

PUBLIC

Document Version: 2608 – 2026-08

Contract Workspace Management APIs

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

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

Content

Contract Workspace Management APIs. 4
Endpoints Included in the Contract Workspace Management APIs. 6
Workflow for the Contract Workspace Management APIs. 10
Creating Contract Workspaces Using the Contract Workspace Modification API. 13
Create Contract Workspaces by Copying Existing Contracts Using the Contract Workspace Modification API. 19
Changing the State of Procurement Contract Workspace Using the Contract Workspace Modification API. 21
Updating Contract Workspaces Using the Contract Workspace Modification API. 27
Changing the State of a Contract Workspace by Using the Contract Workspace State Change API. 33
Creating and Reviewing Contract Terms by Using the Contract Term Management API. 35
Synchronizing Contract Line Item Data of a Contract Workspace With Linked Compliance Contract Using Contract Terms Management API. 37
Getting Contract Workspaces Matching the Provided ID Using the Contract Workspace Retrieval API 38
Searching for Contract Workspaces Using the Contract Workspace Retrieval API. 39
Contract Workspace Attributes Eligible for Specifying in the Select Clause to Search for Contract Workspaces. 43
Getting the Metadata of a Contract Line Items Document Using the Contract Line Items Document Retrieval API. 44
Downloading a Contract Line Items Document Using the Contract Line Items Document Download API. 45
Creating a Contract Line Items Document in a Contract Workspace using the Contract Line Items Document Management API. 46
Updating a Contract Line Items Document Using the Contract Line Items Document Management API 48
Publishing Contract Line Items Document by Using the Contract Line Items Document Publish API 50
Importing Line Items to a Contract Line Items Document Using the Contract Line Items Import API 52
Deleting or Deactivating Line Items in a Contract Line Items Document Using the Contract Line Items Import API. 53
Importing Deactivated Line Items From an Integrated ERP System to a Contract Workspace Using Contract Line Items Import API. 56
Deleting a Contract Workspace Using the Contract Workspace Delete API. 58
Creating a Baseline Contract in SAP Ariba Contracts Using the Contract Workspace Modification API 59
Error Messages Returned by the Contract Workspace Management APIs. 65

API-Specific Disclaimers and Legal Information. 67

Contract Workspace Management APIs

This guide is for a customer developer intending to implement the Contract Workspace Management APIs in a client application to:

- Create and modify procurement contract workspaces and their header information.
- Create and retrieve information about contract terms and contract requests.
- Retrieve the metadata (header fields) of a single procurement contract workspace and search for procurement contract workspaces by providing a search and filter criteria.
- Retrieve the metadata of a Contract Line Items Document (CLID) in a procurement contract workspace.
- Download the CLID from a contract workspace in the Excel file format.
- Create and modify a CLID.
- Import line item information to the CLID of a contract workspace.

The Contract Workspace Management APIs comprises of the following APIs:

- The **Contract Workspace Modification** API, which enables buyers to perform the following tasks:
  - Create a procurement contract workspace by using an existing template.
  - Specify details of the integrated system while creating or modifying a contract workspace.
  - Update the record ID, status, and the link to the related record on the integrated system.
  - Update the metadata (header fields) of a single contract workspace.
- The **Contract Terms Management** API, which enables buyers to perform the following tasks:
  - Create contract terms documents in the contract workspace and corresponding contract requests in SAP Ariba Procurement solutions for suite-integrated solutions.
    Suite integration enables integration between procurement or invoicing solutions from SAP Ariba Procurement solutions and SAP Ariba strategic sourcing solutions such as SAP Ariba Contracts.
  - Retrieve details such as the contract term ID, contract request ID, contract ID, and so on by passing a contract workspace ID.
  - Develop solutions that create and manage contracts in both procurement and sourcing solutions by implementing the **Contract Terms Management** API along with **Contract Compliance** API and the **Contract Workspace Modification** API.
  - Synchronize the contract line item information of a contract workspace with the linked compliance contract by specifying the IDs of the contract workspace and the contract terms document.
- The **Contract Workspace Retrieval** API, which enables buyers to perform the following tasks:
  - Retrieve the metadata (header fields) of a single contract workspace.
  - Search for contract workspaces using several attributes such as the contract title, status, last modified date, integrated system's name, and so on.
- The **Contract Line Items Document Retrieval** API, which enables buyers to get the properties and metadata of a CLID in a procurement contract workspace.
- The **Contract Line Items Document Download** API, which enables buyers to download the CLID from a contract workspace in the Excel file format.
- The **Contract Line Items Document Management** API, which enables buyers to create and modify a CLID.
- The **Contract Line Items Document Publish API**, which enables buyers to publish a contract line items document (CLID) from a contract workspace (procurement).

- The **Contract Line Items Document Import** API, which enables buyers to import line item information to the CLID of a contract workspace.
- The **Contract Workspace Delete API**, which enables buyers to delete a procurement contract workspace that is in the **Draft** state.

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.
- Users who manage the contract workspace information must have membership of the **Contract Administrator** group.

Restrictions

The **Contract Terms Management** API does not support:

- Contract Line Item Documents (CLID)
- Combined spend contracts
- Custom fields or custom terms

Related Guides

For more information about the procurement contract workspace integration functionality for SAP ERP-integrated sites, see the Managing projects, teams, documents, and tasks guide.

Endpoints Included in the Contract Workspace Management APIs

This topic describes the different endpoints supported by each API within the Contract Workspace Management APIs.

Endpoints Included in the Contract Workspace Modification API

The Contract Workspace Modification API consists of the following endpoints:

Endpoint	Description
`POST /contractworkspaces`	Creates a contract workspace using the ContractWorkspace Model Object that contains Contract Header fields metadata. For more information, refer to Creating Contract Workspaces Using the Contract Workspace Modification API [page 13].
`POST /contractWorkspaces/{contractId}/state`	Updates the status of a published procurement contract workspace to `DraftAmendment` (`action` = `amend`) so that you can modify the contract workspace. You can use the same endpoint with the `action` parameter set to `complete amendment` to trigger publishing for **Administrative** or **PriceUpdate** amendments. For more information, refer to Changing the State of Procurement Contract Workspace Using the Contract Workspace Modification API [page 21].
`PATCH /contractworkspaces`	Updates an existing contract workspace by using the provided contract workspace fields for updating the contract. For more information, refer to Updating Contract Workspaces Using the Contract Workspace Modification API [page 27].

Endpoints Included in the Contract Terms Management API

The Contract Terms Management APIs consists of the following endpoints:

Endpoint	Description
`POST /contractWorkspaces/{contractId}/contractTerms`	Creates a contract term in the specified contract workspace and a contract request for compliance contracts in the specified buyer realm. For additional information, refer to Creating and Reviewing Contract Terms by Using the Contract Term Management API [page 35].

Endpoint	Description
`GET /contractWorkspaces/{contractId}/contractTerms`	Retrieves details including contract term ID, contract request ID, and contract ID for the specified contract workspace ID. For additional information, refer to Creating and Reviewing Contract Terms by Using the Contract Term Management API [page 35].
`POST /contractWorkspaces/{contractId}/contractTerms/{contractTermId}/state`	Synchronizes the contract line item information of a contract workspace with the linked compliance contract by specifying the IDs of the contract workspace and the contract terms document. If the request is successful, the API initiates an asynchronous job and returns the job ID for reference. For additional information, refer to Synchronizing Contract Line Item Data of a Contract Workspace With Linked Compliance Contract Using Contract Terms Management API [page 37].

Endpoints Included in the Contract Workspace Retrieval API

The Contract Workspace Retrieval API consists of the following endpoints:

Endpoint	Description
`GET /contractWorkspaces`	Gets contract workspaces matching the provided criteria. For more information, refer to Searching for Contract Workspaces Using the Contract Workspace Retrieval API [page 39].
`GET /contractWorkspaces/$contractId`	Returns the single contract workspace matching the provided contract workspace ID. For more information, refer to Getting Contract Workspaces Matching the Provided ID Using the Contract Workspace Retrieval API [page 38].

Endpoints Included in the Contract Line Items Document Retrieval API

The Contract Line Items Document Retrieval API consists of the following endpoints:

Endpoint	Description
`GET /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}`	Retrieves the document and the metadata associated with the specified document ID and the contract ID. The metadata includes all the standard fields, custom fields, task IDs, task names, line item types (including hierarchical line items and line items with pricing conditions), and translations of line items in the CLID. The API also retrieves all the versions of the CLID excluding the version that is in the **Not Created** status. For more information, refer to Getting the Metadata of a Contract Line Items Document Using the Contract Line Items Document Retrieval API [page 44].

Endpoints Included in the Contract Line Items Document Download API

The Contract Line Items Document Download API consists of the following endpoints:

Endpoint	Description
`GET /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}/files`	Downloads the CLID corresponding to the specified document ID and contract workspace ID. For more information, refer to Downloading a Contract Line Items Document Using the Contract Line Items Document Download API [page 45].

Endpoints Included in the Contract Line Items Document Management API

The Contract Line Items Document Management API consists of the following endpoints:

Endpoint	Description
`POST /contractWorkspaces/{contractId}/contractLineItemDocuments`	Creates a CLID in the specified contract workspace. For more information, refer to Creating a Contract Line Items Document in a Contract Workspace using the Contract Line Items Document Management API [page 46].
`PATCH /contractWorkspaces/{contractId}/contractLineItemDocuments/{id}`	Modifies the attributes of an existing CLID in the specified contract workspace. For more information, refer to Updating a Contract Line Items Document Using the Contract Line Items Document Management API [page 48].

Endpoints Included in the Contract Line Items Document Publish API

The Contract Line Items Document Publish API consists of the following endpoints:

Endpoint	Description
`POST /contractWorkspaces/{contractId}/contractLineItemDocuments/{id}/state`	Publishes the contract line items document (CLID) associated with a procurement contract workspace. For more information, refer to Publishing Contract Line Items Document by Using the Contract Line Items Document Publish API [page 50].

Endpoints Included in the Contract Line Items Document Import API

The Contract Line Items Document Import API consists of the following endpoints:

Endpoint	Description
`POST /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}/files`	Imports the line item data to the CLID of a contract workspace. For more information, refer to Importing Line Items to a Contract Line Items Document Using the Contract Line Items Import API [page 52].
	Deletes or Deactivates Line Items in a Contract Line Items Document of a contract workspace. For more information, refer to Deleting or Deactivating Line Items in a Contract Line Items Document Using the Contract Line Items Import API [page 53].
	Imports Deactivated Line Items From an Integrated ERP System to a contract workspace. For more information, refer to Importing Deactivated Line Items From an Integrated ERP System to a Contract Workspace Using Contract Line Items Import API [page 56].

Endpoints Included in the Contract Workspace Delete API

The Contract Workspace Delete API consists of the following endpoints:

Endpoint	Description
`DELETE /contractWorkspaces/{contractId}`	Deletes a contract workspace in SAP Ariba Contracts. For more information, refer to Deleting a Contract Workspace Using the Contract Workspace Delete API [page 58].

