [page 10].

API Name	Endpoint	Description
Project Document Management Modification API	`PUT /projectDocuments/{documentId}`	Updates folders and project documents of type Standard Document, Main Agreement, and Contract Addendum in a Contract Workspace (Procurement). For additional information, refer to Updating Documents by Using the Project Document Management Modification API [page 18].
Project Document Management State Change API	`POST /projectDocuments/documentID/state`	Updates the state of documents such as standard documents, Main Agreement, and contract addendum in a contact workspace. For a Main Agreement or a contract addendum document, you can change the state from draft to publish, publish to finalize, and from finalize to draft. For standard documents, you can change the state from draft to publish. For additional information, refer to Changing the State of a Document in a Contract Workspace [page 29].

Workflow for Project Document Management API

The Project Document Management API enables you to create and update folders and project documents, search for documents in sourcing projects and contract workspaces, view properties of a specified document or documents associated with a specified project ID or folder ID, and download documents from a sourcing project or contract workspace.

The following workflow describes a logical sequence in which you can invoke and implement the various endpoint of the Project Document Management API.

1. The client application searches for documents in a sourcing project or contract workspace based on:
   - one or more search criteria such as document title, document type, or the last modified date. You can also specify the document properties you want to view along with the search results.
   - project ID or folder ID to view documents. The query returns the list of documents associated with the specified project ID or folder ID.
2. The client application queries for the properties of a document by specifying the document ID.
3. Buyer user reviews the search results or document properties to identify the document to download.
4. The client application downloads the specified document by passing the document ID in the API call.
5. (Optional) The client application creates folders or project documents of type Standard Document, Main Agreement, and Contract Addendum.
6. (Optional) The client application updates folders and project documents of type Standard Document, Main Agreement, and Contract Addendum in a Contract Workspace (Procurement).
7. (Optional) The client application modifies the state of documents such as Main Agreement, Contract Addendum, and standard documents in a Contract Workspace (Procurement).

Identifying and Downloading Project Documents by Using the Project Document Management API

This topic explains how you can use the Project Document Management API to search for project documents, view the properties of a project document, and download specific project documents.

Context

Using the Project Document Management API, you can:

- Search project documents using the document title, the last modified date, the document type, or a combination of any of these filtering options. For example, you can search for ContractTerms modified on 2020-02-27 by specifying the document type and the last modified date as the filtering criteria.
- View the list of all the documents, along with the associated metadata, in a project or folder, or the properties of a specific document if you know its document ID.
- View details of the signed document associated with the original document sent for signature as part of the signature task. At the project or folder level, the API enables you to get details of the signed document associated with each document in the project or the folder.
- Download project documents using the corresponding document ID.

> **Note**
>
> When downloading files through the API, files with non-ASCII characters (for example, é, ü) in their filenames are renamed to `response.bin`. To ensure that files are downloaded with their original filenames, you must enable the corresponding functionality on your site. Contact SAP Ariba Support to enable the **Support for Accurate Filename Handling in Document Download API** feature.

> **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.

> **Note**
>
> For a list of response codes for the API calls, see Error Messages Returned by the Project Document Management API [page 35].

Procedure

1. To search for project documents, use the `GET /projectDocuments` endpoint as follows:

```
GET {{runtime_url}}/projectDocuments?$select=title,projectID,owner&$filter=title
eq {}and documentType and {<documentype>}and lastModifiedDate eq {<yyyy-MM-
dd>}&$top=<number>&$count=<true|
```

```
false>&$skip=0&user=<username>&passwordAdapter=<PasswordAdapterName>&realm=<realm
>
```

Where:

$select Specify the document metadata that you want to be retrieved for the documents that meet the specified search criteria. You can specify any of the following metadata elements:

`name`, `version`,`parentFolderId`, `projectId`, `owner`, `lastModifiedDate`, `creationDate`, `documentType`, `active`, `status`

$filter Specify one or more of the search criteria. You can use the `and` condition to combine multiple criteria.

- `LastModified`–Specify the last modified date along with one of the following operators: `gte` (greater-than or equal), `lte` (lesser-than or equal), `gt` (greater-than), `lt` (lesser-than), or `eq` (equals).
- `DocumentType`–Specify the type of the document with an `eq` operator. Supported values are:

  `ContractAddendum`, `MainAgreement`, `CleansheetDocument`, `LineItemDocument`, `ContentDocument`, `Event`, `SavingsForm`, `AssembledPDF`, `AnalyticalReport`, `CompoundReport`, `ContractClause`, `ProfileQuestionnaire`, `Scorecard` `ContractTerms`

> **Note**
>
> If no filter criterion is selected, a list of documents starting with the latest documents is returned. The number of results depends on the value set for `$top`.
>
> When the filter is set as `DocumentType eq Document`, the response includes the main document type and its subclasses. To display only the main document type and exclude the subclasses, set the **Hide document type subclasses from project document search API results** (`Application.ACM.ExcludeSubclassesFromDocumentSearchAPI`) parameter to **Yes**. For more information about the parameter, see Hide document type subclasses from project document search API results.

$top Specify the number of results to be included in one page of the response. The default and maximum allowed results are 500. For example, `$top=10`.

$skip Specify the number of documents to be skipped (offset) from the beginning of the result set for uses such as pagination. The default setting is 0. You can use the `$skip` parameter to view a particular page. For example, if `$top=10` and `$skip=30`, the query directly loads the fourth page of results.

$count Specify whether the total number of matching results be displayed along with the results. Supported values are `TRUE` and `FALSE`.

A list of documents that match the specified search criteria, along with the document properties you specified for the `$select` parameter, is returned.

> **Note**
>
> The filters specified in the `GET /projectDocuments` API endpoint act like search operations in SAP Ariba solutions with similar performance, which means there could be a slight delay before newly created or updated data appears. If the data is not available, run the API again after some time. The time it takes to

run the API again is determined by the API rate limits that are configured for your application. For more information, see Tracking Your API Usage to Avoid Exceeding Rate Limits.

2. To view all the documents associated with a project ID or folder ID, use the `GET /projectDocuments/documentID` endpoint as follows:

```
GET {{runtime_url}}/projectDocuments/documentID?
user=username&passwordAdapter=PasswordAdapterName&realm=realm
```

Where *documentID* denotes the project ID or the folder ID.

> **Note**
>
> Alternatively, you can use this endpoint and the API call explained in this step to view properties of a specified document. To view properties of a document, specify the document ID as *documentID* in the API call.

3. Review the results to identify the document that you want to download.

Documents that have a signed document associated with them will have the ID of the signed document populated in the *signedDocumentId* field.

> **Note**
>
> Limitation:
>
> If the version of the signed document associated with an original document is changed because of a manual revision or replacement of the document, the API still links the latest version of the signed document with the original document.

4. To download a document from a project or workspace, use the `GET /projectDocuments/documentId/files` endpoint as follows:

```
GET {{runtime_url}}/projectDocuments/documentID/files?
user=username&passwordAdapter=PasswordAdapterName&realm=realm
```

Where *documentID* denotes the document ID of the file that you want to download. While downloading a signed document associated with a document, the *documentID* denotes the signed document ID of the file that you want to download.

> **Note**
>
> This endpoint does not support Project ID or folder ID for *documentID* .

The specified document is downloaded.

Creating a Document or Folder by Using the Project Document Management Creation API

This topic explains how you can create a document or folder by using the Project Document Management Creation API.

