

# ERP-137 Integration Process - Award Bid integration from Keelvar to Ariba Sourcing

<b>Status</b>	Approved
<b>Owner</b>	HUYNH-ext, Danh , MOHAMMAD-ext, Farhan
<b>Stakeholders</b>	
<b>Jira Request ID</b>	 ERP-48 - Jira project doesn't exist or you don't have permission to view it.
<b>Jira Development ID</b>	 ERP-137 - Jira project doesn't exist or you don't have permission to view it.

## High- Level Specification

<b>Application System (Source)</b>	Keelvar
<b>Application System (Target)</b>	Ariba Guided Sourcing
<b>Source System Interface</b>	Standard Keelvar Webhooks and APIs
<b>Target System Interface</b>	Standard Ariba APIs
<b>Business Process Reference</b>	03.03.03.01. Manage Sourcing Projects

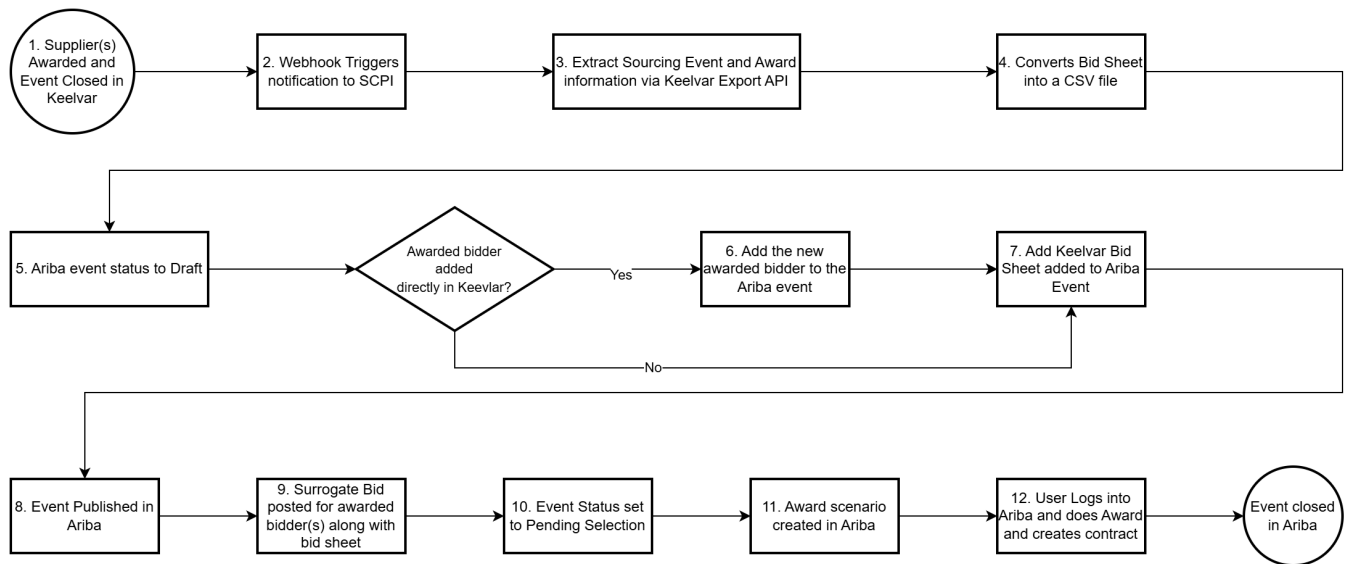
## Functional Overview

Ariba Guided Sourcing is used as the single entry point for all Sourcing Events. If the users can select the Keelvar template to create an Event in Ariba Guided Sourcing, the details of the Event are replicated to Keelvar and any change in Keelvar reflects the status of the Event in Ariba Guided Sourcing. When the user awards one or multiple Bidders or Suppliers in Keelvar, the awarded Bid(s) and Keelvar Bid Sheet must be sent back to Ariba Guided Sourcing.

## Scope and Objectives

The scope of this Functional Specification is to describe the interface between Keelvar and Ariba Guided Sourcing. When an award scenario and awarded bid sheet(s) are created in Keelvar and the event is closed in Keelvar, a Webhook in Keelvar will notify SCPI about the award. After that, SCPI will send the award scenario and awarded bid sheet(s) back to the event in Ariba Guided Sourcing. If any new Bidder is added in Keelvar, the new Bidder and Bidder Contact must also be created in Ariba Guided Sourcing.

## Process Flow Diagram



Step	Description	Comment
1	User awards one or more suppliers or bidders in Keelvar and closes the Event in Keelvar.	
2	A webhook is triggered from Keelvar to SCPI notifying the change along with the Ariba Event ID	
3	SCPI extracts Sourcing Event Feed and Award Feed details from Keelvar Export API	
4	SCPI extracts Keelvar bid sheet and converts it into a CSV file	
5	SCPI sets Ariba event status to Draft	
6	If an awarded Bidder has been added directly in Keelvar, SCPI adds the new bidder to the Ariba event.	Only Suppliers that are available in Ariba can be included to an Event in Keelvar
7	SCPI adds the Keelvar Bid Sheet Item to the Ariba Event	
8	SCPI sets the Ariba Guided Sourcing Event status to Publish	
9	SCPI posts surrogate bid(s) for awarded bidder(s) to Ariba Guided Sourcing Event along with the Bid Sheet as attachment	
10	SCPI sets the Ariba Guided Sourcing Event status to Pending Selection	
11	SCPI creates an Award Scenario with the above sent Bid(s) in Ariba Guided Sourcing Event	
12	User logs into Ariba, awards the bid and creates a contract.	This step is manually executed as part of the Award Process

## Assumptions

- When the event status changes to CLOSED in Keelvar, the same is notified in SCPI via Webhook API
- The users must not reopen the event after they have awarded the Event in Ariba Guided Sourcing. Otherwise, the new award cannot be linked to Ariba Guided Sourcing.
- Ariba APIs use Pacific Standard Time zone as the default time zone for all concerned APIs.

## Dependencies

- Depending on the build of [ERP-224 Supplier Data Synchronization from Ariba to Keelvar](#). This integration synchronises Supplier details between Ariba and Keelvar.
- Depending on the build of [ERP-108 Integration Process - Sourcing event Creation and Update from Ariba Sourcing to Keelvar](#). This integration replicates Events created in Ariba into Keelvar, and initiates the creation process for Suppliers that are attached to the Event, if they do not exist in Keelvar

- Depending on the build of [ERP-138 Event Status Update Interface between Keelvar and Ariba](#). This integration updates the Status of the Event in Ariba during the Awarding process progresses in Keelvar.