Workflow for the Contract Workspace Management APIs

This topic describes the workflow of the different APIs within the Contract Workspace Management APIs.

Workflow for Contract Workspace Modification API

1. A contract administrator (such as the customer support administrator or super user) ensures that all the parameters related to procurement contract workspace creation are enabled.
2. The contract administrator verifies whether a procurement contract workspace template exists.

   > ⓘ Note
   >
   > For information about how to create project templates, see Creating project templates in Project template guide. Available template IDs are displayed on the dashboard under **Manage** > **Templates**.

3. Depending on whether the contract workspace template exists, the client application either creates a contract workspace with a system template ID or updates an existing contract workspace, as follows:
   - To create a contract workspace with a system template ID, the client application uses the `POST /contractworkspaces` endpoint, providing the system template ID. The template may contain all the details related to tasks, documents, teams, and references to integrated systems.
   - To update an existing contract workspace:
     1. If the contract workspace is in published state, the client application changes the state of the contract workspace to amend by using the `POST /contractworkspace/contractId/state` endpoint. The client application provides the contract workspace ID as *`contractID`*.
     2. The client application uses the `PATCH /contractworkspaces` endpoint to update the contract workspace. The client application provides the contract workspace ID and the updated contract workspace fields for updating the contract, such as OOTB (out of the box) header fields and custom header fields, as applicable to the specified amendment type, in the body of the API request.

        > ⓘ Note
        >
        > The out-of-the-box (OOTB) field **Notice Period (in days)** is currently not supported by the Contract Workspace Modification API. You can update or modify this field directly through the SAP Ariba solution's user interface (UI).

     3. If the updated contract workspace was in published state and the state was changed to amend to enable the updates:
        - If the type of amendment is **Administrative** or **PriceUpdate**, the client application changes the state of the contract workspace to amend complete by using the `POST /contractworkspace/contractId/state` endpoint. The client application provides the contract workspace ID as *`contractID`*.
        - If the type of amendment is **Amendment**, **Renewal**, or **Termination**, users manually republish the contract.

Workflow for the Contract Terms Management API

1. Buyer user or the client application creates a contract workspace in SAP Ariba Contracts or SAP Ariba strategic sourcing solutions. Sourcing users can either create the contract workspace from the UI or configure the client application to create the workspace by using the Contract Workspace Modification API.
2. The client application creates a contract terms document in the contract workspace and a contract request in SAP Ariba Procurement solutions.
3. Buyer user completes the approval workflow for the contract request in SAP Ariba Procurement solutions to generate the contract ID. You can complete the approval from the SAP Ariba Procurement solutions UI.
4. (Optional) The client application retrieves information about the contract terms document and the compliance contract.
5. Buyer users can synchronize the contract line item information of a contract workspace with the linked compliance contract by specifying the IDs of the contract workspace and the contract terms document.

Workflow for the Contract Workspace Retrieval API

1. The sourcing user or the client application creates a contract workspace containing the Main Agreement and supporting contract documents for a contract with a supplier.
2. You can search for a single contract workspace details or search for procurement contract workspaces meeting certain criteria.
   - Use the `GET /contractworkspaces$contractId` endpoint to request the contract workspace matching the provided contract workspace ID.
   - Use the `GET /contractworkspaces` endpoint to retrieve a list of contract workspaces matching the search criteria. You can further filter the workspaces based on their attributes.

Workflow for the Contract Line Items Document Retrieval API

The client application gets the properties and metadata of a Contract Line Items Document (CLID) in a procurement contract workspace using the `GET /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}` endpoint. The contract workspace ID is specified in *`contractId`* and the ID of the CLID is specified in *`documentId`*.

Workflow for the Contract Line Items Document Download API

The client application downloads the Contract Line Items Document (CLID) using the `GET /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}/files` endpoint. The contract workspace ID is specified in *`contractId`* and the ID of the CLID is specified in *`documentId`*.

Workflow for the Contract Line Items Document Management API

1. To create a Contract Line Items Document (CLID) in a contract workspace in SAP Ariba Contracts, the client application uses the `POST /contractWorkspaces/{contractId}/contractLineItemDocuments` endpoint. The contract workspace ID is specified in *`contractId`*.
2. To modify a Contract Line Items Document (CLID) in a contract workspace in SAP Ariba Contracts, the client application uses the `PATCH /contractWorkspaces/{contractId}/contractLineItemDocuments/{id}` endpoint. The contract workspace ID is specified in *`contractId`* and the ID of the CLID is specified in *`id`*.

Workflow for the Contract Line Items Document Publish API

The client application publishes the contract line items document (CLID) associated with a contract workspace (procurement) in SAP Ariba Contracts using the `POST /contractWorkspaces/{contractId}/contractLineItemDocuments/{id}/state` endpoint. The contract workspace ID is specified in *`contractID`* and the ID of the CLID is specified in *`id`*.

Workflow for the Contract Line Items Document Import API

The client application uses the `POST /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}/files` endpoint to perform the following tasks.:

- Import the line item data to the CLID of a contract workspace in SAP Ariba Contracts.
- Delete or Deactivate Line Items in a Contract Line Items Document of a contract workspace.
- Import Deactivated Line Items From an Integrated ERP System to a contract workspace.

The contract workspace ID is specified in *`contractId`* and the ID of the CLID is specified in *`documentId`*.

Workflow for the Contract Workspace Delete API

The client application deletes a contract workspace in SAP Ariba Contracts using the `DELETE /contractWorkspaces/{contractId}` endpoint. The contract workspace ID is specified in the *`contractId`*.

Creating Contract Workspaces Using the Contract Workspace Modification API

Use this procedure to create a contract workspace. The Contract Workspace Modification API can be implemented in a client application to create procurement contract workspaces in SAP Ariba Contracts.

Prerequisites

If you have a system template ID that references a template containing all the details related to tasks, documents, and teams, you can use that template ID to create a procurement contract workspace. Available template IDs are displayed on the dashboard under **Manage** > **Templates**.

Context

The Contract Workspace Modification API enables you to create a contract workspace in SAP Ariba Contracts using the `POST /contractworkspaces` endpoint. You can create SAP Ariba Contracts projects (contract requests or contract workspaces) using project templates that can be pre-populated with team members, documents, tasks, references to integrated systems, and other project items. The available templates can change depending on the project information (attributes) you specify. Project templates can also contain conditions that determine the documents, tasks, and phases created according to the project attributes.

> **Note**
>
> If you have integrated SAP Ariba Contracts with SAP Ariba Sourcing and have guided sourcing enabled, you can include predecessor project information in the API query body. However, to enable the Contract Workspace Modification API to identify the predecessor event information, in addition to the predecessor project information, of the contract whenever applicable, the **Enable predecessor event identification for contracts created from guided sourcing projects** (`IdentifyPredecessorEventforContractsCreatedFromGuidedSourcingProjects`) ICM parameter must be enabled. For more information, refer to Enable predecessor event identification for contracts created from guided sourcing projects.

You can create a query body by specifying a project template ID and the properties required to be populated in the contract workspace.

> **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. On the dashboard, open **Manage** > **Templates** to view available template IDs.
2. If the contract workspace does not already exist, construct a query body that can be used in the request to create the contract workspace. Many fields are populated using the domain body parameter, as follows:

   ### Sample Code

   ```
   "domain":"<field_name>",
   "value":"<field_value>"
   ```

3. Invoke the `POST /contractworkspaces` endpoint to create a contract workspace with a system template ID. The template contains all the details related to tasks, documents, teams, and references to integrated systems.

   Request type: POST

   URL: **`{{runtime_url}}/contractworkspaces`**

   The following example creates a procurement contract workspace with the specified details:

   ### Example

   Request URL: `POST {{runtime_url}}/contractWorkspaces?user=adavis&passwordAdapter=PasswordAdapter1&realm=s4All-7`

   Query Body:

   ```
   {
     "title": "Test Search With Spaces",
     "description": "Demo Description",

     "isTestProject": true,

     "hierarchicalType" :"MasterAgreement",
     "parentAgreement":"",
     "templateId": "SYS0052",

     "supplier": {
       "name": "Apex Corporation",
       "systemID": "sid509",
       "organizationIDs": [
         {
           "domain": "internalsystemid",
           "value": "1009"
         },
         {
           "domain": "duns",
           "value": "700000080"
         },
         {
           "domain": "networkid",
           "value": "an70000000080"
         },
         {
           "domain": "oracle",
           "value": "4009"
         },
         {
           "domain": "psoft",
           "value": "share:usa0000022"
         },
         {
   ```

```
        "domain": "sap",
        "value": "0000001050"
      },
      {
        "domain": "buyersystemid",
        "value": "sid509"
      },
      {
        "domain": "sourcingnetworkid",
        "value": "an70000000080"
      }
    ],
    "address": {
      "name": "",
      "uniqueName": "sid509",
      "phone": "",
      "fax": "",
      "lines": [
        "753 Apex Avenue"
      ],
      "city": "Marlborough",
      "state": "MA",
      "postalCode": "1752",
      "country": "US"
    }
  },
  "affectedParties": [
        {
          "name": "Apex Corporation",
          "systemID": "sid509",
          "smVendorID": null,
          "organizationIDs": [
            {
              "domain": "internalsystemid",
              "value": "1009"
            },
            {
              "domain": "duns",
              "value": "700000080"
            },
            {
              "domain": "networkid",
              "value": "an70000000080"
            },
            {
              "domain": "oracle",
              "value": "4009"
            },
            {
              "domain": "psoft",
              "value": "share:usa0000022"
            },
            {
              "domain": "buyersystemid",
              "value": "sid509"
            },
            {
              "domain": "sourcingnetworkid",
              "value": "an70000000080"
            },
            {
              "domain": "sap",
              "value": "2000003"
            }
          ],
          "address": {
            "name": "",
            "uniqueName": "sid509",
```

```
            "phone": "",
            "fax": "",
            "lines": [
              "753 Apex Avenue"
            ],
            "city": "Marlborough",
            "state": "MA",
            "postalCode": "1752",
            "country": "US"
          }
        }
      ],


    "contractAmount": {
    "amount": 1000,
    "currency": "Euro"
  },

  "proposedContractAmount":
  {
      "amount": 1000,
    "currency": "Euro"
  },

      "commodities": [
    {
      "uniqueName": "56",
      "domain": "unspsc",
      "name": "Furniture and Furnishings"
    }
  ],

  "regions": [
    {
      "uniqueName": "APAC",
      "name": "Asia Pacific"
    }
  ],

  "departments": [
    {
      "uniqueName": "100",
      "name": "corporate"
    }
    ],


    "expirationTermType": "Perpetual",

  "effectiveDate": "2019-12-09T20:02:58.303+0000",
  "expirationDate": "2019-12-31T20:02:58.303+0000",
  "agreementDate" :"2020-12-09T20:02:58.303+0000",
     "owner": {
        "uniqueName": "molson",
        "passwordAdapter": "PasswordAdapter1",
        "name": "Max Olson",
        "emailAddress": "devnull@devmail.ariba.com",
        "organization": "Buyer Organization",
        "orgANId": "AN41000000377",
        "orgName": "Buyer Organization"
      },

  "customFields": [
      {
          "fieldId": "cus_CF_Boolean",
        "fieldName": "CF_Boolean",
        "fieldDataType": "booleanValue",
```

```
        "booleanValue": true
      },
      {
          "fieldId": "cus_CF_Text",
        "fieldName": "CF_Text",
        "fieldDataType": "textValue",
         "textValue": [
           "IM UPDATED Again"
         ]
      },
      {
          "fieldId": "cus_eTestA",
        "fieldName": "eTestA",
        "fieldDataType": "textValue",
        "textValue": [
            "Updated Again"
         ]
      }

      ],

        "questions" : [{
      "questionId" : "QI2516",
      "answerText" : "No"
  },{
      "questionId" : "QI2517",
      "answerText" : "No"
  }]
 “accessControl”: [
        “Legal Information”,
        “Private To Team Members”
      ],
  “relatedId” : “12345",
"blockPublish": false,
      "integratedSystemRefs": [
        {
          "integratedSystemName": "Icertis",
          "integratedSystemID": "fg001",
          "status": "Executed",
          "type":"Agreement",
          "urlData": "csow/csow1"
        }
"eventInternalId": "Doc12345678"
"baselineContract": false
}
```