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.
- If you are using SAP Ariba Contracts integrated with Icertis Contracts Intelligence for SAP Ariba Solutions, you cannot process the Main Agreement or Contract Addendum documents in SAP Ariba as these documents are processed in ICI. If you must process the Main Agreement or Contract Addendum document in SAP Ariba, you must disable the integration by setting the `Application.ACM.IcertisIntegration` parameter to `false`. For more information about the parameter, see Enable SAP Ariba Contracts integration and contract data sharing with Icertis. However, note that disabling the integration might adversely impact some of the business process flows and should be done only after careful considerations and consultations.
- You must obtain the parent folder ID in which the document or folder needs to be created.

Context

The `POST /projectDocuments/{parentId}` endpoint of the Project Document Management Creation API enables you to create folders or project documents of type Standard Document, Main Agreement, and Contract Addendum.

> **Note**
>
> In the Project Document Management Creation API, the file size limit for standard documents is 20 MB. When the `Application.ACM.IcertisIntegration` parameter is set to `false`, the file size limit for Main Agreements and Contract Addendum documents is 50 KB. When the `Application.ACM.IcertisIntegration` parameter is set to `true`, the file size limit for Main Agreements and Contract Addendum documents is 20 MB. For more information about the parameter, see Enable SAP Ariba Contracts integration and contract data sharing with Icertis.

> **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 ID of the parent folder in which the document or folder needs to be created. You can view the parent folder ID in the user interface.
2. You can create a folder or a project document.
   - To create a Standard Document, use the `POST /projectDocuments/{parentId}` endpoint.
     Example:

     ```
     POST {{runtime_URL}}/projectDocuments/CW12345?
     user=myUser&passwordAdapter=PasswordAdapter1&realm=mySite
     ```

     Where:
     - `realm` is your site name
     - `user` is the User ID of the user for whom the Project Document Management Creation API will perform the action
     - `passwordAdapter` is the password adapter used to authenticate the user. For example: `PasswordAdapter1`

     At a minimum, the request body must contain the `file` element. You can upload the Standard Document file in the `file` element. File extension of the uploaded file must be the same as the extension specified in the title.
     Optionally, using the `properties` element in the request body, you can update attributes for the Standard Document you have uploaded (header fields in the **Overview** tab):
     - `Name`: Specify the file name of the Standard Document that you want to create. The name specified for the Standard Document must have the same extension as the uploaded file. For example, if the name specified is `Standard_Document.docx` then the uploaded file must be of type `.docx`.
       If this value is not specified, then uploaded file name will be set as the document title. However, if this value is specified, then it is prioritized over the uploaded file name.
       The name specified for the Standard Document is unique at the parent level.
     - `description`: Specifies the description of the Standard Document.
     - `owner`: Specifies the owner of the Standard Document. An owner can be an individual user, a global group or a project group. If the owner information is invalid, an appropriate error message will be returned.
       Default value: current session user
       `owner` element contains the following fields:
       - `name`: Specify the name of the user or the group
       - `uniqueName`: Specify the unique name of the user or the group
       - `passwordAdapter`: Specify the password adapter used to authenticate the user. This field is required only for user.
       - `type`: Specify the owner type. Valid values: `User`, `ProjectGroup`, `Group`
     - Specify the `documentType` element. Valid value: `Document`
     - `useAs`: Valid values are:
       - `ContractContent`: If you specify this value, then only files of type `xlsx` are allowed.
         The Excel file will contain contract terms or pricing terms. However, if the `Parameter Application.ACM.EnableExcelXLSXContractContent` is disabled, then files of type `xlsx` are not allowed.

> **Note**
>
> To download pricing terms in an `xls` file, contact SAP Ariba Support to enable the feature **SC-20703: Enable Old 2003 Excel Format for Source To Contract Pricing Term Excel file** and the parameter `Application.ACM.EnableExcelXLSXContractContent` for your site.

  - `ContractAttachment`
  - `None`
- `accessControl`: Specifies the access control applicable on the document. You can specify multiple values, delimited by commas. For example: `[ “Legal Information”, “Private to Team Members” ]`
- `createAnnouncement`: Specifies whether to announce the creation of the new Standard Document. Set the value to `true` to announce the creation of the Standard Document.
  Default value: `false`
- `publishRequired`: If the value is set to either `true` or `false`, values specified as part of the current version will be updated. If the key-value pair is not specified, then the current version will not be updated.
  Default value: `false`
  Use the following data model for the `properties` field:

```
{
  "name": "Standard_Document.docx",
  "description": "Standard Document",
  "owner":
{
  "name": "Standard_Document.docx",
  "description": "Standard Document",
  "owner": {
    "name": "test",
    "uniqueName": "molson",
    "passwordAdapter": "PasswordAdapter1",
    "type": "User"
  },
  "documentType": "Document",
  "useAs": "None",
  "accessControl": [
    "Legal Information"
  ],
  "createAnnouncement": false,
  "publishRequired": false,
}
```

The Project Document Management Creation API returns details of the Standard Document created. For example:

```
{
    "id": "Doc4583517",
    "name": "Standard_Document.docx",
    "size": 13718,
    "version": "1",
    "parentFolderId": "CW6127",
    "projectId": "CW6127",
    "owner": "molson",
    "lastModifiedDate": "2022-06-13T09:03:07.510+0000",
    "creationDate": "2022-06-13T09:03:07.510+0000",
    "documentType": "Document",
    "active": true,
    "status": "Draft",
    "otherVersion": []
```

```
}
```

- To create a Main Agreement, use the `POST /projectDocuments/{parentId}` endpoint.
  Example:

```
POST {{runtime_URL}}/projectDocuments/CW12345?
user=myUser&passwordAdapter=PasswordAdapter1&realm=mySite
```

  Where:

  - `realm` is your site name
  - `user` is the User ID of the user for whom the Project Document Management Creation API will perform the action
  - `passwordAdapter` is the Password adapter used to authenticate the user. For example: `PasswordAdapter1`

  At a minimum, the request body must contain the following elements:

  - `file`: Select the Main Agreement file. You can upload the Main Agreement file in the `file` element. File extension of the uploaded file must be the same as the extension specified in the title.
  - In the `properties` element, specify the `documentType` field. Valid value: `MainAgreement`

  Optionally, using the `properties` element in the request body, you can update attributes for the Main Agreement you have uploaded (header fields in the **Overview** tab):

  - `Name`: Specify the file name of the Main Agreement that you want to create. The name specified for the Main Agreement must have the same extension as the uploaded file. For example, if the file name is `Main_Agreement.docx` then the uploaded file must be of type `.docx`.
    If this value is not specified, then uploaded file name will be set as the document title. However, if this value is specified, then it is prioritized over the uploaded file name.
    The name specified for the Main Agreement is unique at the parent level.
  - `description`: Specifies the description of the Main Agreement.
  - `owner`: Specifies the owner of the Main Agreement. An owner can be an individual user, a global group or a project group. If the owner information is invalid, an appropriate error message will be returned.
    Default value: current session user
    `owner` element contains the following fields:
    - `name`: Specify the name of the user or the group
    - `uniqueName`: Specify the unique name of the user or the group
    - `passwordAdapter`: Specify the password adapter used to authenticate the user. This field is required only for user.
    - `type`: Specify the owner type. Valid values: `User`, `ProjectGroup`, `Group`
  - `useAs`: Valid values are:
    - `ContractContent`: If you specify this value, then only files of type `xlsx` are allowed.
      The Excel file will contain contract terms or pricing terms. However, if the `Parameter Application.ACM.EnableExcelXLSXContractContent` is disabled, then files of type `xlsx` are not allowed.