## Security, Integrity and Controls

The following are the Security and Authorization considerations for this interface:

- Access to Ariba and Keelvar APIs is limited via a shared secret to secure End Point connection
- Access to API authorization in SAP Ariba Sourcing are being addressed by Ariba standard security controls. Only authorized persons with Ariba administrator's role can access/ change API OAuth Keys.
- Similarly, access to API authorization in Keelvar are being addressed by Keelvar standard security controls. Only authorized persons with Keelvar Organization User role can access/ change the Keelvar authorization token.

## Configuration Requirements

In order to activate the SAP Ariba API for Event Management, following steps need to be completed:

1. Sign in to the SAP Ariba Developer Portal.
2. Click Create application from the home page.
3. Fill out the Create a new application form by entering an application name and description, then click Submit.
4. This generates an Application key that identifies your application within the system. Every API request your application makes must include this key as the value of the apiKey parameter.
5. Ask your organization admin to request API access for your application by displaying the application in My applications and clicking Actions Ask your admin to request API access.
6. A user with the Organization Admin role requests approval for API access.
7. SAP Ariba assesses the request, and once processed and approved, the Organization Admin user receives email with an OAuth client ID for the application.
8. A user with the Organization Admin role generates the OAuth secret and base64-encoded client and secret.

In order to activate Keelvar API, following steps need to be completed:

1. Log into Keelvar Portal
2. Go to Account settings and navigate to API Keys section
3. Click New token button and provide the name and Expiration of the token. Click create to get the token details.

Ariba Guided Sourcing configuration details are described as under:

	Supplementary	Test	Production
Ariba Technical User	R_BTP_ARB_ADMIN		
Ariba Realm ID	745255310-SS-T	744368466-T	744368466
OAuth URL	<a href="https://api-eu.ariba.com/v2/oauth/token">https://api-eu.ariba.com/v2/oauth/token</a>	<a href="https://api-eu.ariba.com/v2/oauth/token">https://api-eu.ariba.com/v2/oauth/token</a>	<a href="https://api-eu.ariba.com/v2/oauth/token">https://api-eu.ariba.com/v2/oauth/token</a>
Request URL	Surrogate Bidding API - <a href="https://eu.openapi.ariba.com/api/sourcing-event-bid/v1/prod">https://eu.openapi.ariba.com/api/sourcing-event-bid/v1/prod</a> Event Management API - <a href="https://eu.openapi.ariba.com/api/sourcing-event/v2/prod/">https://eu.openapi.ariba.com/api/sourcing-event/v2/prod/</a>		

Keelvar configuration is described below:

Request URL	<a href="https://test.keelvar.dev/api">https://test.keelvar.dev/api</a>	<a href="https://my.keelvar.app/api">https://my.keelvar.app/api</a>
-------------	---	---

A webhook for Sourcing Event Feed must be created in Keelvar to receive of the following changes in Sourcing Event:

- The event is closed after all bidding rounds have been completed

A request must be sent to [support@keelvar.com](mailto:support@keelvar.com) and make a whitelist request for the domain of the SCPI system to which the HTTP notifications will be sent.

Creation of Webhook in Keelvar for Sourcing Event Feed requires sending a request as under:

```
curl -X POST https://my.keelvar.app/api/webhooks \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-H 'Authorization: Bearer <YOUR_API_KEY>' \
-d '{
  "name": "Sourcing Event Feed Webhook",
  "description": "Keelvar Sourcing Event Feed Webhook",
  "signing_alg": "HMAC_SHA256",
  "signing_key": "(Signing Key)",
  "triggers": "[SOURCING_EVENTS_FEED_UPDATED]",
  "url": (SCPI webhook listener URL),
}'
```

Keelvar will send the response with the Webhook UUID as under:

```
'description': 'Keelvar Sourcing Event Feed Webhook',  
'name': 'Sourcing Event Feed Webhook',  
'signing_alg': 'HMAC_SHA256',  
'signing_key': '(Signing Key)',  
'triggers': '[SOURCING_EVENTS_FEED_UPDATED]',  
'url': (SCPI webhook listener URL),  
'uuid': '4a55a230-307b-4bab-ac53-e95186b88968'
```

## Special Requirements

Not applicable

## Design Rationale

Since Sourcing Events are originally initiated in SAP Ariba Guided Sourcing and integrated to Keelvar, when the award scenario is completed and awarded bid sheet are created in Keelvar, they must be sent back in SAP Ariba Guided Sourcing. To create the Award Scenario in Ariba, SAP CPI implements an orchestration consisting of multiple API calls, mapping and validation using two APIs in Ariba and one in Keelvar. Since Event Management Service is one of the Ariba APIs used, this orchestration utilises the [JMS Queue Framework](#) used in [ERP-108](#) and [ERP-138](#).

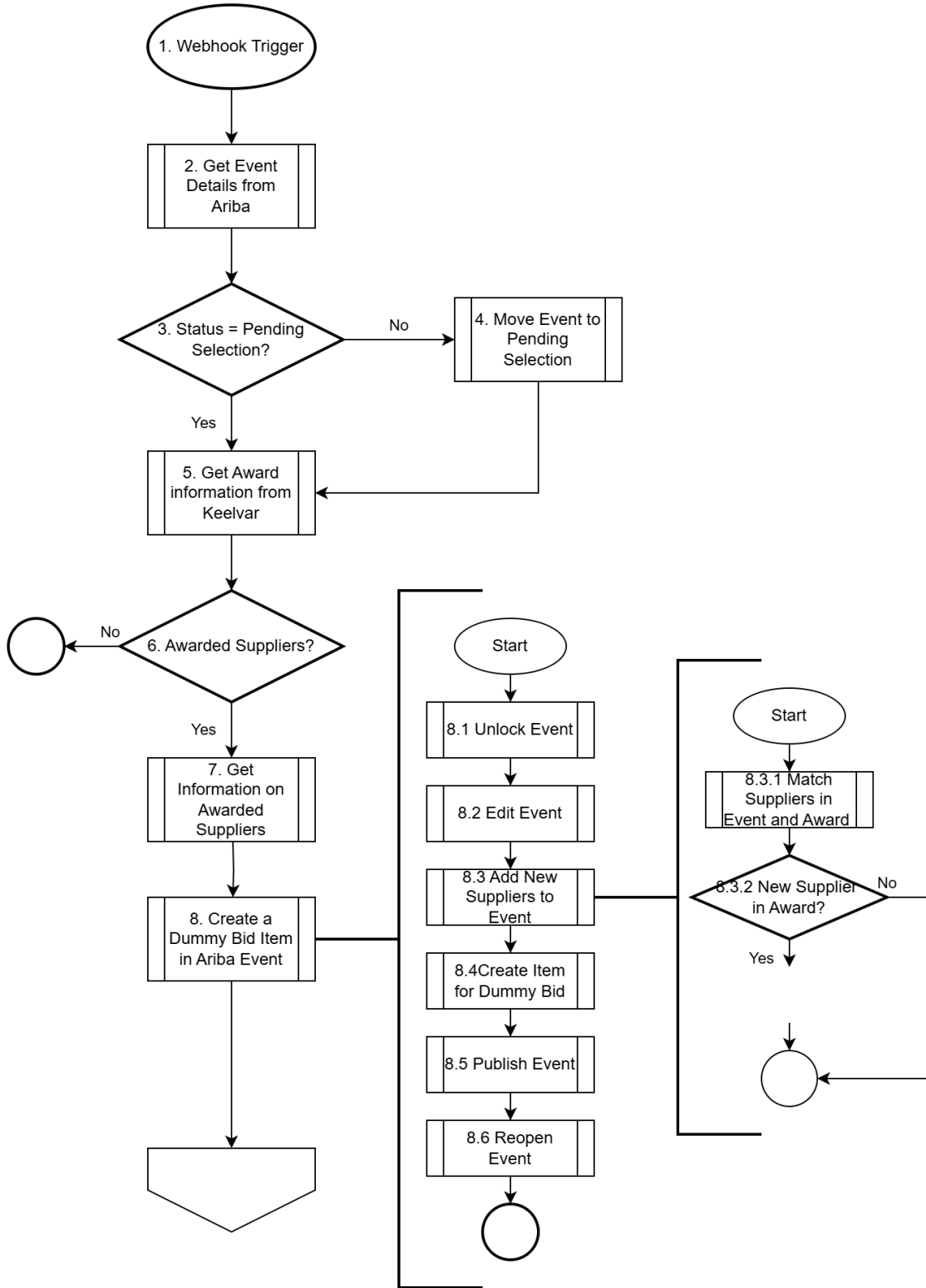
The Orchestration consists of 3 main activities:

Name	Description
Get Keelvar Award Data	Initiated by Keelvar Webhook, which is generated when the Award scenario is completed in Keelvar. If the Event in Ariba is in the expected "Pending Selection" state, then details of the Award, the awarded suppliers and the Contact Person from the Award are first extracted from Keelvar. In the event of a split award, the split percentage is calculated per awarded supplier. The split percentage is the percentage amount of the cumulative award value per supplier divided by total award amount. If required, any awarded suppliers are added to the Event in Ariba before progressing further. A dummy Item is added into the Ariba Event being Awarded to enable Surrogate Bidding in next step
Generate Bid Sheets	Surrogate Bidding is initiated for each Award to a Supplier. This involves exporting and Importing Bid Sheet (in EXCEL format) to the Event. Once this is completed, Keelvar Bid Sheet converted to a CSV file is attached to the Surrogate Bids. Once this step is completed, the payload containing the Award details and Supplier details are passed to the next step
Create Award Scenario	Finally, the Ariba Event is stopped from doing further bidding activities. Using the Award details received from Keelvar, an Award Scenario is created and uploaded to the Event.

The following Flow Diagrams shows the integration steps that are executed to complete the Award through the above 3 steps.

## IFlow to Get Keelvar Award Data

IFlow: Get Keelvar Award Data

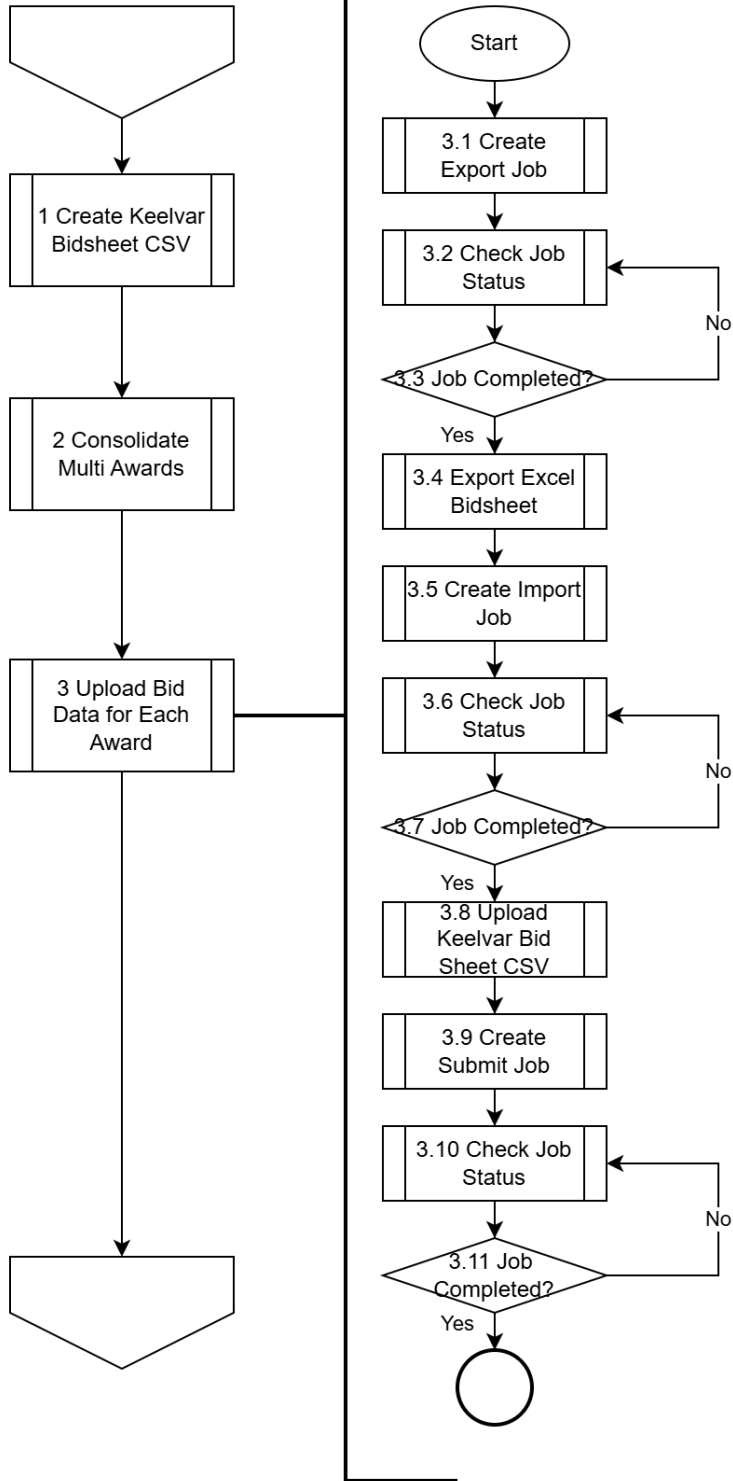


Step	Name	Additional Details	API Usage
1	Award Notification triggered from Keelvar via Webhook	See <a href="#">NFR-2006</a> for details on Keelvar Webhook	
2	Get Event Details	Details of the Event including Supplier Invitations are extracted from Ariba	Ariba