`accessControl` Use the `accessControl` object to specify access control permission for the contract workspace. For a list of access control values available in SAP Ariba Contracts, see About the PermissionBitVec element in contract exports. However, the availability of these access control groups depends on the site configurations. Some of these groups may not be available to you.

> **Note**
>
> For the `accessControl` object, the value must be entered in **English** irrespective of the locale of your site.

`relatedID` Use the `relatedID` object to specify a related ID. For example, the ID for a related record or document, such as an outline agreement or a contract in an external system.

`blockPublish` Use the `blockPublish` object to specify if the contract workspace can be published after creation.

`integratedSystemRefs` Use the `integratedSystemRefs` object to specify external systems (such as Icertis, and SAP Fieldglass) to be integrated with the contract workspace.

> **Note**
>
> You can include multiple `integratedSystemRefs` objects in the query body when you are creating a contract workspace that is integrated with multiple integrated systems.

`eventInternalId` The predecessor sourcing project ID to associate with the contract.

> **Note**
>
> - If you are enter the ID of a guided sourcing full project, the full project is associated as the **Predecessor Project** of the contract. The contract is also automatically associated with the subprojects in the full project.
> - If you enter the ID of a subproject in a guided sourcing full project, the subproject is associated as the **Predecessor Event** and the full project is associated as the **Predecessor Project** of the contract. The contract is not associated with other subprojects in the full project.
> - If you enter the ID of a guided sourcing event, the event is associated as the **Predecessor Project** and the **Predecessor Event** of the contract.
> - If you are creating a baseline contract, you must set the `baselineContract` property to **`true`**.

Results

If the request is successful, the API creates a contract workspace with the specified details in SAP Ariba Contracts. For a list of response codes, refer to Error Messages Returned by the Contract Workspace Management APIs [page 65].

Next Steps

Open the contract in SAP Ariba Contracts and verify the details.

Create Contract Workspaces by Copying Existing Contracts Using the Contract Workspace Modification API

Implement the Contract Workspace Modification API to create a contract workspace by copying contents from an existing contract workspace.

Prerequisites

When using the Contract Workspace Modification API to create a contract workspace having a hierarchical relationship with another contract, ensure that the parent agreement already exists and has a valid hierarchical type. The parent agreement of a subagreement can be either a master agreement or a subagreement.

Context

The Contract Workspace Modification API enables you to create a contract workspace by copying contents from an existing contract workspace in SAP Ariba Contracts using the `POST /contractworkspaces` endpoint.

> **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.

To create a contract workspace by copying contents from an existing contract workspace, you must include the following details in the request body:

- Contract title (`title`)
- Contract description (`description`)
- Effective date (`effectiveDate`)
- ID of the source contract workspace you wish to copy (`copyFromContractID`)

You can include other contract-related information such as commodities, regions, departments, suppliers, and so on, in the request body. If you do, those details will replace the corresponding information copied the source contract workspace.

> **Note**
>
> The API uses the template from the source contract to create a new contract. Therefore, template-related information (`templateID`) should not be included in the request body.

Procedure

Invoke the `POST /contractworkspaces` endpoint to create a contract workspace.

Request type: POST

URL: **`{{runtime_url}}/contractworkspaces`**

The following example creates a contract workspace (procurement) with the specified details:

> **Example**
>
> Request URL: `POST {{runtime_url}}/contractWorkspaces?user=adavis&passwordAdapter=PasswordAdapter1&realm=s4All-7`
>
> Query Body:

```
{
  "title": "Test Contract Workspace (Procurement)",
  "description": "This contract is created by copying content from an existing contract",
  "effectiveDate": "2025-06-24T20:02:58.303+0000",
  "copyFromContractID": "CW10001010"
}
```

`copyFromContractID` Use the `copyFromContractID` object to specify the ID of the contract you wish to copy.

> **Note**
>
> For users who have SAP Ariba Contracts integrated with SAP Ariba Sourcing, and who have further integrated with Icertis Contract Intelligence (ICI) for SAP Ariba Solutions to manage all contract-related tasks: If you are creating a supplier contract in a guided sourcing event set up for pre-award negotiation, specify the ID of the linked baseline contract in the `copyFromContractID` object.

Results

If the request is successful, the API creates a contract workspace with the specified details in SAP Ariba Contracts. For a list of response codes, refer to Error Messages Returned by the Contract Workspace Management APIs [page 65].

> **Note**
>
> The following details from the source contract workspace are not copied into the new contract workspace:
>
> - Contract documents
> - Project groups
> - Project phases and tasks.

Next Steps

Open the contract in SAP Ariba Contracts and verify the details.

Changing the State of Procurement Contract Workspace Using the Contract Workspace Modification API

Use this procedure to change the state of a procurement 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.
- Users who manage the procurement contract workspace information must have membership of the **Contract Administrator** group.

Context

You can change the status of a contract workspace to `OnHold`, `Draft`, `DraftAmendement`, and `Closed` by using the `POST /contractWorkspaces/{contractId}/state` endpoint of the Contract Workspace Modification API. When you want to update a procurement contract workspace that is in the published state, you must change the state of the contract workspace using the `amend` action. Once contract workspace is updated, we can either publish contract workspace or call the `completeAmendment` action.

> **Note**
>
> When contract workspace is amended for **Administrative** or **PriceUpdate** amendment type, user can complete amendment by using the `completeAmendment` action. For amendments of types such as **Amendment**, **Renewal**, or **Termination**, users must publish the changes either manually by using the user interface or by using an API. For more information about changing the status of the contract workspace to publish, see Changing the State of a Contract Workspace by Using the Contract Workspace State Change API [page 33].
>
> The `action` and `amendmentAttributes` parameters of the `/contractWorkspaces/{contractId}/state` endpoint must be entered in English irrespective of the locale of your site.

> **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 contract workspace for which you want to change the state.
2. Invoke the `/contractWorkspaces/{contractId}/state` endpoint.

```
POST {{runtime_url}}/contractWorkspaces/{contractId}/state?
user=user1&passwordAdapter=pwd1&realm=realm1
{
  "action": "amend" | "completeAmendment" | "putOnHold" | "resume" | "close"
}
```

where

contractId denotes the ID of the contract workspace that you want to update.

action specifies the action you want to perform on the workspace. Valid values: `amend`, `completeAmendment`, `putOnHold`, `resume`, and `close`.

Refer to the following table for the sample request body and response you receive on successfully submitting this endpoint for each action value.

When the `action` Value is...	From State	Sample Request Body	Response You Receive...
`amend`	• `Published`	`{ "action": "amend", "amendAttributes": { "amendmentType": "Administrative" | "PriceUpdate", "amendmentReason": "performing amendment", "isTemplateUpgradeRequired": true } }`	`{ "contractId": "CW1234", "contractStatus": "Published", "version": 3 "displayVersion": "v2 (editing)", "warnings": [ "string" ] }`

When the `action` Value is...	From State	Sample Request Body	Response You Receive...

		<pre>{
		  "action":
		"amend",
		
		"amendAttributes
		": {
		
		"amendmentType":
		 "Renewal" |
		"Amendment" |
		"Termination",
		
		"amendmentReason
		": "performing
		amendment",
		
		"isTemplateUpgra
		deRequired":
		true
		  }
		}</pre>
			<pre>{
			  "contractId":
			"CW1234",
			  "contractStatus":
			"DraftAmendment",
			  "version": 3,
			  "displayVersion":
			"v2 (editing)",
			  "warnings": [
			  "string"
			  ]
			}</pre>

	`Closed` (non-published **Administrative** or **PriceUpdate** amendment)	<pre>{
		  "action":
		"amend",
		
		"amendAttributes
		": {
		
		"amendmentType":
		
		"Administrative"
		 |
		"PriceUpdate",
		
		"amendmentReason
		": "performing
		amendment",
		
		"isTemplateUpgra
		deRequired":
		true
		  }
		}</pre>
			<pre>{
			  "contractId":
			"CW1234",
			  "contractStatus":
			"Closed",
			  "version": 3,
			  "displayVersion":
			"v2 (editing)",
			  "warnings": [
			  "string"
			  ]
			}</pre>

When the `action` Value is...	From State	Sample Request Body	Response You Receive...
	`Closed` (non-published **Amendment**, **Renewal**, or **Termination** amendment)	`{ "action": "amend", "amendAttributes": { "amendmentType": "Amendment" | "Renewal" | "Termination", "amendmentReason": "performing amendment", "isTemplateUpgradeRequired": true } }`	`{ "contractId": "CW1234", "contractStatus": "DraftAmendment", "version": 3, "displayVersion": "v2 (editing)", "warnings": [ "string" ] }`
`completeAmendment`	• `Published` • `Closed` (non-published **Administrative** or **PriceUpdate** amendment)	`{ "action": "completeAmendment" }`	`{ "contractId": "CW1234", "contractStatus": "Published" | "Closed", "version": 3 "displayVersion": "v3", "warnings": [ "string" ] }`
`putOnHold`	• `Draft` • `DraftAmendment`	`{ "action": "putOnHold" }`	`{ "contractId": "CW1234", "contractStatus": "OnHold", "version": 1, "displayVersion": "Original", "warnings": [ "string" ] }`

When the `action` Value is...	From State	Sample Request Body	Response You Receive...
	`Published`	`{ "action": "putOnHold" }`	`{ "contractId": "CW1234", "contractStatus": "OnHold", "version": 1, "displayVersion": "v1", "warnings": [ "string" ] }`
`resume`	`OnHold` (when you put a Draft contract workspace on hold)	`{ "action": "resume" }`	`{ "contractId": "CW1234", "contractStatus": "Draft", "version": 1, "displayVersion": "Original", "warnings": [ "string" ] }`
	`OnHold` (when you put a DraftAmendment contract workspace on hold)		`{ "contractId": "CW1234", "contractStatus": "DraftAmendment", "version": 1, "displayVersion": "Original", "warnings": [ "string" ] }`
	`OnHold` (when you put a Published contract workspace on hold)	`{ "action": "resume", "amendAttributes": { "amendmentReason": "amendment reasons", "isTemplateUpgradeRequired": "true" } }`	`{ "contractId": "CW1234", "contractStatus": "DraftAmendment", "version": 2, "displayVersion": "v1 (editing)", "warnings": [ "string" ] }`

When the `action` Value is...	From State	Sample Request Body	Response You Receive...