      > **Note**
      >
      > To download pricing terms in an `xls` file, contact SAP Ariba Support to enable the feature **SC-20703: Enable Old 2003 Excel Format for Source To Contract Pricing Term Excel file** and the parameter `Application.ACM.EnableExcelXLSXContractContent` for your site.

    - `ContractAttachment`

- `None`

- `accessControl`: Specifies the access control applicable on the document. You can specify multiple values, delimited by commas. For example: `[ “Legal Information”, “Private to Team Members” ]`
- `createAnnouncement`: Specifies whether to announce the creation of the new Main Agreement. Set the value to `true` to announce the creation of the Main Agreement.
  Default value: `false`
- `publishRequired`: If the value is set to either `true` or `false`, values specified as part of the current version will be updated. If the key-value pair is not specified, then the current version will not be updated. Default value: `false`
- `autoFinalizeOnPublish`: If the value is set to `true`, then the Main Agreement is automatically finalized when the Contract Workspace (Procurement) containing the Main Agreement is published. If the value is set to `false`, then the Main Agreement is not automatically finalized when the Contract Workspace (Procurement) containing the Main Agreement is published.
  Default value - `false`
  Use the following data model for the `properties` field:

```
{
  "name": "Main_Agreement.docx",
  "description": "Main Agreement",
  "owner": {
    "name": "test",
    "uniqueName": "molson",
    "passwordAdapter": "PasswordAdapter1",
    "type": "User"
  },
  "documentType": "MainAgreement",
  "useAs": "None",
  "accessControl": [
    "Legal Information"
  ],
  "createAnnouncement": false,
  "publishRequired": false,
  "autoFinalizeOnPublish": false
}
```

The Project Document Management Creation API returns details of the created Main Agreement.
For example:

```
{
    "id": "Doc30964352",
    "name": "Main_Agreement.docx",
    "size": 13712,
    "version": "1",
    "parentFolderId": "CW42675",
    "projectId": "CW42675",
    "owner": "molson",
    "lastModifiedDate": "2022-06-21T10:44:38.081+0000",
    "creationDate": "2022-06-21T10:44:38.081+0000",
    "documentType": "Main Agreement",
    "active": true,
    "status": "Draft",
    "otherVersion": []
}
```

- To create a Contract Addendum, use the `POST /projectDocuments/{parentId}` endpoint.

Example:

```
POST {{runtime_URL}}/projectDocuments/CW12345?
user=myUser&passwordAdapter=PasswordAdapter1&realm=mySite
```

Where:

- `realm` is your site name
- `user` is the User ID of the user for whom the Project Document Management Creation API will perform the action
- `passwordAdapter` is the Password adapter used to authenticate the user. For example: `PasswordAdapter1`

At a minimum, the request body must contain the following elements:

- `file`: Select the Contract Addendum file. You can upload the Contract Addendum file in the `file` element. File extension of the uploaded file must be the same as the extension specified in the title.
- In the `properties` element, specify the `documentType` field. Valid value: `ContractAddendum`

Optionally, using the `properties` element in the request body, you can update attributes for the Contract Addendum you have uploaded (header fields in the **Overview** tab):

- `Name`: Specify the file name of the Contract Addendum that you want to create. The name specified for the Contract Addendum must have the same extension as the uploaded file. For example, if the file name is `Contract_Addendum.docx` then the uploaded file must be of type `.docx`.
  If this value is not specified, then uploaded file name will be set as the document title. However, if this value is specified, then it is prioritized over the uploaded file name.
  The name specified for the Contract Addendum is unique at the parent level.
- `description`: Specifies the description of the Contract Addendum.
- `owner`: Specifies the owner of the Contract Addendum. An owner can be an individual user, a global or a project group. If the owner information is invalid, an appropriate error message will be returned.
  Default value: current session user
  `owner` element contains the following fields:
  - `name`: Specify the name of the user or the group
  - `uniqueName`: Specify the unique name of the user or the group
  - `passwordAdapter`: Specify the password adapter used to authenticate the user. This field is required only for user.
  - `type`: Specify the owner type. Valid values: `User`, `ProjectGroup`, `Group`
- `useAs`: Valid values are:
  - `ContractContent`: If you specify this value, then only files of type `xlsx` are allowed.
    The Excel file will contain contract terms or pricing terms. However, if the `Parameter Application.ACM.EnableExcelXLSXContractContent` is disabled, then files of type `xlsx` are not allowed.

    > **Note**
    >
    > To download pricing terms in an `xls` file, contact SAP Ariba Support to enable the feature **SC-20703: Enable Old 2003 Excel Format for Source To Contract Pricing Term Excel file** and the parameter `Application.ACM.EnableExcelXLSXContractContent` for your site.

  - `ContractAttachment`
  - `None`

- `accessControl`: Specifies the access control applicable on the document. You can specify multiple values, delimited by commas. For example: `[ “Legal Information”, “Private to Team Members” ]`
- `createAnnouncement`: Specifies whether to announce the creation of the new Contract Addendum. Set the value to `true` to announce the creation of the Contract Addendum.
  Default value: `false`
- `publishRequired`: If the value is set to either `true` or `false`, values specified as part of the current version will be updated. If the key-value pair is not specified, then the current version will not be updated. Default value: `false`
- `autoFinalizeOnPublish`: If the value is set to `true`, then the Contract Addendum is automatically finalized when the Contract Workspace (Procurement) containing the Contract Addendum is published. If the value is set to `false`, then the Contract Addendum is not automatically finalized when the Contract Workspace (Procurement) containing the Contract Addendum is published.
  Default value - `false`
  Use the following data model for the `properties` field:

```
{
  "name": "Contract_Addendum.docx",
  "description": "Contract Addendum",
  "owner": {
    "name": "test",
    "uniqueName": "molson",
    "passwordAdapter": "PasswordAdapter1",
    "type": "User"
  },
  "documentType": "ContractAddendum",
  "useAs": "None",
  "accessControl": [
    "Legal Information"
  ],
  "createAnnouncement": false,
  "publishRequired": false,
  "autoFinalizeOnPublish": false
}
```

The Project Document Management Creation API returns details of the created Contract Addendum. For example:

```
{
    "id": "Doc30968627",
    "name": "Contract_Addendum.docx",
    "size": 13685,
    "version": "1",
    "parentFolderId": "CW42675",
    "projectId": "CW42675",
    "owner": "molson",
    "lastModifiedDate": "2022-06-21T10:51:27.613+0000",
    "creationDate": "2022-06-21T10:51:27.613+0000",
    "documentType": "Contract Addendum",
    "active": true,
    "status": "Draft",
    "otherVersion": []
}
```

- To create a folder, use the `POST /projectDocuments/{parentId}` endpoint.