	from Ariba		Event Management API
3	Check Status equals "Pending Selection"	Event status is Ariba must be in "Pending selection" and the document locked by BTP admin	
4	If not in Pending Selection, move the Event to Pending Selection and lock to CPI User	If the status is not in Pending selection, CPI must change the event status and lock the document (CR-0273)	Ariba Event Management API
5	Get Award Information from Keelvar	Using the Keelvar ID received from the Webhook, the Award information is extracted from Keelvar	Keelvar Export API
6	Awarded Suppliers?	If the event has been closed without any awards, the integration ends	
7	Get Information on Awarded Suppliers	Looping through the list of suppliers, details about each Supplier is extracted from Keelvar. In the event of a split award, the split percentage is calculated per awarded supplier. The split percentage is the percentage amount of the cumulative award value per supplier divided by total award amount. This is necessary since Suppliers that are not initially assigned to Ariba Event may have been awarded in Keelvar	Keelvar Supplier Management API
8	Initiate the creation of a Dummy Bid Item in Ariba	See the sub steps below	
8.1	Unlock Event	Action to Unlock Document is initiated	Ariba Event Management API
8.2	Edit Event	Action to Edit Ariba Event is initiated	Ariba Event Management API
8.3	Add New Suppliers to Event	See sub steps below	
8.3.1	Match Suppliers in Event and Award	Check if the Suppliers in the Award received from Keelvar are already assigned in Ariba	
8.3.2	New Suppliers in Award?	If no, skip rest of the sub steps	
8.3.3	Add Supplier to Event	New Suppliers and their Contacts are added to the Ariba Event. The Supplier and Contact need to exist in Ariba for this to succeed	Ariba Event Management API
8.4	Create Item for Dummy Bid	A new Item is created in Ariba Event to accept the Bid sheet from Ariba	Ari8
8.5	Publish Event	Action to Publish the Event is initiated	Ariba Event Management API
8.6	Reopen Event	Action to Reopen the Event is initiated	Ariba Event Management API

## IFlow to Generate Bid Sheets

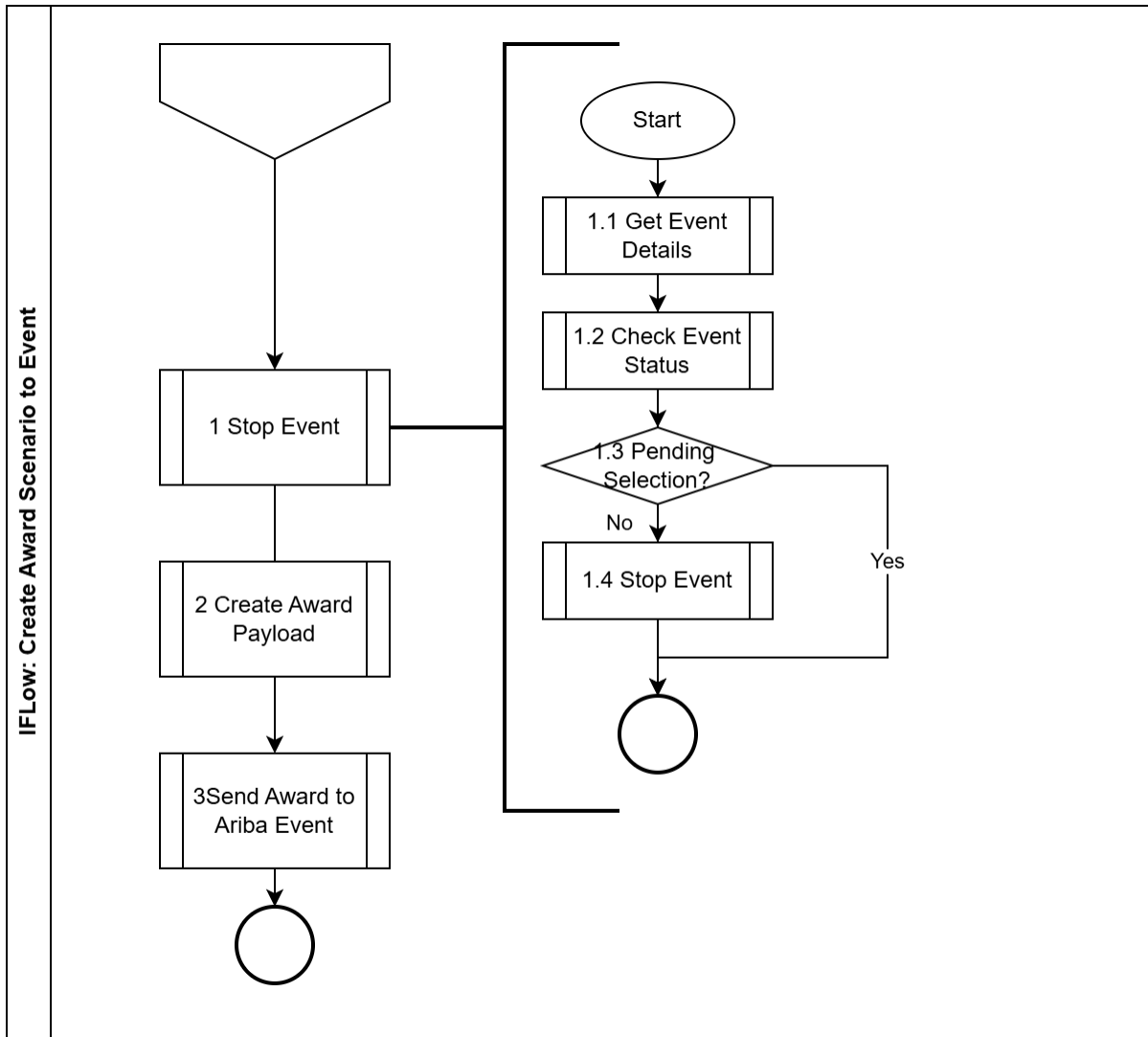
IFlow: Generate Bid Sheets



Step	Name	Additional Details	API Usage
1	Create Keelvar Bidsheet CSV	Transform Keelvar BidSheet data to a CSV file to be uploaded to Ariba	
2	Consolidate Awards by Supplier	Multiple Lots (line items) in the Keelvar Award are combined based on the supplier	
3	Upload Bid Data for Each Award	See sub steps below, executed for each record in the Keelvar Award	