`close`	• `Draft`
	• `DraftAmendement`
	• `Published`
		<pre>{
		"action": "close"
		}</pre>
			<pre>{
			"contractId": "CW1234",
			"contractStatus": "Closed",
			"version": 1,
			"displayVersion": "v1",
			"warnings": [
			"string"
			]
			}</pre>

where

amendmentType indicates the type of amendment. The supported values are `Administrative`, `Renewal`, `Amendment`, `Termination`, and `PriceUpdate`.

> **Note**
>
> You can do the `PriceUpdate` amendment only when there is a CLID in the contract workspace.

amendmentReason specifies the reason for the amendment.

isTemplateUpgradeRequired indicates whether the amendment requires a template upgrade. Supported values are `true` and `false`.

contractStatus indicates the status of the contract after changing the state. Valid values: `DraftAmendement`, `OnHold`, `Draft`, and `Closed`.

version indicates the version of the contract workspace after the updates are made.

displayVersion indicates the current version of the contract workspace as displayed in UI.

warnings contains any warning messages.

For a list of error messages, see Error Messages Returned by the Contract Workspace Management APIs [page 65].

Updating Contract Workspaces Using the Contract Workspace Modification API

The Contract Workspace Modification API can be implemented in a client application to use this procedure to update procurement contract workspaces in SAP Ariba Contracts.

Prerequisites

You have the unique identifier of the contract workspace you want to update.

Context

The Contract Workspace Modification API enables you to update a contract workspace in SAP Ariba Contracts using the `PATCH /contractworkspaces` endpoint.

> ⓘ Note
>
> If you have integrated SAP Ariba Contracts with SAP Ariba Sourcing and have guided sourcing enabled, you can include predecessor project information in the API query body. However, to enable the Contract Workspace Modification API to identify the predecessor event information, in addition to the predecessor project information, of the contract whenever applicable, the **Enable predecessor event identification for contracts created from guided sourcing projects** (`IdentifyPredecessorEventforContractsCreatedFromGuidedSourcingProjects`) ICM parameter must be enabled. For more information, refer to Enable predecessor event identification for contracts created from guided sourcing projects.

You can construct a query body by specifying a contract workspace ID and the properties to be updated in the contract workspace.

> ⓘ 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. Determine the fields you want to modify, and construct a query body that can be used in the request to update the contract workspace. Many fields are populated using the domain body parameter, as follows:

   > {·} Sample Code
   >
   ```
   "domain":"<field_name>",
   ```

```
"value":"<field_value>"
```

2. Invoke the `PATCH /contractworkspaces` endpoint to update the existing contract workspace, providing the contract workspace fields for updating the contract.

   Request type: PATCH

   URL: **`{{runtime_url}}/contractworkspaces`**

   The following table describes the query parameters available with this endpoint.

Parameter	Description
`user` and `passwordAdapter`	These parameters specify your API credentials.
`realm`	This parameter specifies the unique identifier for your SAP Ariba site.
`silentUpdate`	This parameter specifies whether the contract workspace must be updated without changing the last modified date. This parameter is used by the integrated systems to update the *`integratedSystemRef`* and *`blockPublish`* fields of published contract workspaces. This is an optional parameter.

The following example updates a contract workspace with the specified details:

Example

Request URL: `PATCH {{runtime_url}}/contractWorkspaces?user=adavis&passwordAdapter=PasswordAdapter1&realm=s4All-7&silentUpdate=true`

Query Body:

```
{
  "title": "Test Search With Spaces",
  "description": "Demo Description",

  "isTestProject": true,

  "hierarchicalType" :"MasterAgreement",
  "parentAgreement":"",
  "templateId": "SYS0052",

  "supplier": {
    "name": "Apex Corporation",
    "systemID": "sid509",
    "organizationIDs": [
      {
        "domain": "internalsystemid",
        "value": "1009"
      },
      {
        "domain": "duns",
        "value": "700000080"
      },
      {
```

```
        "domain": "networkid",
        "value": "an70000000080"
      },
      {
        "domain": "oracle",
        "value": "4009"
      },
      {
        "domain": "psoft",
        "value": "share:usa0000022"
      },
      {
        "domain": "sap",
        "value": "0000001050"
      },
      {
        "domain": "buyersystemid",
        "value": "sid509"
      },
      {
        "domain": "sourcingnetworkid",
        "value": "an70000000080"
      }
    ],
    "address": {
      "name": "",
      "uniqueName": "sid509",
      "phone": "",
      "fax": "",
      "lines": [
        "753 Apex Avenue"
      ],
      "city": "Marlborough",
      "state": "MA",
      "postalCode": "1752",
      "country": "US"
    }
  },
  "affectedParties": [
        {
          "name": "Apex Corporation",
          "systemID": "sid509",
          "smVendorID": null,
          "organizationIDs": [
            {
              "domain": "internalsystemid",
              "value": "1009"
            },
            {
              "domain": "duns",
              "value": "700000080"
            },
            {
              "domain": "networkid",
              "value": "an70000000080"
            },
            {
              "domain": "oracle",
              "value": "4009"
            },
            {
              "domain": "psoft",
              "value": "share:usa0000022"
            },
            {
```

```
              "domain": "buyersystemid",
              "value": "sid509"
            },
            {
              "domain": "sourcingnetworkid",
              "value": "an70000000080"
            },
            {
              "domain": "sap",
              "value": "2000003"
            }
          ],
          "address": {
            "name": "",
            "uniqueName": "sid509",
            "phone": "",
            "fax": "",
            "lines": [
              "753 Apex Avenue"
            ],
            "city": "Marlborough",
            "state": "MA",
            "postalCode": "1752",
            "country": "US"
          }
        }
      ],


    "contractAmount": {
    "amount": 1000,
    "currency": "Euro"
  },

  "proposedContractAmount":
  {
      "amount": 1000,
    "currency": "Euro"
  },

      "commodities": [
    {
      "uniqueName": "56",
      "domain": "unspsc",
      "name": "Furniture and Furnishings"
    }
  ],

  "regions": [
    {
      "uniqueName": "APAC",
      "name": "Asia Pacific"
    }
  ],

  "departments": [
    {
      "uniqueName": "100",
      "name": "corporate"
    }
    ],


    "expirationTermType": "Perpetual",
```

```
  "effectiveDate": "2019-12-09T20:02:58.303+0000",
  "expirationDate": "2019-12-31T20:02:58.303+0000",
  "agreementDate" :"2020-12-09T20:02:58.303+0000",
     "owner": {
        "uniqueName": "molson",
        "passwordAdapter": "PasswordAdapter1",
        "name": "Max Olson",
        "emailAddress": "devnull@devmail.ariba.com",
        "organization": "Buyer Organization",
        "orgANId": "AN41000000377",
        "orgName": "Buyer Organization"
      },

  "customFields": [
      {
          "fieldId": "cus_CF_Boolean",
        "fieldName": "CF_Boolean",
        "fieldDataType": "booleanValue",
        "booleanValue": true
      },
      {
          "fieldId": "cus_CF_Text",
        "fieldName": "CF_Text",
        "fieldDataType": "textValue",
         "textValue": [
            "IM UPDATED Again"
         ]
      },
      {
          "fieldId": "cus_eTestA",
        "fieldName": "eTestA",
        "fieldDataType": "textValue",
        "textValue": [
            "Updated Again"
         ]
      }

      ],
 “accessControl”: [
        “Legal Information”,
        “Private To Team Members”
      ],
  “relatedId” : “12345",
"blockPublish": true,
      "integratedSystemRefs": [
        {
          "integratedSystemName": "Icertis",
          "integratedSystemID": "fg001",
          "status": "Executed",
          "type":"Agreement",
          "urlData": "csow/csow1"
        }
    ]
"eventInternalId": "Doc12345678"
}
```

In the preceding example, the parameter `silentUpdate` is set to **`true`** in the endpoint which implies that the updates are performed without changing the last modified date of the contract workspace.

> ⓘ Note
>
> If you update the supplier information in the contract workspace, then you must resynchronize the line items information with the linked compliance contract using the Contracts Terms Management API. For

more information, refer to Synchronizing Contract Line Item Data of a Contract Workspace With Linked Compliance Contract Using Contract Terms Management API [page 37].

`accessControl` Use the `accessControl` object to specify access control permission for the contract workspace. For a list of access control values available in SAP Ariba Contracts, see About the PermissionBitVec element in contract exports. However, the availability of these access control groups depends on the site configurations. Some of these groups may not be available to you.

> ⓘ Note
>
> For the `accessControl` object, the value must be entered in English irrespective of the locale of your site.

`relatedID` Use the `relatedID` object to specify a related ID. For example, the ID for a related record or document, such as an outline agreement or a contract in an external system.

`blockPublish` Use the `blockPublish` object to specify if the contract workspace can be published after creation.

`integratedSystemRefs` Use the `integratedSystemRefs` object to specify external systems (such as Icertis, SAP Fieldglass) to be integrated with the contract workspace.

> ⓘ Note
>
> You can include multiple `integratedSystemRefs` objects in the query body when you are updating a contract workspace that's integrated with multiple integrated systems.

`eventInternalId` The predecessor sourcing project ID to associate with the contract.

> ⓘ Note
>
> - If you are enter the ID of a guided sourcing full project, the full project is associated as the **Predecessor Project** of the contract. The contract is also automatically associated with the subprojects in the full project.
> - If you enter the ID of a subproject in a guided sourcing full project, the subproject is associated as the **Predecessor Event** and the full project is associated as the **Predecessor Project** of the contract. The contract is not associated with other subprojects in the full project.
> - If you enter the ID of a guided sourcing event, the event is associated as the **Predecessor Project** and the **Predecessor Event** of the contract.

Results

If the request is successful, the API updates the contract workspace with the specified details in SAP Ariba Contracts.

> **Note**
>
> If you change the predecessor event of a contract workspace, the data copied from the previous predecessor event remains in the contract.

For a list of response codes, refer to Error Messages Returned by the Contract Workspace Management APIs [page 65].