Example:

```
POST {{runtime_URL}}/projectDocuments/CW12345?
user=myUser&passwordAdapter=PasswordAdapter1&realm=mySite
```

Where:

- `realm` is your site name
- `user` is the User ID of the user for whom the Project Document Management Creation API will perform the action
- `passwordAdapter` is the Password adapter used to authenticate the user. For example: `PasswordAdapter1`

At a minimum, the request body must contain the following elements:

- `Name`: Specify the name of the folder that you want to create.
  The name of the folder must be unique at the parent level. For example, `Folder1`.
- In the `properties` element, specify the `documentType` field. Valid value: `Folder`

Optionally, you can update attributes of the folder using the `properties` element in the request body:

- `description`: Specify a description of the folder.
- `owner`: Specify the owner of the folder. An owner can be an individual user, a global group or a project group. If the owner information is invalid, an appropriate error message will be returned.
  Default value: current session user
  `owner` element contains the following fields:
  - `name`: Specify the name of the user or the group
  - `uniqueName`: Specify the unique name of the user or the group
  - `passwordAdapter`: Specify the password adapter used to authenticate the user. This field is required only for user.
  - `type`: Specify the owner type. Valid values: `User`, `ProjectGroup`, `Group`
- `accessControl`: Specifies the access control applicable on the document. You can specify multiple values, delimited by commas. For example: `[ “Legal Information”, “Private to Team Members” ]`

Use the following data model for the `properties` field:

```
{
  "name": "Folder1",
  "description": "Folder",
  "owner": {
    "name": "test",
    "uniqueName": "molson",
    "passwordAdapter": "PasswordAdapter1",
    "type": "User"
  },
  "documentType": "Folder",
  "accessControl": [
    "Legal Information"
  ],
}
```

The Project Document Management Creation API returns details of the created folder.
For example:

```
{
    "id": "WS4584427",
    "name": "Folder1",
    "version": "1",
```

```
    "parentFolderId": "CW6127",
    "projectId": "CW6127",
    "owner": "molson",
    "lastModifiedDate": "2022-06-13T12:10:38.782+0000",
    "creationDate": "2022-06-13T12:10:38.782+0000",
    "documentType": "Folder",
    "active": true,
    "childrenDocument": []
}
```

To troubleshoot your queries to the Project Document Management Creation API, refer to Error Messages Returned by the Project Document Management API [page 35].

Updating Documents by Using the Project Document Management Modification API

This topic explains how you can update documents by using the Project Document Management Modification API.

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.
- If you are using SAP Ariba Contracts integrated with Icertis Contracts Intelligence for SAP Ariba Solutions, you cannot process the Main Agreement or Contract Addendum documents in SAP Ariba as these documents are processed in ICI. If you must process the Main Agreement or Contract Addendum document in SAP Ariba, you must disable the integration by setting the `Application.ACM.IcertisIntegration` parameter to `false`. For more information about the parameter, see Enable SAP Ariba Contracts integration and contract data sharing with Icertis. However, note that disabling the integration might adversely impact some of the business process flows and should be done only after careful considerations and consultations.
- You must obtain the document ID that you want to update.

Context

The `PUT /projectDocuments/{documentId}` endpoint of the Project Document Management Modification API enables you to update folders and project documents of type Standard Document, Main Agreement, and Contract Addendum in a Contract Workspace (Procurement).

> **Note**
>
> In the Project Document Management Modification API, the file size limit for standard documents is 20 MB. When the `Application.ACM.IcertisIntegration` parameter is set to `false`, the

file size limit for Main Agreements and Contract Addendum documents is 50 KB. When the `Application.ACM.IcertisIntegration` parameter is set to `true`, the file size limit for Main Agreements and Contract Addendum documents is 20 MB. For more information about the parameter, see Enable SAP Ariba Contracts integration and contract data sharing with Icertis.

> **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 ID of the document or folder that you want to update. You can view the document ID in the user interface.
2. You can update a folder or a project document.
   - To update a Standard Document, use the `PUT /projectDocuments/{documentId}` endpoint. Example:

     ```
     PUT {{runtime_URL}}/projectDocuments/Doc12345678?
     user=myUser&passwordAdapter=PasswordAdapter1&realm=mySite
     ```

     Where:
     - `realm` is your site name
     - `user` is the User ID of the user for whom the Project Document Management Modification API will perform the action
     - `passwordAdapter` is the password adapter used to authenticate the user. For example: `PasswordAdapter1`

     In the request body, you can send the `file` that you want to replace and `properties` metadata. At a minimum, either of these fields is mandatory.
     In the `file` element, select the Standard Document file that you want to replace with.
     The Project Document Management Modification API supports all the file extensions that are supported in the UI, except for the Zip format extraction.
     Using the `properties` element in the request body, you can update attributes for the Standard Document you have uploaded (header fields in the **Overview** tab):
     - `name`: You can change the title of the Standard Document using this field. If this value is not specified, then the current version document name remains as the document title. The name specified for the Standard Document must have the same extension as the uploaded file. For example, if the name specified is `Standard_Document.docx` then the uploaded file must be of type `.docx`. The name specified for the Standard Document is unique at the parent level.
     - `description`: Specify the description of the Standard Document.
     - `editors`: Specify the editor of the Standard Document. An editor can be an individual user, a global group or a project group. If the editor information is not provided or invalid, an appropriate error message will be returned.
       `editors` element contains the following fields:
       - `name`: Specify the name of the user or the group.

- `uniqueName`: Specify the unique name of the user or the group. If `uniqueName` is null, then `editors` value is set to empty.
- `passwordAdapter`: Specify the password adapter used to authenticate the user. This field is required only for user.
- `type`: Specify the owner type. Valid values: `User`, `ProjectGroup`, `Group`

- `useAs`: Valid values are:
  - `ContractContent`: If you specify this value, then only files of type `xlsx` are allowed.
    The Excel file will contain contract terms or pricing terms. However, if the `Parameter Application.ACM.EnableExcelXLSXContractContent` is disabled, then files of type `xlsx` are not allowed.

    > **Note**
    >
    > To download pricing terms in an `xls` file, contact SAP Ariba Support to enable the feature **SC-20703: Enable Old 2003 Excel Format for Source To Contract Pricing Term Excel file** and the parameter `Application.ACM.EnableExcelXLSXContractContent` for your site.

  - `ContractAttachment`
  - `None`
- `owner`: Specify the owner of the Standard Document. An owner can be an individual user, a global group or a project group. If the owner information is not provided or invalid an appropriate error message will be returned.
  The `owner` element contains the following fields:
  - `name`: Specify the name of the user or the group.
  - `uniqueName`: Specify the unique name of the user or the group.
  - `passwordAdapter`: Specify the password adapter used to authenticate the user. This field is required only for user.
  - `type`: Specify the owner type. Valid values: `User`, `ProjectGroup`, `Group`
- `saveWithNewVersion`:
  If the value is set to `true`, a major version is created. For example, if v1 is the current version, then the new version will be v2.
  If the value is set to `false`, a minor version gets created. For example, if the current version is v1, then the new version will be v1(editing).
  If the key-value pair is not sent as part of the `properties` element, a minor version gets created.
- `versionComment`: Comment for the new version of the document.
- `publishRequired`: If the value is set to either `true` or `false`, values specified as part of the current version will be updated. If the key-value pair is not specified or if the value is empty, the current version value will not be updated.
- `accessControl`: Specify the access control applicable on the document. You can specify multiple values, delimited by commas. For example: `[ “Legal Information”, “Private to Team Members” ]`

Use the following data model for the `properties` field:

```
{
    "name": "StandardDocument.docx",
    "description" : "standard document description",
    "editors": {
            "name": "test",
            "uniqueName": "customersupportadmin",
            "passwordAdapter": "PasswordAdapter1",
```

```
            "type": "User"
     },
    "useAs": "ContractAttachment",
    "owner": {
           "name": "test",
           "uniqueName": "customersupportadmin",
           "passwordAdapter": "PasswordAdapter1",
           "type": "User"
     },
    "saveWithNewVersion":true,
    "versionComment" : "my version comment",
    "publishRequired": true,
    "accessControl": [
        "Legal Information",
        "Private To Team Members"
     ]
}
```