3.1	Create Export Job	Request Ariba to create an Export Job for Bidsheet in Excel format	Ariba Surrogate Bidding API
3.2	Check Job Status	Check if the above job is completed.	Ariba Surrogate Bidding API
3.3	Job Completed?	If status is not Completed, pause and recheck (step 2.2) a defined number of times. Exception raised if job status does not change after the number of iteration	
3.4	Export Excel Bidsheet	Export the Ariba Excel Bidsheet. The contents are not updated in CPI	Ariba Surrogate Bidding API
3.5	Create Import Job	Request Ariba to create an Import Job for the Bidsheet in Excel format received in step 2.4	Ariba Surrogate Bidding API
3.6	Check Job Status	Check if the above job is completed.	Ariba Surrogate Bidding API
3.7	Job Completed?	If status is not Completed, pause and recheck (step 2.6) a defined number of times. Exception raised if job status does not change after the number of iteration	
3.8	Upload Keelvar BidSheet CSV	The Keelvar Bidsheet created in Step 1 is attached to the Award	Ariba Surrogate Bidding API
3.9	Create Submit Job	Request Ariba to create a Submit Job for Keelvar Bidsheet	Ariba Surrogate Bidding API
3.10	Check Job Status	Check if the above job is completed.	Ariba Surrogate Bidding API
3.11	Job Completed?	If status is not Completed, pause and recheck (step 2.10) a defined number of times. Exception raised if job status does not change after the number of iteration	

### IFlow to Create Award Scenario



Step	Name	Additional Details	API Usage
1	Stop Event	See sub steps below	
1.1	Get Event Details	Details of the Event including Supplier Invitations are extracted from Ariba	Ariba Event Management API
1.2	Check Event Status	Get Details of the Event and get the current status of the Event	
1.3	Status = Pending Selection?	If the Event is in Pending Selection Status, go to step 2	
1.4	Stop Event	If not in Pending Selection, initiate Stop Action on the Event	Ariba Event Management API
2	Create Award Payload	Create the Award Scenario Payload	
3	Send Award to Ariba Event	Send the Award Scenario to Ariba	Ariba Event Management API

## API use

### Ariba API Specifications

Creation of API Applications in Ariba API is executed in the EU tenant. Ariba events are timestamped on Pacific Standard Time (PST), therefore time stamps selected

[SAP Ariba APIs Portal](#) (Logon Required)

API	Used Endpoints	Used for
Strategic Sourcing - Event Management API	/events/*	Get Event Details Post Supplier Invitations
Surrogate Bidding API	/jobs	Change Event Status

## Data Structures

### Event Status Update Mapping

#### Source Structure

The following Keelvar Award API fields will be used to provide the required information for this interface:

Field	Description
awards/tags/{name=Ariba Event ID}/value	The Ariba Event ID
awards/suppliers/external_id	The System ID of the Bidder or Supplier
awards/awards/cost	The value for the awarded lot
sourcing_events/rounds/awarded_cost	Total award value of the sourcing event
awards/sourcing_event	The UUID of the Keelvar Sourcing Event
awards/fields/Contact ID	The Contact ID of the Bidder or Supplier Contact

#### Target Structure

The following Ariba fields will be used to provide the required information for this interface:

Field	Description
eventId or supplierBids/eventId	The Ariba Event ID
supplierBids/winningSplitValue	The decimal value of the share awarded to the Bidder or the supplier. The range is between 0 and 1.
actionName	The name of the operation to change the state of the Ariba Event.



/feeds/sourcing-events	tags/(name=Ariba Event ID)/value	Y	The Ariba Event ID	events/{eventID}/items	eventID	Y	The Ariba Event ID	connection
Static Value	Static Value = Ariba Technical User	Y	Technical User Name	events/{eventID}/items	user	Y	Technical User Name	connection
Static Value	Static Value = ThirdPartyUser	Y	Technical User PasswordAdapter	events/{eventID}/items	PasswordAdapter	Y	Technical User PasswordAdapter	connection
Static Value	Static Value = Realm ID	Y	The Ariba Guided Sourcing Realm ID	events/{eventID}/items	realm	Y	The Ariba Guided Sourcing Realm ID	connection
Static Value	Static Value = Keelvar Bid Sheet	Y	The Name of the Bid Sheet item	events/{eventID}/items	title	Y	The Name of the Bid Sheet item	payload
Static Value	Static Value = 4	Y	The Type of the Bid Sheet item (Line Item Type)	events/{eventID}/items	itemType	Y	The Type of the Bid Sheet item (Line Item Type)	payload
Static Value	Static Value =Award Percentage	Y	The Term name for Award Percentage	events/{eventID}/items	terms/title	Y	The Term name for Award Percentage	payload
Static Value	Static Value = Bid	Y	The Term name for Bid	events/{eventID}/items	terms/title	Y	The Term name for Bid	payload
Static Value	Static Value = 100	Y	The initial value for quantity	events/{eventID}/items	terms/value/quantity/Value/amount	Y	The initial value for quantity	payload
Static Value	Static Value = %	Y	the unit of measure of quantity (Percentage)	events/{eventID}/items	terms/value/quantity/Value/unitofMeasure	Y	the unit of measure of quantity (Percentage)	payload
Static Value	Static Value = Keelvar Bid Sheet Attachment	Y	The Term Name for Bid Sheet attachment	events/{eventID}/items	terms/title	Y	The Term Name for Bid Sheet attachment	payload

The following Mapping Table is applicable to trigger a Job required to send a Surrogate Bid to Ariba Guided Sourcing Event. The Surrogate Bidding API is used for all requests:

API or Portlet Name	Source Field	Required (Y/N)	Description	API or Portlet Name	Target Field	Required (Y/N)	Description	Rule Type	Rule Instruction
Static Value	Static Value = Ariba Technical User	Y	Technical User Name	jobs	user	Y	Technical User Name		connection
Static Value	Static Value = ThirdPartyUser	Y	Technical User PasswordAdapter	jobs	Password Adapter	Y	Technical User PasswordAdapter		connection
Static Value	Static Value = Realm ID	Y	The Ariba Guided Sourcing Realm ID	jobs	realm	Y	The Ariba Guided Sourcing Realm ID		connection
Static Value	Static Value = Allowed Operation are: 1. EXPORT 2. IMPORT 3. ADD_ATTACHMENT_RESPONSE 4. SUBMIT	Y	The Operation which has to be done for the Surrogate Bidding	jobs	operation	Y	The Operation which has to be done for the Surrogate Bidding		connection
/feeds/awards	tags/(name=Ariba Event ID)/value	Y	The Ariba Event ID	jobs	eventId	Y	The Ariba Event ID		connection - URL
manage/suppliers	fields/Contact ID	Y	Ariba Bidder Contact ID	jobs	supplierId	Y	Ariba Bidder Contact ID		payload
Previous Request	Base64 encoded file	Y	The Exported Bid Sheet from Keelvar which has been converted into a CSV file	jobs	file	Y	The Exported Bid Sheet from Keelvar which has been converted into a CSV file		payload
Static Value	Static Value = "1"	Y	The Item Number for the Keelvar Bid Sheet Item	jobs	itemNumber	Y	The Item Number for the Keelvar Bid Sheet Item		payload