Next Steps

Open the contract in SAP Ariba Contracts and verify the details.

Changing the State of a Contract Workspace by Using the Contract Workspace State Change API

Use this procedure to change the status of the contract workspace from draft or draft amendment to publish by using the Contract Workspace State Change 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.
- In suite-integrated solutions, if you want contract requests invoked through the `publish` action of the Contract Workspace State Change API to be automatically approved in SAP Ariba, ensure that the `Application.Contract.EnableAutoApprovalOfCRsInvokedThroughPublishAction` parameter is set to **Yes**. By default, this parameter is set to **Yes** so that the contract requests are automatically approved in SAP Ariba. When this parameter is set to **No**, the contract requests go through the manual approval workflow. For information on the parameter, see Enable auto approval of contract requests invoked through Publish action.
- Users who manage the procurement contract workspace information must be members of the **Contract Administrator** group.

Context

The `POST /contractWorkspaces/{contractId}/state` endpoint of the Contract Workspace State Change API enables you to change the status of the contract workspace from draft or draft amendment to publish. On successful submission of the API request, a `jobId` is returned by the Contract Workspace State Change API. Using the `jobId`, you can get the status of an asynchronous job request. For more information about getting the status of an asynchronous job request, see Getting the Status of an Asynchronous Job Using the Asynchronous Requests Management API.

> **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 contract workspace for which you want to change the state.
2. Invoke the `/contractWorkspaces/{contractId}/state` endpoint:

```
POST {{runtime_url}}/contractWorkspaces/{contractId}/state?
user=user1&passwordAdapter=pwd1&realm=realm1
{
  "action": "publish"
}
```

where

contractId	denotes the ID of the contract workspace that you want to update.
action	specifies the action you want to perform on the workspace. Valid value: **publish**

The Contract Workspace State Change API returns the job ID and a message indicating the status of the request.

```
{
    "jobId": "AAAAAA00Aaa",
    "message": "Job submitted successfully. Use jobId to get status of the job."
}
```

where

jobId	indicates the job ID after the API request is submitted successfully. Using the `jobId`, you can get the status of an asynchronous job request. For more information about getting the status of an asynchronous job request, see Getting the Status of an Asynchronous Job Using the Asynchronous Requests Management API.
message	contains the message indicating the status of the request.

> **Note**
>
> If you publish a contract workspace that is linked to a compliance contract, then all the header level data of the contract workspace, except the supplier information, is synchronized with the linked compliance contract.

For a list of error messages, see Error Messages Returned by the Contract Workspace Management APIs [page 65].

Creating and Reviewing Contract Terms by Using the Contract Term Management API

This topic describes the procedure for adding new contract terms and retrieving information about contract terms by using the Contract Terms Management 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.
- Before you can create contract terms, you must have created a contract workspace in SAP Ariba Contracts or SAP Ariba strategic sourcing solutions.

Context

The Contract Terms API enables you to create a contract terms document in the contract workspace and a corresponding contract request in Ariba compliance contracts. You can also use the API to query for contact terms details and compliance contract details.

> **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 contract workspace where you want to create the contract terms.
2. To create a contract terms document in the contract workspace and a corresponding contract request in SAP Ariba Procurement solutions, use the `POST /contractWorkspaces/{contractId}/contractTerms` endpoint as follows:

```
POST {{runtime_url}}/contractWorkspaces/contractworkspaceID/contractTerms?
user=user1&passwordAdapter=pwd1&realm=realm1
RequestBody:
{
"sspRealm" = "realm1"
}
```

   where

`contractworkspaceID`	is the ID of the contract workspace where you want to create the contract terms document.
`realm1`	is the name of the realm to which the sourcing and procurement applications belong. In case of a Federated Process Control (also known as multi ERP or mERP) setup, specify the child site (realm) ID for the `sspRealm` field in the request body.

   On successful creation of the contract terms document and the corresponding contract request, you receive a success message. The response body contains the contract terms document ID and contract request ID along with other details about the newly-created contract terms document and the contract request.

   If the `POST` operation encountered an error, you receive an error message that indicates the nature of the error.
3. To view details about the contract terms document in a contract workspace and the corresponding contract request, use the `GET /contractWorkspaces/{contractId}/contractTerms` endpoint as follows:

```
GET {{runtime_url}}/contractWorkspaces/contractworkspaceID/contractTerms?
user=user1&passwordAdapter=pwd1&realm=realm1
```

   where

`contractworkspaceID`	is the ID of the contract workspace from which you want to retrieve contract terms document details.

   When the `GET` request is successful, you receive the following details in the response:

```
{
  "documentId": "Doc12345",
  "owner": "Robin",
  "status": "Draft",
  "title": "Untitled Contract Terms",
  "lastModifiedDate": "Sat May 23 18:02:12 GMT 2020",
  "creationDate": "Sat May 23 18:02:06 GMT 2020",
  "version": "1"
  "sspContractRequestId": "CR1",
  "sspContractId": "C1",
  "sspContractStatus": "Not Created",
  "sspContractRequestStatus": "Composing",
}
```

   where

`documentId`	is the contract terms document ID in the contract workspace.
`sspContractRequestId`	is the contract request ID in the procurement solution.
`sspContractId`	is the contract ID in the procurement solution.

If the `GET` request encountered an error, you receive a status message that indicates the nature of the error.

Results

For a list of response codes for the Contract Terms Management API, see Error Messages Returned by the Contract Workspace Management APIs [page 65].

Synchronizing Contract Line Item Data of a Contract Workspace With Linked Compliance Contract Using Contract Terms Management API

Use this procedure to implement the Contract Terms Management API in a client application to synchronize the contract line item data in a SAP Ariba Contracts contract workspace with the linked compliance contract.

Context

The `POST /contractWorkspaces/{contractId}/contractTerms/{contractTermId}/state` endpoint of the Contract Terms Management API synchronizes the contract line item details in the specified SAP Ariba Contracts contract workspace with the compliance contract linked to it.

> ⓘ 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. Invoke the `POST /contractWorkspaces/{contractId}/contractTerms/{contractTermId}/state` endpoint.

   > ✤ Example
   >
   > ```
   > POST {{runtime_url}}/contractWorkspaces/{contractId}/contractTerms/
   > {contractTermId}/state?user=myUser&passwordAdapter=myPassword&realm=myRealm
   > ```

{contractId}	The ID of the contract workspace whose line item information you want to synchronize.
{contractTermId}	The ID of the compliance contract linked with the specified contract workspace.
user	The unique name of the user on whose behalf the API call is invoked.
passwordAdapter	The password adapter for authenticating the user on whose behalf the request was initiated.
realm	The SAP Ariba Contracts site to which the API request is sent.

> **Note**
>
> All the parameter values are mandatory.

2. In the request body, enter the value for the variable *action* as **SyncLineItemsToTerms**.

Results

If the request is successful, the API initiates an asynchronous job to synchronize the contract line item data from the specified contract workspace with the linked compliance contract and returns a job ID for reference.

For information about the response codes returned by the API, refer to Error Messages Returned by the Contract Workspace Management APIs [page 65].

Getting Contract Workspaces Matching the Provided ID Using the Contract Workspace Retrieval API

Use this procedure to get contract workspaces matching the provided ID using the Contract Workspace Retrieval API.

Prerequisites

You must have a contract workspace containing a main agreement and supporting contract documents for a contract with a supplier, and you must have the associated contract workspace ID. For information about contract workspace IDs and how to obtain them, see SOAP ContractId and WorkspaceId elements for contract workspaces in SOAP web service API integration guide.

Context

If you created a contract workspace associated to store the contract Main Agreement and supporting contract documents for a contract with a supplier, you can use the contract workspace ID to search for the contract workspace matching the provided ID.

> **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 contract workspace ID of the contract workspace for which you require information. For information about contract workspace IDs and how to obtain them, see SOAP ContractId and WorkspaceId elements for contract workspaces in SOAP web service API integration guide.
2. Invoke the `GET /contractworkspaces$contractId` endpoint to request the contract workspace matching the provided contract workspace ID.

   Request type: GET

   URL: **`{{runtime_url}}/contractworkspaces/$id`**

Searching for Contract Workspaces Using the Contract Workspace Retrieval API

Use this procedure to retrieve details of the contract workspaces created in SAP Ariba to your client application by applying different search and filter criteria available with the Contract Workspace Retrieval API.

Prerequisites

You have the contract workspace attributes such as the contract type (for example, service contract), title, ID, region, and so on, as required by the API parameters to search and filter data.

Context

The `GET /contractworkspaces` endpoint of the Contract Workspace Retrieval API enables you to search contract workspaces in the system.

You can choose the contract workspace attributes that you want to retrieve by specifying them in the `select` clause of the search query, for example, Contract Type, Title, Contract ID, Region, and so on.

> **Note**
>
> If the title contains the word **and**, ensure that you enclose the entire title string within quotation marks. For example, if the title is *`CW bread and butter`*, specify it as **"CW bread and butter"**.

For a complete list of contract workspace attributes that can be specified in the `select` clause, see Contract Workspace Attributes Eligible for Specifying in the Select Clause to Search for Contract Workspaces [page 43].

You can further streamline the search results by applying filter criteria using the following contract workspace attributes:

- Title
- Last Modified Date
- Document Type
- Include Deleted
- Contract Status
- Integrated Systems

> **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. Invoke the `GET /contractworkspaces` endpoint to retrieve a list of contract workspaces matching the search criteria.

   Request type: GET

   URL: **`{{runtime_url}}/contractworkspaces`**
2. (Optional) Use the required query parameters to search and filter the result set.

   The following example request URL gets information about contracts matching the provided title:

   ```
   {{runtime_url}}contractWorkspaces?$filter=Title eq
   CW&$top=500&$count=true&user={username}&passwordAdapter={password}&realm={realm}&
   $select=Title,ContractId,ContractStatus,Supplier,TimeZoneID,EffectiveDate,Expirat
   ionDate,CreateDate,Version,LastPublishedDate,AmendmentType,AmendmentReason,Origin
   ,HierarchicalType,ParentAgreement,ExpirationTermType,Commodity,Region
   ```

   The following table describes the query parameters that allow you to filter the result set.

Parameter	Description
`$select`	This parameter is used to specify the fields of the contract workspace that the API must fetch. **Example** `$select=Title,ContractId,Region` In this example, the API fetches the *`Title`*, *`ContractId`*, and *`Region`* fields of the contract workspace.