The Project Document Management Modification API returns details of the updated Standard Document. For example:

```
{
    "id": "Doc28548356",
    "name": "StandardDocument.docx",
    "size": 237180,
    "version": "5",
    "parentFolderId": "CW39412",
    "projectId": "CW39412",
    "owner": "customersupportadmin",
    "lastModifiedDate": "2022-06-17T06:06:09.889+0000",
    "creationDate": "2022-05-05T07:32:38.155+0000",
    "documentType": "Document",
    "active": true,
    "status": "Draft",
    "otherVersion": [
        {
            "id": "Doc28552269",
            "name": "mydoc_01.docx",
            "version": "4",
            "parentFolderId": "CW39412",
            "projectId": "CW39412",
            "owner": "customersupportadmin",
            "lastModifiedDate": "2022-05-05T11:17:25.738+0000",
            "creationDate": "2022-05-05T07:32:38.155+0000",
            "documentType": "Document",
            "active": true,
            "status": "Draft"
        },
        ...,
        {
            "id": "Doc28532774",
            "name": "Ice cream.docx",
            "version": "1",
            "parentFolderId": "CW39412",
            "projectId": "CW39412",
            "owner": "customersupportadmin",
            "lastModifiedDate": "2022-05-05T07:32:38.155+0000",
            "creationDate": "2022-05-05T07:32:38.155+0000",
            "documentType": "Document",
            "active": true,
            "status": "Draft"
        }
    ]
}
```

- To update a Main Agreement, use the `PUT /projectDocuments/{documentId}` endpoint.
  Example:

  ```
  PUT {{runtime_URL}}/projectDocuments/Doc12345678?
  user=myUser&passwordAdapter=PasswordAdapter1&realm=mySite
  ```

  Where:

  - `realm` is your site name
  - `user` is the User ID of the user for whom the Project Document Management Modification API will perform the action
  - `passwordAdapter` is the Password adapter used to authenticate the user. For example: `PasswordAdapter1`

  In the request body, you can send the `file` that you want to replace and `properties` metadata. At a minimum, either of these fields is mandatory.
  In the `file` element, select the Main Agreement file that you want to replace with. The Project Document Management Modification API supports only files of docx extension.
  Using the `properties` element in the request body, you can update attributes for the Main Agreement you have uploaded (header fields in the **Overview** tab):

  - `name`: You can change the title of the Main Agreement using this field. If this value is not specified, then the current version document name remains as the document title. The name specified for the Main Agreement must have the same extension as the uploaded file. For example, if the name specified is `Main_Agreement.docx` then the uploaded file must be of type `.docx`. The name specified for the Main Agreement is unique at the parent level.
  - `description`: Specify the description of the Main Agreement.
  - `editors`: Specify the editor of the Main Agreement. An editor can be an individual user, a global group or a project group. If the editor information is not provided or invalid, an appropriate error message will be returned.
    `editors` element contains the following fields:
    - `name`: Specify the name of the user or the group.
    - `uniqueName`: Specify the unique name of the user or the group. If `uniqueName` is null, then `editors` value is set to empty.
    - `passwordAdapter`: Specify the password adapter used to authenticate the user. This field is required only for user.
    - `type`: Specify the owner type. Valid values: `User`, `ProjectGroup`, `Group`
  - `useAs`: Valid values are:
    - `ContractContent`: If you specify this value, then only files of type `xlsx` are allowed.
      The Excel file will contain contract terms or pricing terms. However, if the `Parameter Application.ACM.EnableExcelXLSXContractContent` is disabled, then files of type `xlsx` are not allowed.

      > **Note**
      >
      > To download pricing terms in an `xls` file, contact SAP Ariba Support to enable the feature **SC-20703: Enable Old 2003 Excel Format for Source To Contract Pricing Term Excel file** and the parameter `Application.ACM.EnableExcelXLSXContractContent` for your site.

    - `ContractAttachment`
    - `None`

- `owner`: Specify the owner of the Main Agreement. An owner can be an individual user, a global group or a project group. If the owner information is not provided or invalid an appropriate error message will be returned.
  The `owner` element contains the following fields:
  - `name`: Specify the name of the user or the group.
  - `uniqueName`: Specify the unique name of the user or the group.
  - `passwordAdapter`: Specify the password adapter used to authenticate the user. This field is required only for user.
  - `type`: Specify the owner type. Valid values: `User`, `ProjectGroup`, `Group`
- `saveWithNewVersion`: If the value is set to `true`, a major version is created. For example, if v1 is the current version, then the new version will be v2.
  If the value is set to `false`, a minor version gets created. For example, if the current version is v1, then the new version will be v1(editing).
  If the key-value pair is not sent as part of the `properties` element, a minor version gets created.
- `versionComment`: Comment for the new version of the document.
- `publishRequired`: If the value is set to either `true` or `false`, values specified as part of the current version will be updated. If the key-value pair is not specified or if the value is empty, the current version value will not be updated.
- `accessControl`: Specify the access control applicable on the document. You can specify multiple values, delimited by commas. For example: `[ “Legal Information”, “Private to Team Members” ]`
- `autoFinalizeOnPublish`: If the value is set to `true`, then the Main Agreement is automatically finalized when the Contract Workspace (Procurement) containing the Main Agreement is published. If the value is set to `false`, then the Main Agreement is not automatically finalized when the Contract Workspace (Procurement) containing the Main Agreement is published. If the key-value pair is not specified or if the value is empty, the current version value will not be updated.

Use the following data model for the `properties` field:

```
{
    "name": "MainAgreement.docx",
    "description" : "Main Agreement document description",
    "editors": {
            "name": "test",
            "uniqueName": "customersupportadmin",
            "passwordAdapter": "PasswordAdapter1",
            "type": "User"
     },
    "useAs": "None",
    "owner": {
           "name": "test",
           "uniqueName": "customersupportadmin",
           "passwordAdapter": "PasswordAdapter1",
           "type": "User"
     },
    "saveWithNewVersion":true,
    "versionComment" : "document has been modified",
    "publishRequired": true,
    "accessControl": [
        "Legal Information",
        "Private To Team Members"
     ],
    "autoFinalizeOnPublish": false
}
```

The Project Document Management Modification API returns details of the updated Main Agreement. For example:

```
{
    "id": "Doc28548356",
    "name": "MainAggreement.docx",
    "size": 237180,
    "version": "14",
    "parentFolderId": "CW39412",
    "projectId": "CW39412",
    "owner": "customersupportadmin",
    "lastModifiedDate": "2022-07-04T12:48:16.010+0000",
    "creationDate": "2022-05-05T07:32:38.155+0000",
    "documentType": "Document",
    "active": true,
    "status": "Draft",
    "otherVersion": [
        {
            "id": "Doc31236598",
            "name": "MainAggreement.docx",
            "version": "13",
            "parentFolderId": "CW39412",
            "projectId": "CW39412",
            "owner": "customersupportadmin",
            "lastModifiedDate": "2022-07-04T12:39:57.572+0000",
            "creationDate": "2022-05-05T07:32:38.155+0000",
            "documentType": "Document",
            "active": true,
            "status": "Draft"
        },
        ...,
        {
            "id": "Doc28532774",
            "name": "Ice cream.docx",
            "version": "1",
            "parentFolderId": "CW39412",
            "projectId": "CW39412",
            "owner": "customersupportadmin",
            "lastModifiedDate": "2022-05-05T07:32:38.155+0000",
            "creationDate": "2022-05-05T07:32:38.155+0000",
            "documentType": "Document",
            "active": true,
            "status": "Draft"
        }
    ]
}
```

- To update a Contract Addendum, use the `PUT /projectDocuments/{documentId}` endpoint. Example:

```
PUT {{runtime_URL}}/projectDocuments/Doc12345678?
user=myUser&passwordAdapter=PasswordAdapter1&realm=mySite
```

Where:

- `realm` is your site name
- `user` is the User ID of the user for whom the Project Document Management Modification API will perform the action
- `passwordAdapter` is the password adapter used to authenticate the user. For example: `PasswordAdapter1`

In the request body, you can send the `file` that you want to replace and `properties` metadata. At a minimum, either of these fields is mandatory.