The following Mapping Table is applicable to get the status of the Job required to send a Surrogate Bid to Ariba Guided Sourcing Event. The Ariba Event Management API is used for all requests:

API or Portlet Name	Source Field	Required (Y/N)	Description	API or Portlet Name	Target Field	Required (Y/N)	Description	Rule Type	Rule Instruction
		Y	The Job ID retrieved from			Y	The Job ID retrieved		

Previous Request	jobld		the previous request	jobs/{jobld}	jobld		from the previous request		
Static Value	Static Value = Ariba Technical User	Y	Technical User Name	jobs/{jobld}	user	Y	Technical User Name		
Static Value	Static Value = ThirdPartyUser	Y	Technical User PasswordAdapter	jobs/{jobld}	Password Adapter	Y	Technical User PasswordAdapter		
Static Value	Static Value = Realm ID	Y	The Ariba Guided Sourcing Realm ID	jobs/{jobld}	realm	Y	The Ariba Guided Sourcing Realm ID		

The following Mapping Table is applicable to export the Surrogate Bid response file:

API or Portlet Name	Source Field	Required (Y/N)	Description	API or Portlet Name	Target Field	Required (Y/N)	Description	Rule Type	Rule Instruction
Previous Request	jobld	Y	The Job ID retrieved from the previous request	jobs/{jobld}	jobld	Y	The Job ID retrieved from the previous request		
Previous Request	fileld	Y	The File ID retrieved from the previous request	jobs/{jobld}	fileld	Y	The File ID retrieved from the previous request		
Static Value	Static Value = Ariba Technical User	Y	Technical User Name	jobs/{jobld}	user	Y	Technical User Name		
Static Value	Static Value = ThirdPartyUser	Y	Technical User PasswordAdapter	jobs/{jobld}	Password Adapter	Y	Technical User PasswordAdapter		
Static Value	Static Value = Realm ID	Y	The Ariba Guided Sourcing Realm ID	jobs/{jobld}	realm	Y	The Ariba Guided Sourcing Realm ID		

The following Mapping Table is applicable to create a manual Award Scenario in Ariba Guided Sourcing Event. The Ariba Event Management API is used for all requests:

API or Portlet Name	Source Field	Required (Y/N)	Description	API or Portlet Name	Target Field	Required (Y/N)	Description	Rule Type	Rule Instruction
	tags(name="Ariba Document ID")/value	Y	The Ariba Event ID	/events/{eventId}/scenarios	eventId	Y	The Ariba Event ID		
/feed/awards	Concatenation of Static value = "Keelvar Scenario" and rounds /name	Y	Name of the Manual Scenario	/events/{eventId}/scenarios	title	Y	Name of the Manual Scenario		
Static Value	Static value = "0"	Y	Scenario Type i.e Manual	/events/{eventId}/scenarios	scenarioType	Y	Scenario Type i.e Manual		
Static Value	Static value = "Primary"	Y	Bid Type	/events/{eventId}/scenarios	supplierBids/bidType	Y	Bid Type	Loop	Loop for each awarded bidder in case of split bid
event={EventID}	itemID(title="Bid sheet from Keelvar")	Y	The Item ID of the Bid sheet from Keelvar item	/events/{eventId}/scenarios	supplierBids/itemId	Y	The Item ID of the Bid sheet from Keelvar item	Loop	Loop for each awarded bidder in case of split bid
/feeds/awards	tags(name="Ariba Document ID")/value	Y	The Ariba Event ID	/events/{eventId}/scenarios	supplierBids/eventId	Y	The Ariba Event ID	Loop	Loop for each awarded bidder in case of split bid
manage/suppliers	suppliers/external_id	Y	Contact ID of the Bidder	/events/{eventId}/scenarios	supplierBids/invitationId	Y	Contact ID of the Bidder	Loop	Loop for each awarded bidder in case of split bid
Static Value	Static value = "1"	Y	Award By Percentage	/events/{eventId}/scenarios	supplierBids/winningSplitType	Y	Award By Quantity	Loop	Loop for each awarded bidder in case of split bid
Calculated field	Sum (awards/awards/cost)/sourcing_events/rounds/awarded_cost*100	Y	Percentage Award	/events/{eventId}/scenarios	supplierBids/winningSplitValue	Y	Percentage Award	Loop; Multiple	Loop for each awarded bidder in case of split bid and/or multiple lots; The split percentage is the percentage amount of the cumulative award value per supplier divided by total award amount

## Processing Logic

### Processing within Source

A webhook should be triggered to complete Event closure activities which are described as under:

1. Unlock the Ariba Event. Set the status to draft (EDIT operation) to add the Keelvar Bid Sheet Item
2. Add the Supplier Invitation to the Ariba Event (if applicable)
3. Add Keelvar Bid Sheet to the Ariba Event
4. Set the status to publish and reopen the event when the Bid Sheet Item have been added to the event
5. Submit the surrogate Bid(s)
6. Set the status to Pending Selection (STOP operation) once the Surrogate Bid has been submitted
7. Create award scenario in Ariba Guided Sourcing

## Processing within Middleware

### Trigger for Sourcing Event Feed Webhook in Keelvar

Once webhook is triggered in Keelvar, the SCPI should receive a notification as under:

```
{
  "trigger": "AWARD_FEED_UPDATED",
  "timestamp": "2024-11-04T11:13:56.619494Z",
  "payload": {
    "sourcing_event": "19e44441-81c1-11f0-b2f5-f884135dd221",
    "status": "CLOSED",
    "name": "My Event",
    "sourcing_requests": ["c819a007-bac7-4172-eb1d-272a9933f1a2"],
    "tags": [
      {
        "name": "Commodity",
        "value": "0005 OFFICE FURNITURE"
      },
      {
        "name": "GBU",
        "value": "TS & MATERIALS"
      },
      {
        "name": "Ariba Event ID",
        "value": "Doc123456789"
      }
    ]
  }
}
```

Then, a request has to be made to Keelvar to fetch Sourcing Event Details using the Sourcing Event UUID from the above notification as under:

```
curl --location 'https://test.keelvar.dev/api/feeds/sourcing-events?id= request UUID from above' \
--header 'Authorization: Bearer YOUR_API_TOKEN';
```

The response from Keelvar should contain the Sourcing Event Details which will be used as source in the subsequent process.

Sample JSON for Payload is available here:

### Edit Event Status in Ariba Guided Sourcing

SCPI must create a payload based on the Mapping Table above to update the status of the Ariba Guided Sourcing Event.