Parameter	Description
`$filter`	This parameter is used to apply filter criteria for the search query. The following contract workspace attributes can be used as filter criteria: • Title • Last Modified Date **Note** You can use the operators `lt` (less than), `lte` (less than or equal to), `gt` (greater than), `gte` (greater than or equal to) to further streamline the filter criteria. The `lt` and `lte` operators convert the specified timestamp value to the GMT time zone and add an hour to the result. The `gt` and `gte` operators convert the specified timestamp value to the GMT time zone and subtract an hour from the result. Example: `LastModified lt 2022-9-12T22:22:22Z and LastModified gte 2010-11-15T22:22:22Z` will search for data whose last modified date value is greater than or equal to **2010-11-15T21:22:22Z** and less than **2010-11-15T21:22:22Z**. • Include Deleted • Contract Status • Integrated Systems **Example** `$count=true&$filter=LastModified lt 2022-9-12T22:22:22Z and LastModified gte 2010-11-15T22:22:22Z and Title eq 'Service Contract' and includeDeleted eq true and ContractStatus eq Draft, Published and IntegratedSystems eq "ABC"` In this example, the API fetches the contract workspaces modified between **2010-11-15T21:22:22Z** and **2022-9-12T23:22:22Z**, with the title as **Service Contract**, including deleted contract workspaces in the **Draft** or **Published** state that are integrated with the external system **ABC**. **Note** The API uses the **eq** operator to fetch records that not only match the filter criterion but also contain the filter criterion. So, if two contract workspaces are having the titles, **Service Contract** and **Service Contract 1** respectively, both are fetched by the API.

Parameter	Description
`$top`	This parameter specifies the number of contracts to fetch. **Note** The default value of this parameter is **500**. To fetch more than 500 records, specify the required value for this parameter. For example: `$top=2000`
`$skip`	This parameter specifies the number of contracts to skip (offset) from the beginning of the result set for uses, such as pagination. The default value is **0**.
`$count`	If set to true, this parameter specifies the count of contracts.
`$expand`	This parameter specifies additional information to fetch such as the plant information provided by suppliers in the "Retail Contract Terms" object of the Contract Workspace. This is an optional parameter and is used only when you have integrated Retail Sourcing with SAP Ariba Contracts. **Example** `GET /contractWorkspaces?$expand=relatedFactories`
`realm`	This parameter specifies the unique identifier for your SAP Ariba site.
`user` and `passwordAdapter`	These parameters specify your API credentials.

Contract Workspace Attributes Eligible for Specifying in the Select Clause to Search for Contract Workspaces

You can construct a search query to retrieve specific attributes of a contract workspace using the Contract Workspace Retrieval API. The contract workspace attributes that you can explicitly specify in the select clause of the search query are listed below.

- Title
- Owner
- Contract Id
- ContractStatus
- Supplier
- Effective Date
- Expiration Date
- Creation Date
- Amount
- Version

- Template Id
- Business System Id
- Amendment Type
- Amendment Reason
- Origin
- Hierarchical Type
- Parent Agreement
- Expiration Term Type
- Commodity
- Region
- Internal Id

Getting the Metadata of a Contract Line Items Document Using the Contract Line Items Document Retrieval API

The Contract Line Items Document Retrieval API can be implemented in a client application to use this procedure to get the metadata of the Contract Line Items Document (CLID) in a procurement contract workspace in SAP Ariba Contracts.

Prerequisites

To get the metadata of a CLID from a contract workspace, you must have the IDs of the procurement contract workspace and the corresponding CLID.

Context

The Contract Line Items Document Retrieval API enables you to get the properties and metadata of a Contract Line Items Document (CLID) in a contract workspace using the `GET /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}` endpoint. The response contains the metadata that includes all the standard and custom fields, task IDs and names, line item types (including hierarchical line items and line items with pricing conditions), and translations of line items in the CLID. The API also retrieves all the versions of the CLID excluding the version that is in the **Not Created** status.

> ### 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.

For a list of response codes for the API calls, see Error Messages Returned by the Contract Workspace Management APIs [page 65].

Procedure

Invoke the `GET /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}` endpoint to get the metadata of a CLID in a procurement contract workspace.

```
GET {{runtime_url}}/contractWorkspaces/{contractId}/contractLineItemDocuments/
{documentId}
```

`documentId`	The CLID identifier.
`contractId`	The procurement contract workspace identifier.

Downloading a Contract Line Items Document Using the Contract Line Items Document Download API

Use this procedure to download a contract line items document. The Contract Line Items Document Download API can be implemented in a client application to download the Contract Line Items Document (CLID) in a procurement contracts workspace in SAP Ariba Contracts.

Prerequisites

To download a CLID from a procurement contract workspace, you must have the IDs of the contract workspace and the corresponding CLID.

Context

The Contract Line Items Document Download API enables you to download a CLID associated with a procurement contract workspace by using the `GET /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}/files` endpoint. The CLID is downloaded in the standard Excel format.

> **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.

For a list of response codes for the API calls, see Error Messages Returned by the Contract Workspace Management APIs [page 65].

> **Note**
>
> API requests for Contract Line Items Documents (CLID) with a high volume of attribute data may exceed the standard 60-second response time, causing the requests to fail. The actual response time may vary depending on your site configuration. As a workaround, download the CLID using the **Download Contract Terms** SOAP web service. For more information, see Download Contract Terms.
>
> For example, a CLID with approximately 1000 line items, each with 40 attributes, may exceed the standard 60-second response time.

Procedure

Invoke the `GET /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}/files` endpoint to download the CLID from a procurement contract workspace.

```
GET {{runtime_url}}/contractWorkspaces/{contractId}/contractLineItemDocuments/
{documentId}/files
```

`documentId`	The CLID identifier.
`contractId`	The procurement contract workspace identifier.

Creating a Contract Line Items Document in a Contract Workspace using the Contract Line Items Document Management API

The Contract Line Items Document Management API can be implemented in a client application to create a Contract Line Items Document (CLID) in a contract workspace.

Prerequisites

You must have the ID of the contract workspace where you want to create the CLID.

Context

The `POST /contractWorkspaces/{contractId}/contractLineItemDocuments` endpoint of the Contract Line Items Document Management API enables you to create a CLID in a contract workspace. You can create a query body specifying the fields to be included in the CLID.

> **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

Invoke the `POST /contractWorkspaces/{contractId}/contractLineItemDocuments` endpoint.

> **Example**
>
> ```
> POST {{runtime_url}}/contractWorkspaces/CW12345/contractLineItemDocuments?
> user=myUser&passwordAdapter=myPassword&realm=mySite
> ```

Where,

- `user`, is the ID of the user on whose behalf the API call is invoked. This is a required parameter.
- `passwordAdapter`, is the password for authenticating the user on whose behalf the request was initiated. This is a required parameter.
- `realm`, is the SAP Ariba Contracts site to which the API request is sent. This is a required parameter.

In the request body, specify the fields that you want to add to the CLID. A sample request body is as follows:

> **Example**
>
> > **Sample Code**
>
> ```
> {
>     "BaseLanguage" : "en",
>     "Description" : "Test CLID",
>     "Owner" :
>     {
>         "uniqueName": "myUser",
>         "passwordAdapter": "myPassword",
>         "type": "User"
>     },
>     "Participant" :
>     {
>         "uniqueName": "myUser",
>         "passwordAdapter": "myPassword",
>         "type": "User"
>     },
>     "Title" : "Sample Contracts Line Items Document",
>     "ContentType" : "Standard",
>     "PublishRequired" : true,
>     "AllowFormulas": false,
>     "ConversionRateSet" : "CCRS1",
>     "Permissions" : ["ReadLegal"],
>     "SaveToNewVersion": true,
> ```

```
    "VersionComment" : "new version"
}
```

Results

If the request is successful, the Contract Line Items Document Management API returns the details of the CLID created in the specified contract workspace. For information about the response codes returned by the API, see Error Messages Returned by the Contract Workspace Management APIs [page 65].

Updating a Contract Line Items Document Using the Contract Line Items Document Management API

The Contract Line Items Document Management API can be implemented in a client application to update an existing Contract Line Items Document (CLID) in a contract workspace.

Prerequisites

- You must have the ID of the contract workspace where the CLID that you want to update exists.
- You must have the ID of the CLID that you want to update.

Context

The `PATCH /contractWorkspaces/{contractId}/contractLineItemDocuments/{id}` endpoint of the Contract Line Items Document Management API enables you to update an existing CLID in a contract workspace. You can create a request body specifying the fields of the CLID that you want to update.

> **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

Invoke the `PATCH /contractWorkspaces/{contractId}/contractLineItemDocuments/{id}` endpoint.

Example

Request URL:

```
PATCH {{runtime_url}}/contractWorkspaces/CW12345/contractLineItemDocuments/
Doc1234567?user=myUser&passwordAdapter=myPassword&realm=mySite
```

Where,

- `user`, is the ID of the user on whose behalf the API call is invoked. This is a required parameter.
- `passwordAdapter`, is the password for authenticating the user on whose behalf the request was initiated. This is a required parameter.
- `realm`, is the SAP Ariba Contracts site to which the API request is sent. This is a required parameter.

In the request body, specify the fields that you want to update in the CLID. A sample request body is as follows:

Example

Sample Code

```
{
    "BaseLanguage" : "en",
    "Description" : "Update CLID",
    "Owner" :
    {
        "uniqueName": "myUser",
        "passwordAdapter": "myPassword",
        "type": "User"
    },
    "Participant" :
    {
        "uniqueName": "myUser",
        "passwordAdapter": "myPassword",
        "type": "User"
    },
    "Title" : "Sample Contracts Line Items Document",
    "ContentType" : "Standard",
    "PublishRequired" : true,
    "AllowFormulas": false,
    "ConversionRateSet" : "CCRS1",
    "Permissions" : ["ReadLegal"],
    "SaveToNewVersion": true,
    "VersionComment" : "new version"
}
```

Results

If the request is successful, the Contract Line Items Document Management API returns the details of the CLID updated in the specified contract workspace. For information about the response codes returned by the API, see Error Messages Returned by the Contract Workspace Management APIs [page 65].

Publishing Contract Line Items Document by Using the Contract Line Items Document Publish API

Use this procedure to publish a contract line items document (CLID) from a procurement contracts workspace in SAP Ariba Contracts by using the Contract Line Items Document Publish 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.
- Users who manage the procurement contract workspace information must be members of the **Contract Administrator** group.
- You must have the ID of the contract workspace where the CLID that you want to publish exists.
- You must have the ID of the CLID that you want to publish.

Context

The Contract Line Items Document Publish API enables you to publish a CLID in a procurement contract workspace by using the `POST /contractWorkspaces/{contractId}/contractLineItemDocuments/{id}/state` endpoint.

> **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

Use the `POST /contractWorkspaces/{contractId}/contractLineItemDocuments/{id}/state` endpoint to publish the CLID associated with a procurement contract workspace.

```
POST {{runtime_url}}/contractWorkspaces/CW12345/contractLineItemDocuments/
Doc12345678/state?user=myUser&passwordAdapter=myPassword&realm=mySite
{
  "action": "publish"
}
```

where

contractId is the ID of the procurement contract workspace.

id is the CLID identifier.

User is the ID of the user on whose behalf the API call is invoked.

> **Note**
>
> The user must be a member of the **Contract Administrator** group.

passwordAdapter is the password for authenticating the user on whose behalf the request was initiated.