In the `file` element, select the Contract Addendum file that you want to replace with. The Project Document Management Modification API supports only files of `docx` extension.
Using the `properties` element in the request body, you can update attributes for the Contract Addendum you have uploaded (header fields in the **Overview** tab):

- `name`: You can change the title of the Contract Addendum using this field. If this value is not specified, then the current version document name remains as the document title. The name specified for the Contract Addendum must have the same extension as the uploaded file. For example, if the name specified is `Contract_Addendum.docx` then the uploaded file must be of type `.docx`. The name specified for the Contract Addendum is unique at the parent level.
- `description`: Specify the description of the Contract Addendum.
- `editors`: Specify the editor of the Contract Addendum. An editor can be an individual user, a global group or a project group. If the editor information is not provided or invalid, an appropriate error message will be returned.
  `editors` element contains the following fields:
  - `name`: Specify the name of the user or the group.
  - `uniqueName`: Specify the unique name of the user or the group. If `uniqueName` is null, then `editors` value is set to empty.
  - `passwordAdapter`: Specify the password adapter used to authenticate the user. This field is required only for user.
  - `type`: Specify the owner type. Valid values: `User`, `ProjectGroup`, `Group`
- `useAs`: Valid values are:
  - `ContractContent`: If you specify this value, then only files of type `xlsx` are allowed.
    The Excel file will contain contract terms or pricing terms. However, if the `Parameter Application.ACM.EnableExcelXLSXContractContent` is disabled, then files of type `xlsx` are not allowed.

    > ⓘ Note
    >
    > To download pricing terms in an `xls` file, contact SAP Ariba Support to enable the feature **SC-20703: Enable Old 2003 Excel Format for Source To Contract Pricing Term Excel file** and the parameter `Application.ACM.EnableExcelXLSXContractContent` for your site.

  - `ContractAttachment`
  - `None`
- `owner`: Specify the owner of the Contract Addendum. An owner can be an individual user, a global group or a project group. If the owner information is not provided or invalid an appropriate error message will be returned.
  The `owner` element contains the following fields:
  - `name`: Specify the name of the user or the group.
  - `uniqueName`: Specify the unique name of the user or the group.
  - `passwordAdapter`: Specify the password adapter used to authenticate the user. This field is required only for user.
  - `type`: Specify the owner type. Valid values: `User`, `ProjectGroup`, `Group`
- `saveWithNewVersion`: If the value is set to `true`, a major version is created. For example, if v1 is the current version, then the new version will be v2.
  If the value is set to `false`, a minor version gets created. For example, if the current version is v1, then the new version will be v1(editing).
  If the key-value pair is not sent as part of the `properties` element, a minor version gets created.

- `versionComment`: Specify comment for the new version of the document.
- `publishRequired`: If the value is set to either `true` or `false`, values specified as part of the current version will be updated. If the key-value pair is not specified or if the value is empty, the current version value will not be updated.
- `accessControl`: Specify the access control applicable on the document. You can specify multiple values, delimited by commas. For example: `[ “Legal Information”, “Private to Team Members” ]`
- `autoFinalizeOnPublish`: If the value is set to `true`, then the Contract Addendum is automatically finalized when the Contract Workspace (Procurement) containing the Contract Addendum is published. If the value is set to `false`, then the Contract Addendum is not automatically finalized when the Contract Workspace (Procurement) containing the Contract Addendum is published. If the key-value pair is not specified or if the value is empty, the current version value will not be updated.

Use the following data model for the `properties` field:

```
{
    "name": "ContractAddendum.docx",
    "description" : "Contract Addendum document description",
    "editors": {
            "name": "test",
            "uniqueName": "customersupportadmin",
            "passwordAdapter": "PasswordAdapter1",
            "type": "User"
     },
    "useAs": "None",
    "owner": {
           "name": "test",
           "uniqueName": "customersupportadmin",
           "passwordAdapter": "PasswordAdapter1",
           "type": "User"
     },
    "saveWithNewVersion":true,
    "versionComment" : "my version comment",
    "publishRequired": true,
    "accessControl": [
        "Legal Information",
        "Private To Team Members"
     ],
    "autoFinalizeOnPublish": false
}
```

The Project Document Management Modification API returns details of the updated Contract Addendum. For example:

```
{
    "id": "Doc28548356",
    "name": "ContractAddendum.docx",
    "size": 237180,
    "version": "15",
    "parentFolderId": "CW39412",
    "projectId": "CW39412",
    "owner": "customersupportadmin",
    "lastModifiedDate": "2022-07-04T12:59:54.205+0000",
    "creationDate": "2022-05-05T07:32:38.155+0000",
    "documentType": "Document",
    "active": true,
    "status": "Draft",
    "otherVersion": [
        {
            "id": "Doc31324674",
            "name": "MainAggreement.docx",
```

```
            "version": "14",
            "parentFolderId": "CW39412",
            "projectId": "CW39412",
            "owner": "customersupportadmin",
            "lastModifiedDate": "2022-07-04T12:48:16.010+0000",
            "creationDate": "2022-05-05T07:32:38.155+0000",
            "documentType": "Document",
            "active": true,
            "status": "Draft"
        },
        ...,
        {
            "id": "Doc28532774",
            "name": "Ice cream.docx",
            "version": "1",
            "parentFolderId": "CW39412",
            "projectId": "CW39412",
            "owner": "customersupportadmin",
            "lastModifiedDate": "2022-05-05T07:32:38.155+0000",
            "creationDate": "2022-05-05T07:32:38.155+0000",
            "documentType": "Document",
            "active": true,
            "status": "Draft"
        }
    ]
}
```

- To update a folder, use the `PUT /projectDocuments/{documentId}` endpoint.
  Example:

  ```
  PUT {{runtime_URL}}/projectDocuments/WS12345?
  user=myUser&passwordAdapter=PasswordAdapter1&realm=mySite
  ```

  Where:

  - `realm` is your site name
  - `user` is the User ID of the user for whom the Project Document Management Modification API will perform the action
  - `passwordAdapter` is the Password adapter used to authenticate the user. For example: `PasswordAdapter1`

  In the request body, specify `properties` attributes for the folder you want to update:

  - `name`: You can change the title of the folder using this field. If this value is not specified, then the current version folder name remains as the folder title. The name specified for the folder is unique at the parent level.
  - `description`: Specify the description of the folder.
  - `owner`: Specify the owner of the folder. An owner can be an individual user, a global group or a project group. If the owner information is not provided or invalid an appropriate error message will be returned.
    The `owner` element contains the following fields:
    - `name`: Specify the name of the user or the group.
    - `uniqueName`: Specify the unique name of the user or the group.
    - `passwordAdapter`: Specify the password adapter used to authenticate the user. This field is required only for user.
    - `type`: Specify the owner type. Valid values: `User`, `ProjectGroup`, `Group`

- `accessControl`: Specify the access control applicable on the document. You can specify multiple values, delimited by commas. For example: `[ “Legal Information”, “Private to Team Members” ]`
- `saveWithNewVersion`: If the value is set to `true`, a major version is created. For example, if v1 is the current version, then the new version will be v2.
  If the value is set to `false`, a minor version gets created. For example, if the current version is v1, then the new version will be v1(editing).
  If the key-value pair is not sent as part of the `properties` element, a minor version gets created.
- `versionComment`: Specify comment for the new version of the folder.