Sample JSON for Payload is available here:

Unlock -

Edit -

Publish -

Stop -

Reopen -

Then, a sample request to update the Job Status in Ariba Guided Sourcing is as below:

```
curl --location 'https://eu.openapi.ariba.com/api/sourcing-event/v2/prod//jobs?realm=$realmId&user=$aribaTechnicalUser&passwordAdapter=ThirdPartyUser' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'apiKey: <API KEPY>' \
--header 'Authorization: Bearer ARIBA_TOKEN' \
--data 'based on JSON above'
```

### Add Supplier Invitation to Event in Ariba Guided Sourcing

SCPI must create a payload based on the Mapping Table above to add the Supplier Invitation to the Ariba Guided Sourcing Event. This needs to be done only if the Bidder has been added to the Keelvar Event directly in Keelvar.

Sample JSON for Payload is available here:

Then, a sample request to add the supplier Invitation in Ariba Guided Sourcing is as below:

```
curl --location 'https://eu.openapi.ariba.com/api/sourcing-event/v2/prod/events/$eventID/items/supplierInvitations?realm=$realmId&user=$aribaTechnicalUser&passwordAdapter=ThirdPartyUser' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'apiKey: <API KEY>' \
--header 'Authorization: Bearer ARIBA_TOKEN' \
--data 'based on JSON above'
```

### **Add Keelvar Bid Sheet to Event in Ariba Guided Sourcing**

SCPI must create a payload based on the Mapping Table above to add the Bid Sheet Item to the Ariba Guided Sourcing Event.

Sample JSON for Payload is available here:

Then, a request to add Bid Sheet Item to Ariba Guided Sourcing must be made as under:

```
curl --location 'https://eu.openapi.ariba.com/api/sourcing-event/v2/prod/events/Doc2698789795/items?showTotal=true&inheritTerms=false&inheritCommodityAndRegion=true&removeEmptyOwnerTerms=false&realm=$realmId&user=$aribaTechnicalUser&passwordAdapter=ThirdPartyUser' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'apiKey: ARIBA_API_KEY';
--header 'Authorization: Bearer ARIBA_TOKEN';
--data 'As per Payload above'
```

### **Download the Keelvar Bid Sheet and Convert it into csv file**

A request is made to retrieve Award feed from Keelvar for the Sourcing Event from the Webhook Trigger Notification as under:

```
curl --location 'https://test.keelvar.dev/api/feeds/awards?sourcing_event=$event_uuid' \
--header 'Authorization: Bearer Bearer API_TOKEN';
```

The response from Keelvar is similar to this:

This response must be converted into a csv file. For each object in the award feed a row-entry should be added in the csv file. The CSV file should have following columns:

1. Awarded Supplier (The award feed contains only the UUID of the supplier. The external \_id (Ariba ID) of the supplier should be retrieved from Sourcing Event feed response using the Supplier UUID)
2. share\_awarded
3. Each column under the bid\_sheet\_values object

### **Calculation for multiple lots awarded to a Supplier**

In the event of a split award, the split percentage is calculated per awarded supplier. The split percentage is the percentage amount of the cumulative award value per supplier divided by total award amount.

### **Submit a Surrogate Bid to the Event in Ariba Guided Sourcing**

Surrogate Bids must be submitted for each of the Bidders that have been awarded in Keelvar. The processing of a Surrogate Bid to Ariba is described below. The Ariba Event ID is needed for each of the following steps:

1. An Export Job must be created in Ariba to export the Bid response sheet. Ariba send the Export Job ID as the response
2. Check the status of the Export Job. Ariba sends the status of the Export Job ID as well as the File ID as the response
3. If the Job is completed, send a request to Download the Bid response sheet. This request requires the Export Job ID as well as File ID as a part of the request parameter. Ariba sends a Base 64 encoded file as response.
4. An Import Job must be created in Ariba to import the Bid response sheet. The above downloaded file must be added in the body of the request. Ariba sends the Import Job ID as the response.
5. Check the status of the Import Job.
6. If the Job is completed, send a request to Add attachment to the Event Item. Send the Keelvar Bid Sheet which has been converted into a csv file as a part of the body of this request. Ariba sends the Add Attachment Job ID as the response.
7. Check the status of the Add Attachment Job
8. If the Job is completed, send a request to submit the Surrogate Bid. Ariba will send the Submit Job ID as response.
9. Check the status of the Submit Job.

The request for Export, Import, Add Attachment and Submit Jobs must be sent in a form format. The form details for each of the above steps are provided here:

The request for Jobs to be sent as under:

```
curl --location 'https://eu.openapi.ariba.com/api/sourcing-event-bid/v1/prod/jobs?realm=$realmId&user=$aribaTechnicalUser&passwordAdapter=ThirdPartyUser' \
--header 'Content-Type: multipart/form-data' \
--header 'Accept: application/json' \
--header 'apiKey: ARIBA_API_KEY';
--header 'Authorization: Bearer ARIBA_TOKEN';
--form 'form Data as described above';
```

In response, Ariba sends back the Id of the Job.

#2

The request for Job status can be sent as under:

```
curl --location 'https://eu.openapi.ariba.com/api/sourcing-event-bid/v1/prod/jobs/$jobId?realm=$realmId&user=$aribaTechnicalUser&passwordAdapter=ThirdPartyUser' \
--header 'Accept: application/json' \
--header 'apiKey: ARIBA_API_KEY';
--header 'Authorization: Bearer ARIBA_TOKEN';
```

In response, Ariba send back the Id of the Job. It also includes the FileId if the operation of the Job request was Export.

#3

The request to download the Ariba Bid response sheet can be sent as under:

```
curl --location 'https://eu.openapi.ariba.com/api/sourcing-event-bid/v1/prod/jobs/$jobId/files/$fileId?realm=$realmId&user=$aribaTechnicalUser&passwordAdapter=ThirdPartyUser' \
--header 'Accept: application/json' \
--header 'apiKey: ARIBA_API_KEY';
--header 'Authorization: Bearer ARIBA_TOKEN';
```

#4

The request to Import Ariba Bid Response Sheet must be sent in a form format. The form details for each of the above steps are provided here:

The request for Jobs to be sent as under:

```
curl --location 'https://eu.openapi.ariba.com/api/sourcing-event-bid/v1/prod/jobs?realm=$realmId&user=$aribaTechnicalUser&passwordAdapter=ThirdPartyUser' \
--header 'Content-Type: multipart/form-data' \
--header 'Accept: application/json' \
--header 'apiKey: ARIBA_API_KEY';
--header 'Authorization: Bearer ARIBA_TOKEN';
--form 'form Data as described above';
```

#5

The request for Job status can be sent as under:

```
curl --location 'https://eu.openapi.ariba.com/api/sourcing-event-bid/v1/prod/jobs/$jobId?realm=$realmId&user=$aribaTechnicalUser&passwordAdapter=ThirdPartyUser' \
--header 'Accept: application/json' \
--header 'apiKey: ARIBA_API_KEY';
--header 'Authorization: Bearer ARIBA_TOKEN';
```

In response, Ariba send back the status of the Job. If the Import is successful, move on to the next step.