Realm is the SAP Ariba Contracts site to which the API request is sent.

action is the action you want to perform. While publishing the CLID, specify the action as **publish**.

The Contract Line Items Document Publish API returns the job ID and a message indicating the status of the request.

```
{
    "jobId": "AAAAA1234Aaa",
    "message": "Job submitted successfully. Use jobId to get status and result of
the job."
}
```

where

jobId indicates the job ID after the API request is submitted successfully. Using the `GET /jobs/{jobId}` endpoint, you can get the status of an asynchronous job request. For more information about getting the status of an asynchronous job request, refer to Getting the Status of an Asynchronous Job Using the Asynchronous Requests Management API. Using the `GET /jobs/{jobId}/errors` endpoint, you can download the error file for a failed job request. For more information about downloading the error file for a failed job request, refer to Downloading the Error File of a Failed Job Request Using the Asynchronous Requests Management API.

message contains the message indicating the request status.

To troubleshoot your queries to the Contract Line Items Document Publish API, refer to Error Messages Returned by the Contract Workspace Management APIs [page 65].

Importing Line Items to a Contract Line Items Document Using the Contract Line Items Import API

The Contract Line Items Import API can be implemented in a client application to import line item information to a Contract Line Items Document (CLID) in a contract workspace.

Prerequisites

- You must have the ID of the contract workspace and the ID of the CLID to which you want to import line items.
- You must have the preformatted copy of the CLID in an Excel spreadsheet that is updated with the data that you want to import. For more information about how to download a CLID to an Excel spreadsheet, refer to Adding Items to Standard-Capacity Contract Line Items Documents.

Context

The `POST /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}/files` endpoint of the Contract Line Items Import API enables you to add or replace line items information in a Contract Line Items Document (CLID) of a contract workspace by importing the data from an Excel file.

> **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. To add line items or update existing line items in a CLID, invoke the `POST /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}/files` endpoint.

   > **Example**
   >
   > ```
   > POST {{runtime_url}}contractWorkspaces/CW12345/contractLineItemDocuments/
   > Doc123455678/files?user=myUser&passwordAdapter=myPassword&realm=mySite&action=add
   > ```

   Where,

   - `user`POST, is the ID of the user on whose behalf the API call is invoked. This is a required parameter.
   - `passwordAdapter`, is the password for authenticating the user on whose behalf the request was initiated. This is a required parameter.
   - `realm`, is the SAP Ariba Contracts site to which the API request is sent. This is a required parameter.
   - `action`, is the type of action you wish to perform. The valid values are **add** and **replace**. While adding or updating line items in a CLID, specify the action as **add**.

In the request body, ensure that the Excel file containing line item information that you want to import is associated with the `file` element.

If the data import is successful. the line items from the Excel file are added to the CLID of the specified contract workspace. For information about the response codes returned by the API, see Error Messages Returned by the Contract Workspace Management APIs [page 65].

2. To replace line items of a CLID, invoke the `POST /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}/files` endpoint.

   > Example
   >
   > ```
   > POST
   > {{runtime_url}}contractWorkspaces/CW12345/contractLineItemDocuments/Doc123455678/files?user=myUser&passwordAdapter=myPassword&realm=mySite&action=replace
   > ```

   Where,

   - `user`, the ID of the user on whose behalf the API call is invoked. This is a required parameter.
   - `passwordAdapter`, is the password for authenticating the user on whose behalf the request was initiated. This is a required parameter.
   - `realm`, the SAP Ariba Contracts site to which the API request is sent. This is a required parameter.
   - `action`, the type of action you wish to perform. The valid values are **add** and **replace**. While replacing line items in a CLID, specify the action as **replace**.

   In the request body, ensure that the Excel file containing line item information that you want to import is associated with the `file` element.

   If the data import is successful, existing line items (if any) in the specified CLID are replaced with line items from the Excel file. For information about the response codes returned by the API, see Error Messages Returned by the Contract Workspace Management APIs [page 65].

Deleting or Deactivating Line Items in a Contract Line Items Document Using the Contract Line Items Import API

Use this procedure to implement the Contract Line Items Import API in a client application to delete or deactivate line items in a Contract Line Items Document (CLID) in a contract workspace.

Prerequisites

- Your administrator must enable the **Allow Line Item Deletion Through Excel Import** (Application.ACM.AllowLineItemDeletionThroughExcelImport) ICM parameter. For information about this parameter, see Allow Deleting Line Items Through Excel Import.
- You must have atleast one line item in the CLID that you want to modify.
- You must have the ID of the contract workspace and the ID of the CLID to which you want to import content.

Context

Use the `POST /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}/files` endpoint of the Contract Line Items Import API to delete or deactivate line items in a CLID of a contract workspace by importing the data from an Excel file.

> ### 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. Edit the Excel version of the CLID to prepare the data to import.

   For information about editing the Excel version of a CLID, see Editing Excel Versions of Contract Line Items Documents.

   In the **Contract Item Information** worksheet, enter a specific value in the **Item Status** column for the line items that you want to delete or deactivate. To delete a line item, enter **`HardDeleted`**. To deactivate a line item, enter **`Deleted`**.

   > ### Note
   >
   > For hierarchical line items that have child line items, make note of the following:

Item Status (Parent Line Item)	Item Status (Child Line Item)	Result
`Deleted`	**`Deleted`** or no entry	The parent and child line items are deactivated.
`HardDeleted`	**`HardDeleted`** or no entry	The parent and child line items are deleted.
`Deleted`	**`HardDeleted`**	The parent line item is deactivated and child line item is deleted.
`HardDeleted`	**`Deleted`**	The parent and child line items are deleted.

2. To import the Excel file, invoke the `POST /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}/files` endpoint from the client application.

   > ### Example
   >
   > ```
   > POST {{runtime_url}}/
   > contractWorkspaces/CW12345/contractLineItemDocuments/Doc123455678/files?
   > user=myUser&passwordAdapter=myPassword&realm=mySite&action=update
   > ```

   `user` Specifies the ID of the user on whose behalf the API call is invoked. This is a mandatory value.

`passwordAdapter` Specifes the password for authenticating the user on whose behalf the request was initiated. This is a mandatory value.

`realm` Specifies the SAP Ariba Contracts site to which the API request is sent. This is a mandatory value.

`action` Specifies the action to perform. The valid values are **add**, **update**, and **replace**.

> ⓘ Note
>
> Ensure that you specify **update** in the `action` parameter.

In the request body, associate the Excel file that contains the updated line item information to import, with the `file` element.

Results

If the data import is successful, the contents from the Excel file are imported to the CLID of the contract workspace. For information about the response codes returned by the API, see Error Messages Returned by the Contract Workspace Management APIs [page 65].

Next Steps

On the CLID details page of the contract workspace, verify that the updates to the line items are as expected.

> ⓘ Note
>
> Deleted line items are not shown in the line items table on the CLID details page. Deactivated line items are highlighted in grey in the line items table and are not editable.

Importing Deactivated Line Items From an Integrated ERP System to a Contract Workspace Using Contract Line Items Import API

Use this procedure to implement the Contract Line Items Import API in a client application to import deactivated line items from an integrated ERP system to a contract workspace.

Prerequisites

- Your administrator must enable the following ICM parameters:
  - **Allow Line Item Deletion Through Excel Import** (Application.ACM.AllowLineItemDeletionThroughExcelImport). For information about this parameter, see Allow Deleting Line Items Through Excel Import.
  - **Allow Importing Deactivated ERP Items Through Excel Import** (Application.ACM.AllowDeletedERPItemsThroughCLIExcelImport). For information about this parameter, see Allow Importing Deactivated Line Items From an ERP System Through Excel Import.

    > **Note**
    >
    > This parameter is available only after you enable the **Allow Line Item Deletion Through Excel Import** parameter.
- You must have the ID of the contract workspace and the ID of the CLID to which you want to import content.

Context

Use the `POST /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}/files` endpoint of the Contract Line Items Import API to import deactivated line items from an integrated ERP system to a contract workspace that does not contain the line items information in the associated Contracts Line Items Document (CLID).

> **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. Edit the Excel version of the CLID to prepare the data to import.

   For information about editing the Excel version of a CLID, see Editing Excel Versions of Contract Line Items Documents.

In the **Contract Item Information** worksheet, enter the line item information as recorded in the ERP system. In the **Item Status** column for each line item, enter a value as **`Deleted`**.

2. To import the Excel file, invoke the `POST /contractWorkspaces/{contractId}/contractLineItemDocuments/{documentId}/files` endpoint from the client application.

> ### Example
>
> ```
> POST {{runtime_url}}/
> contractWorkspaces/CW12345/contractLineItemDocuments/Doc123455678/files?
> user=myUser&passwordAdapter=myPassword&realm=mySite&action=update
> ```

`user` Specifies the ID of the user on whose behalf the API call is invoked. This is a mandatory value.

`passwordAdapter` Specifes the password for authenticating the user on whose behalf the request was initiated. This is a mandatory value.

`realm` Specifies the SAP Ariba Contracts site to which the API request is sent. This is a mandatory value.

`action` Specifies the action to perform. The valid values are **add**, **update**, and **replace**.

> ### Note
>
> Ensure that you specify **update** in the `action` parameter.

In the request body, associate the Excel file that contains the updated line item information to import, with the `file` element.

Results

If the data import is successful, the contents from the Excel file are imported to the CLID. For information about the response codes returned by the API, see Error Messages Returned by the Contract Workspace Management APIs [page 65].

Next Steps

On the CLID details page of the contract workspace, verify that the updates to the line items are as expected.

> ### Note
>
> Deactivated line items are highlighted in grey in the line items table and are not editable.

Deleting a Contract Workspace Using the Contract Workspace Delete API

Use this procedure to delete a procurement contract workspace using the Contract Workspace Delete API.

Context

The `DELETE /contractWorkspaces/{contractId}` endpoint of the Contract Workspace Delete API enables you to delete a contract workspace.

You can only delete procurement contract workspaces that are in the **Draft** state. In addition, ensure that the following conditions are met:

- The procurement contract workspace must not have project **To Do** tasks that are in the **Started** state or the **Completed** state.
- The procurement contract workspace must not have document tasks (such as **To Do**, **Review**, **Approval**, **Negotiation**, and **Signature**) that are in the **Started** state, **Completed** state, or the **Active** state.
- The procurement contract workspace must not contain sub-projects having tasks in the **Started** state or the **Completed** state.
- The procurement contract workspace must not contain phases having **To Do** tasks in the **Started** state.
- The procurement contract workspace must not contain phases having **Approval**, **Review**, or **Negotiation** tasks in the **Started** or the **Completed** state.

> ⓘ 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

Invoke the `DELETE /contractWorkspaces/{contractId}` endpoint.