Use the following data model for the `properties` field:

```
{
    "name": "MyFolder",
    "description" : "Folder description",
    "owner": {
           "name": "test",
           "uniqueName": "customersupportadmin",
           "passwordAdapter": "PasswordAdapter1",
           "type": "User"
     },
    "saveWithNewVersion":true,
    "versionComment" : "my version comment",
    "accessControl": [
        "Legal Information",
        "Private To Team Members"
     ]
}
```

The Project Document Management Modification API returns details of the updated folder.
For example:

```
{
    "id": "Doc28548356",
    "name": "MyFolder",
    "size": 237180,
    "version": "8",
    "parentFolderId": "CW39412",
    "projectId": "CW39412",
    "owner": "customersupportadmin",
    "lastModifiedDate": "2022-06-17T06:18:09.966+0000",
    "creationDate": "2022-05-05T07:32:38.155+0000",
    "documentType": "Document",
    "active": true,
    "status": "Draft",
    "otherVersion": [
        {
            "id": "Doc30819101",
            "name": "ContractAddendum.docx",
            "version": "7",
            "parentFolderId": "CW39412",
            "projectId": "CW39412",
            "owner": "customersupportadmin",
            "lastModifiedDate": "2022-06-17T06:11:34.671+0000",
            "creationDate": "2022-05-05T07:32:38.155+0000",
            "documentType": "Document",
            "active": true,
            "status": "Draft"
        },
        ...,
        {
            "id": "Doc28532774",
            "name": "Ice cream.docx",
```

```
            "version": "1",
            "parentFolderId": "CW39412",
            "projectId": "CW39412",
            "owner": "customersupportadmin",
            "lastModifiedDate": "2022-05-05T07:32:38.155+0000",
            "creationDate": "2022-05-05T07:32:38.155+0000",
            "documentType": "Document",
            "active": true,
            "status": "Draft"
        }
    ]
}
```


To troubleshoot your queries to the Project Document Management Modification API, refer to Error Messages Returned by the Project Document Management API [page 35].

Changing the State of a Document in a Contract Workspace

This topic explains how you can use the Project Document Management State Change API to modify the state of documents such as Main Agreement, Contract Addendum, and standard documents in a contract workspace.

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.
- Ensure that the user whose credentials are passed in the API call is a project owner or a team member.

Context

You can use the `POST /projectDocuments/documentID/state` endpoint to change the state of a document in the contract workspace. The following table lists the documents and supported state change options:

Document Type	Current State	New State
Main Agreement Contract addendum	Draft	Publish
	Publish	Finalize
	Finalize	MakeDraft
Standard documents	Draft	Publish

> **Note**
>
> The Project Document Management State Change API does not modify the state of documents in contract workspaces that are in Published, Put on Hold, Closed, or Expired state. Information for suppliers

> **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

To change the state of a document use the `POST /projectDocuments/`*`documentID`*`/state` endpoint as follows:

```
POST {{runtime_url}}/projectDocuments/documentID/state?
user=username&passwordAdapter=PasswordAdapterName&realm=realm
Request body:
{
  "action": "action"
}
```

Where

`username` is the name of the user whose credentials are used to authenticate the API request.

`passwordadapter` is the password adapter used to authenticate the user specified in `user`.

`realm` `is the name of your site.`

`documentID` is the ID of the document for which you want to modify the state.

In the `Request body`, specify the new state you want to apply to the document. Supported values are:

`Publish`
`Finalize`
`MakeDraft`

You receive one of the following messages:

- `204` to indicate successful state change.
- `400` to indicate a bad request.
- `401` to indicate an unauthorized request.
- `403` to indicate a forbidden request.
- `404` to indicate that the specified document was not found.

Interpretation of Project Document Management API Responses

This topic explains how you can interpret the responses from the Project Document Management API.

The `GET /projectDocuments` and `GET /projectDocuments/documentId` endpoints return the following details of the documents that match the specified search criteria:

> **Note**
>
> For the `GET /projectDocuments` endpoint, the details in the response are controlled by the options specified for the `$select` parameter in the query.

```
{
id    string
name    string
size    integer
version    string
parentFolderId    string
projectId    string
owner    string
lastModifiedDate    string
creationDate    string
documentType    string
active    string
status    string
otherVersion    [...]
childrenDocument    [...]

}
```

Field	Description
`id`	Contains the ID of the document, folder, project, or workspace.
`name`	Contains the name of document, folder, project, or workspace.
`size`	Indicates the size of the file in bytes.
`version`	Indicates the version of the file.
`parentFolderId`	Contains the ID of the parent folder.
`projectId`	Contains the ID of the sourcing project or workspace.
`owner`	Contains the user name of the document owner.
`lastModifiedDate`	Contains the last modified date of the document or folder.
`creationDate`	Contains the date the document or folder was created.
`documentType`	Indicates the type of the document.
`active`	Indicates whether the document is active or not.
`status`	Contains the status of the document.
`otherVersion`	Indicates whether the document has other versions, and if yes, provides details of that.

Field	Description
`childrenDocument`	Contains details of the child documents.

You can use a combination these fields to understand the folder structure and relationship between various elements and documents.

The following sections provide some example outputs to help you understand different responses in various use cases.

For a Header-Level Document


```
{
  "id": "Doc12739754",
  "name": "Document.docx",
  "size": 16049,
  "version": "1",
  "parentFolderId": "CW27570",
  "projectId": "CW27570",
  "owner": "customersupportadmin",
  "lastModifiedDate": "2020-04-24T08:54:56.882+0000",
  "creationDate": "2020-04-24T08:54:56.882+0000",
  "documentType": "Document",
  "active": true,
  "status": "Draft",
  "otherVersion": []
}
```


For a Document That Is Within a Folder


```
{
  "id": "WS12739767",
  "name": "Document Inside A Folder",
  "version": "1",
  "parentFolderId": "CW27570",
  "projectId": "CW27570",
  "owner": "customersupportadmin",
  "lastModifiedDate": "2020-04-24T08:59:09.843+0000",
  "creationDate": "2020-04-24T08:57:36.966+0000",
  "documentType": "Folder",
  "active": true,
  "childrenDocument": [
    {
      "id": "Doc12739788",
      "name": "Document.docx",
      "size": 16049,
      "version": "1",
      "parentFolderId": "WS12739767",
      "projectId": "CW27570",
      "owner": "customersupportadmin",
      "lastModifiedDate": "2020-04-24T08:59:09.843+0000",
      "creationDate": "2020-04-24T08:59:09.843+0000",
      "documentType": "Document",
      "active": true,
      "status": "Draft",
```

```
      "otherVersion": []
    }
  ]
}
```

For a Document Within a Subfolder

```
{
  "id": "WS12739814",
  "name": "Parent Folder",
  "version": "1",
  "parentFolderId": "CW27570",
  "projectId": "CW27570",
  "owner": "customersupportadmin",
  "lastModifiedDate": "2020-04-24T09:04:51.904+0000",
  "creationDate": "2020-04-24T09:03:31.754+0000",
  "documentType": "Folder",
  "active": true,
  "childrenDocument": [
    {
      "id": "WS12739824",
      "name": "Child Folder 1",
      "version": "1",
      "parentFolderId": "WS12739814",
      "projectId": "CW27570",
      "owner": "customersupportadmin",
      "lastModifiedDate": "2020-04-24T09:05:03.534+0000",
      "creationDate": "2020-04-24T09:04:40.772+0000",
      "documentType": "Folder",
     "active": true,
      "childrenDocument": [
        {
          "id": "Doc12739825",
          "name": "Document.docx",
          "size": 16049,
          "version": "1",
          "parentFolderId": "WS12739824",
          "projectId": "CW27570",
          "owner": "customersupportadmin",
          "lastModifiedDate": "2020-04-24T09:05:03.534+0000",
          "creationDate": "2020-04-24T09:05:03.534+0000",
          "documentType": "Document",
          "active": true,
          "status": "Draft",
          "otherVersion": []
        }
      ]
    }
  ]
}
```

For a Document in a Subfolder of a Subfolder

```
{
  "id": "WS12740572",
  "name": "Parent Folder",
  "version": "1",
```

```
  "parentFolderId": "CW27570",
  "projectId": "CW27570",
  "owner": "customersupportadmin",
  "lastModifiedDate": "2020-04-24T09:14:28.354+0000",
  "creationDate": "2020-04-24T09:13:45.146+0000",
  "documentType": "Folder",
  "active": true,
  "childrenDocument": [
    {
      "id": "WS12740576",
      "name": "Level 1",
      "version": "2",
      "parentFolderId": "WS12740572",
      "projectId": "CW27570",
      "owner": "customersupportadmin",
      "lastModifiedDate": "2020-04-24T09:15:08.073+0000",
      "creationDate": "2020-04-24T09:14:16.520+0000",
      "documentType": "Folder",
      "active": true,
      "childrenDocument": [
        {
          "id": "WS12740580",
          "name": "Sub Level 1",
          "version": "1",
          "parentFolderId": "WS12740576",
          "projectId": "CW27570",
          "owner": "customersupportadmin",
          "lastModifiedDate": "2020-04-24T09:15:32.343+0000",
          "creationDate": "2020-04-24T09:14:58.883+0000",
          "documentType": "Folder",
          "active": true,
          "childrenDocument": [
            {
              "id": "WS12740582",
              "name": "Sub Level 2",
              "version": "1",
              "parentFolderId": "WS12740580",
              "projectId": "CW27570",
              "owner": "customersupportadmin",
              "lastModifiedDate": "2020-04-24T09:15:44.962+0000",
              "creationDate": "2020-04-24T09:15:20.426+0000",
              "documentType": "Folder",
              "active": true,
              "childrenDocument": [
                {
                  "id": "Doc12740583",
                  "name": "MainAgreement.docx",
                  "size": 59206,
                  "version": "1",
                  "parentFolderId": "WS12740582",
                  "projectId": "CW27570",
                  "owner": "customersupportadmin",
                  "lastModifiedDate": "2020-04-24T09:15:44.963+0000",
                  "creationDate": "2020-04-24T09:15:44.963+0000",
                  "documentType": "Main Agreement",
                  "active": true,
                  "status": "Draft",
                  "otherVersion": []
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

For a Document That Has a Task Attached to It

```
{
  "id": "Doc12740556",
  "name": "OpenAPI Contract Addendum.docx",
  "size": 15257,
  "version": "1",
  "parentFolderId": "CW27570",
  "projectId": "CW27570",
  "owner": "customersupportadmin",
  "lastModifiedDate": "2020-04-24T09:07:30.697+0000",
  "creationDate": "2020-04-24T09:07:30.697+0000",
  "documentType": "Document",
  "active": true,
  "status": "Draft",
  "otherVersion": []
}
```

Error Messages Returned by the Project Document Management API

HTTP Response Codes

HTTP Response Code	HTTP Message	Additional Information
400	Bad Request. One or more parameters specified in the request are incorrect.	Double check the spelling of your request parameter, then resubmit. For more information, refer to Identifying and Downloading Project Documents by Using the Project Document Management API [page 7].

HTTP Response Code	HTTP Message	Additional Information
400	Bad request.	Double check the spelling of your request parameter, then resubmit. For more information, refer to Creating a Document or Folder by Using the Project Document Management Creation API [page 10] and Updating Documents by Using the Project Document Management Modification API [page 18].
401	Unauthorized request. You are using an access token that is expired or incorrect.	
401	Unauthorized	For more information, refer to Creating a Document or Folder by Using the Project Document Management Creation API [page 10] and Updating Documents by Using the Project Document Management Modification API [page 18].
403	Forbidden	
404	No matching entries found.	For more information, refer to Identifying and Downloading Project Documents by Using the Project Document Management API [page 7].
404	Not found	For more information, refer to Updating Documents by Using the Project Document Management Modification API [page 18].

Document and Project Types Supported by the Project Document Management API

The following table lists the document types supported by the endpoints of the Project Document Management API.

	Supported Actions			
Document Type	**Search for**	**View Properties of**	**Download**	**Download Signed Document**
ContractAddendum	Yes	Yes	Yes	Yes
MainAgreement	Yes	Yes	Yes	Yes
CleansheetDocument	Yes	Yes	No	No
LineItemDocument	Yes	Yes	No	Yes
ContentDocument	Yes	Yes	No	No
Event	Yes	Yes	No	No
SavingsForm	Yes	Yes	No	No
AssembledPDF	Yes	Yes	Yes	Yes
AnalyticalReport	Yes	Yes	No	No
CompoundReport	Yes	Yes	No	No
ContractClause	Yes	Yes	No	No
ProfileQuestionnaire	Yes	Yes	No	No
Scorecard	Yes	Yes	No	No
ContractTerms	Yes	Yes	No	No
Documents (for example, attachments)	Yes	Yes	Yes	Yes
Documents in a folder	Yes	Yes	Yes	Yes

The following table lists the project types supported by the endpoints of the Project Document Management API.

	Supported Actions		
Project Type	**Search for Documents**	**View Properties of Documents**	**Download Documents**
Knowledge Project	Yes	Yes	Yes

	Supported Actions		
Project Type	**Search for Documents**	**View Properties of Documents**	**Download Documents**
Contract Request (Procurement)	Yes	Yes	Yes
Contract Workspace (Procurement)	Yes	Yes	Yes
Contract Workspace (Internal)	Yes	Yes	Yes
Contract Request (Sales)	Yes	Yes	Yes
Contract Workspace (Sales)	Yes	Yes	Yes
Supplier Qualification Management Project	Yes	Yes	Yes
Supplier Performance Management Project	Yes	Yes	Yes
Sourcing Project	Yes	Yes	Yes
Sourcing Request	Yes	Yes	Yes
Supplier Registration	Yes	Yes	Yes
Supplier Request	Yes	Yes	Yes
Supplier Qualification	Yes	Yes	Yes
Process Flow	Yes	Yes	Yes
SM Modular Questionnaire	Yes	Yes	Yes
Supplier Disqualification Project	Yes	Yes	Yes
Quick Survey	Yes	Yes	No

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.

www.ariba.com

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.