> ✤ Example
>
> ```
> DELETE {{runtime_url}}/contractWorkspaces/{contractId}?
> user=myUser&passwordAdapter=myPassword&realm=myRealm
> ```

`{contractId}`	Specifies the ID of the contract workspace to delete. This is a mandatory value.
`user`	Specifies the unique name of the user on whose behalf the API call is invoked. This is a mandatory value.
`passwordAdapter`	Specifies the password adapter for authenticating the user on whose behalf the request was initiated. This is a mandatory value.

`realm`	Specifies the SAP Ariba Contracts site to which the API request is sent. This is a mandatory value.

Results

If the request is successful, the API deletes the specified contract workspace.

Related Information

Error Messages Returned by the Contract Workspace Management APIs [page 65]

Creating a Baseline Contract in SAP Ariba Contracts Using the Contract Workspace Modification API

Use this procedure to implement the Contract Workspace Modification API in a client application to create a baseline contract in SAP Ariba Contracts.

Prerequisites

- Your site must have guided sourcing enabled. For more information, see Setting Up Guided Sourcing.
- Your site must have SAP Ariba Contracts integrated with SAP Ariba Sourcing.
- Your administrator must enable the **Enable Pre-Award Negotiation In Guided Sourcing events** (`Application.ACM.EnablePreAwardNegotiationInGuidedSourcing`) ICM parameter. For information about this parameter, refer to Enable Pre-Award Negotiation In Guided Sourcing Events.
- To enable the Contract Workspace Modification API to identify the predecessor event information, in addition to the predecessor project information, to the contract whenever applicable, the **Enable predecessor event identification for contracts created from guided sourcing projects** (`IdentifyPredecessorEventforContractsCreatedFromGuidedSourcingProjects`) ICM parameter must be enabled. For more information, refer to Enable predecessor event identification for contracts created from guided sourcing projects.
- Find the ID of the project template that you use to create the procurement contract workspace. The available template IDs are displayed on the dashboard under **Manage** > **Templates**.
- The user whose credentials are passed in the API call must be the initiator of the job or must have permissions to view the job.
- 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.

Context

The Contract Workspace Modification API enables you to create a baseline contract workspace in SAP Ariba Contracts using the `POST /contractworkspaces` endpoint. The baseline contract workspaces can be created using project templates that can be pre-populated with team members, documents, tasks, references to integrated systems, and other project items. The available templates can change depending on the project information (attributes) you specify. Project templates can also contain conditions that determine the documents, tasks, and phases created according to the project attributes.

You can create a query body by specifying a project template ID and the properties required to be populated in the baseline contract workspace.

> ⓘ 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

Invoke the `POST /contractworkspaces` endpoint to create a baseline contract workspace with a system template ID. The template contains all the details related to tasks, documents, teams, and references to integrated systems.

Request type: POST

URL: **`{{runtime_url}}/contractworkspaces`**

The following example creates a baseline contract workspace with the specified details:

> ✤ Example
>
> Request URL: `POST {{runtime_url}}/contractWorkspaces?user=adavis&passwordAdapter=PasswordAdapter1&realm=s4All-7`
>
> Query Body:

```
{
  "title": "Test Search With Spaces",
  "description": "Demo Description",

  "isTestProject": true,

  "hierarchicalType" :"MasterAgreement",
  "parentAgreement":"",
  "templateId": "SYS0052",

  "supplier": {
    "name": "Apex Corporation",
    "systemID": "sid509",
    "organizationIDs": [
      {
        "domain": "internalsystemid",
        "value": "1009"
      },
      {
        "domain": "duns",
```

```
        "value": "700000080"
      },
      {
        "domain": "networkid",
        "value": "an70000000080"
      },
      {
        "domain": "oracle",
        "value": "4009"
      },
      {
        "domain": "psoft",
        "value": "share:usa0000022"
      },
      {
        "domain": "sap",
        "value": "0000001050"
      },
      {
        "domain": "buyersystemid",
        "value": "sid509"
      },
      {
        "domain": "sourcingnetworkid",
        "value": "an70000000080"
      }
    ],
    "address": {
      "name": "",
      "uniqueName": "sid509",
      "phone": "",
      "fax": "",
      "lines": [
        "753 Apex Avenue"
      ],
      "city": "Marlborough",
      "state": "MA",
      "postalCode": "1752",
      "country": "US"
    }
  },
  "affectedParties": [
        {
          "name": "Apex Corporation",
          "systemID": "sid509",
          "smVendorID": null,
          "organizationIDs": [
            {
              "domain": "internalsystemid",
              "value": "1009"
            },
            {
              "domain": "duns",
              "value": "700000080"
            },
            {
              "domain": "networkid",
              "value": "an70000000080"
            },
            {
              "domain": "oracle",
              "value": "4009"
            },
            {
              "domain": "psoft",
              "value": "share:usa0000022"
            },
            {
```

```
            "domain": "buyersystemid",
            "value": "sid509"
          },
          {
            "domain": "sourcingnetworkid",
            "value": "an70000000080"
          },
          {
            "domain": "sap",
            "value": "2000003"
          }
        ],
        "address": {
          "name": "",
          "uniqueName": "sid509",
          "phone": "",
          "fax": "",
          "lines": [
            "753 Apex Avenue"
          ],
          "city": "Marlborough",
          "state": "MA",
          "postalCode": "1752",
          "country": "US"
        }
      }
    ],


  "contractAmount": {
  "amount": 1000,
  "currency": "Euro"
},

"proposedContractAmount":
{
    "amount": 1000,
  "currency": "Euro"
},

    "commodities": [
  {
    "uniqueName": "56",
    "domain": "unspsc",
    "name": "Furniture and Furnishings"
  }
],

"regions": [
  {
    "uniqueName": "APAC",
    "name": "Asia Pacific"
  }
],

"departments": [
  {
    "uniqueName": "100",
    "name": "corporate"
  }
  ],


  "expirationTermType": "Perpetual",

"effectiveDate": "2019-12-09T20:02:58.303+0000",
"expirationDate": "2019-12-31T20:02:58.303+0000",
"agreementDate" :"2020-12-09T20:02:58.303+0000",
```

```
      "owner": {
         "uniqueName": "molson",
         "passwordAdapter": "PasswordAdapter1",
         "name": "Max Olson",
         "emailAddress": "devnull@devmail.ariba.com",
         "organization": "Buyer Organization",
         "orgANId": "AN41000000377",
         "orgName": "Buyer Organization"
      },

  "customFields": [
      {
          "fieldId": "cus_CF_Boolean",
         "fieldName": "CF_Boolean",
         "fieldDataType": "booleanValue",
         "booleanValue": true
      },
      {
          "fieldId": "cus_CF_Text",
         "fieldName": "CF_Text",
         "fieldDataType": "textValue",
          "textValue": [
            "IM UPDATED Again"
          ]
      },
      {
          "fieldId": "cus_eTestA",
         "fieldName": "eTestA",
         "fieldDataType": "textValue",
         "textValue": [
            "Updated Again"
          ]
      }

      ],

        "questions" : [{
      "questionId" : "QI2516",
      "answerText" : "No"
  },{
      "questionId" : "QI2517",
      "answerText" : "No"
  }]
 “accessControl”: [
        “Legal Information”,
        “Private To Team Members”
      ],
  “relatedId” : “12345",
"blockPublish": false,
      "integratedSystemRefs": [
        {
          "integratedSystemName": "Icertis",
          "integratedSystemID": "fg001",
          "status": "Executed",
          "type":"Agreement",
          "urlData": "csow/csow1"
        }
"eventInternalId": "Doc12345678"
"baselineContract": true
}
```

baselineContract The flag that indicates whether the contract workspace being created is a baseline contract. When set to **`true`**, the contract workspace is created as a baseline contract.

eventInternalId The predecessor sourcing project ID to associate with the baseline contract. You must provide a valid predecessor sourcing project ID when `baselineContract` is set to **`true`**.

> **Note**
>
> - If the sourcing project is a single-event guided sourcing project, the contract shows the **Predecessor Project** and the **Predecessor Event** attributes. Both the attributes have a link to the same project.
> - If the sourcing project is a subproject within a guided sourcing full project:
>   - The contract shows the **Predecessor Project** attribute with a link to the guided sourcing full project, and the **Predecessor Event** attribute with a link to the subproject.
>   - The contract is not associated with any other subprojects in the guided sourcing full project.

`accessControl` The access control permission for the contract workspace. For a list of access control values available in SAP Ariba Contracts, see About the PermissionBitVec element in contract exports. The availability of these access control groups depends on the site configurations. Some of these groups may not be available to you.

> **Note**
>
> The value must be specified in **English** irrespective of the locale of your site.

`relatedID` The related ID, for example, the ID for a related record or document, such as an outline agreement or a contract in an external system.

`blockPublish` The flag to specify whether the contract workspace can be published after creation.

`integratedSystemRefs` The external systems (such as Icertis Contract Intelligence (ICI) for SAP Ariba Solutions and SAP Fieldglass) to be integrated with the contract workspace.

> **Note**
>
> You can include multiple `integratedSystemRefs` objects in the request body when you are creating a contract workspace that is integrated with multiple integrated systems.

Results

If the request is successful, a baseline contract with the specified details is created in SAP Ariba Contracts. For a list of response codes, refer to Error Messages Returned by the Contract Workspace Management APIs [page 65].

Next Steps

After a baseline contract is created, you can continue working on the baseline contract from the SAP Ariba Contracts user interface or use the Contract Workspace Management APIs [page 4].

Error Messages Returned by the Contract Workspace Management APIs

The APIs within the Contract Workspace Management APIs returns response codes when you submit a request. The following table describes the response codes:

HTTP response code	Message	Description
400	`The API key passed in the request is not valid.`	The API key passed in the query headers does not match the API key in the OAuth token.
	`Bad Request`	The request was unsuccessful because of one of the following reasons: • The required feature is not enabled. • This is not applicable to the `POST /contractworkspaces` endpoint. • One or more parameters specified in the request are incorrect. • A baseline contract cannot be created without linking a predecessor project to it. Please provide the predecessor project ID in the eventInternalId field and try again. • The following fields cannot be updated : [supplier] • Updating the predecessor project of a baseline contract is not allowed. Please remove the field `eventInternalId` from the request body and try again. • Action 'amend' is currently not supported. • The request was unsuccessful because the parent agreement specified for the subagreement contract does not have a valid hierarchical type.

HTTP response code	Message	Description
401	Unauthorized	The request was unsuccessful because of one of the following reasons: • Invalid expression. • The OAuth token has expired or is invalid. A new OAuth token needs to be generated. • The feature is not enabled.
403	Forbidden	The request was unsuccessful because of one of the following reasons: • Contract data is missing in the request. • You do not have permission to access the requested resource.
404	Not Found	The request was unsuccessful because of one of the following reasons: • Request has one or more invalid contracts. • Your query requests a resource that is not in the system.
429	Rate limit error	You have exceeded the maximum permitted query rate. Refer to the developer portal discovery page for this API for the applicable query rates.
504	Gateway Time-out	The server timeout time has been exceeded, possibly due to network traffic or reliability issues. Retry your query.

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.