#6

The request to Add Attachment (Keelvar Bid Sheet) to Ariba Bid must be sent in a form format. The form details for each of the above steps are provided here:

The request for Jobs to be sent as under:

```
curl --location 'https://eu.openapi.ariba.com/api/sourcing-event-bid/v1/prod/jobs?realm=$realmId&user=$aribaTechnicalUser&passwordAdapter=ThirdPartyUser' \
--header 'Accept: application/json' \
--header 'apiKey: ARIBA_API_KEY';
--header 'Authorization: Bearer ARIBA_TOKEN';
--form 'form Data as described above';
```

In response, Ariba sends back the Id of the Job.

#7

The request for Job status can be sent as under:

```
curl --location 'https://eu.openapi.ariba.com/api/sourcing-event-bid/v1/prod/jobs/$jobId?realm=$realmId&user=$aribaTechnicalUser&passwordAdapter=ThirdPartyUser' \
--header 'Accept: application/json' \
--header 'apiKey: ARIBA_API_KEY';
--header 'Authorization: Bearer ARIBA_TOKEN';
```

In response, Ariba send back the status of the Job. If the Import is successful, move on to the next step.

#8

The request to Submit Bid must be sent in a form format. The form details for each of the above steps are provided here:

The request for Jobs to be sent as under:

```
curl --location 'https://eu.openapi.ariba.com/api/sourcing-event-bid/v1/prod/jobs?realm=$realmId&user=$aribaTechnicalUser&passwordAdapter=ThirdPartyUser' \
--header 'Content-Type: multipart/form-data' \
--header 'Accept: application/json' \
--header 'apiKey: ARIBA_API_KEY';
--header 'Authorization: Bearer ARIBA_TOKEN';
--form 'form Data as described above';
```

In response, Ariba sends back the Id of the Job.

#9

The request for Job status can be sent as under:

```
curl --location 'https://eu.openapi.ariba.com/api/sourcing-event-bid/v1/prod/jobs/$jobId?realm=$realmId&user=$aribaTechnicalUser&passwordAdapter=ThirdPartyUser' \
--header 'Accept: application/json' \
--header 'apiKey: ARIBA_API_KEY';
--header 'Authorization: Bearer ARIBA_TOKEN';
```

In response, Ariba send back the status of the Job. If the status is successful, the surrogate bid has been submitted

### **Create award scenario in Ariba Guided Sourcing**

Then, the payload for Award Scenario creation must be done based on the Mapping Table above. The payload should be similar to this one:

In order to create an award scenario, a request must be sent to Ariba Event Management API as under:

```
curl --location 'https://eu.openapi.ariba.com/api/sourcing-event-bid/v1/prod/events/$eventId/scenarios?realm=$realmId&user=$aribaTechnicalUser&passwordAdapter=ThirdPartyUser' \
--header 'Content-Type: multipart/form-data' \
--header 'Accept: application/json' \
--header 'apiKey: ARIBA_API_KEY';
--header 'Authorization: Bearer ARIBA_TOKEN';
--data 'as per payload described above';
```

### Logic for calculation of winning split value

CPI will iterate through each and every lot which is awarded and group them by awarded suppliers

Winning split value per Supplier = Total award cost per supplier (Sum of cost of all the awarded lots)/(Total award cost) \*100

## **Processing within Target**

Not Applicable

## **Interface Dependency**

- Depending on the build of [ERP-108 Integration Process - Sourcing event Creation and Update from Ariba Sourcing to Keelvar](#)
- Depending on the build of [ERP-138 Event Status Update Interface between Keelvar and Ariba](#)

## **Interface Constraints**

Rate Limit for Ariba Surrogate Bid API - No specific API limits in place for this API but dynamic run time throttling is in effect.

Rate Limit for Ariba Event Management API - (Requests): 5/second, 80/minute, 3500/hour

Rate Limit for Keelvar api/feeds/awards - Burst (60/min), sustained (43,200/day)

Rate Limit for Keelvar api/feeds/sourcing-events - Burst (60/min), sustained (43,200/day)

## Delivery Requirements

Not Applicable

## Delta or Full Load Requirements

All requests for creation and update of sourcing events are done via delta load since last run.

## Interface Alert & Monitoring

The following should be monitored:

- API endpoints are available.
- Keelvar token expiration.
- Failures on inbound processing and outbound side need to be handled . Please check the Error Handling section.
- Messages are processed in reasonable time (XX elapse time).

## Interface Reporting

Not Applicable

## Language Requirements

Not Applicable

## User Interface Requirements

Not Applicable

## Volumetrics

10 events per day

## Performance Consideration

Not Applicable

## Error Handling

- SAP Cloud ALM (CALM) will be used to capture integration execution errors.
- To correct the errors, the administrator must ensure that the data inconsistency is resolved in the concerned system and then must resend the request again to the target system.
- Refer to the link below to troubleshoot Error Messages Returned by the Event Management API.

[Error Messages Returned by the Event Management API](#)

- Refer to the Link below to troubleshoot Error Message Returned by Keelvar API

[Open API specification](#)

## Testing

### How to Test

- Create and select Award Scenario in Keelvar
- Close event in Keelva

## Test Conditions and Expected Results

ID	Condition	Expected Results
1	Create a RFP in Ariba Guided Sourcing includes populating all required header fields and inviting suppliers/bidders into the event	Header fields of RFP are populated and suppliers /bidders are added into event
2	Publish RFP in Ariba Guided Sourcing	RFP is published in Ariba Guided Sourcing with "Published" state and open for editing  RFP is integrated to Keelvar via API
3	Open the event linked to RFP created in Ariba	The event is opened
4	Create Bid sheet and Cost Calculation in Keelvar	Bid sheet and Cost Calculation are created in Keelvar
5	Add new bidders directly in the event in Keelvar	New bidders are added into event in Keelvar
6	Publish event in Keelvar	Event status is "Published" in Keelvar  RFP state in Ariba Guided Sourcing is "Published", but locked for editing
7	Bidders submit bid in Keelvar	Receive bid for the event in Keelvar
8	Create Award Scenario in Keelvar	Award Scenario is created in Keelvar
9	Select Award Scenario in Keelvar and award bidders (includes new added bidders in Keelvar)	Award Scenario is selected in Keelvar
10	Close event in Keelvar	Event status is changed to "Closed" in Keelvar
11	New awarded Bidder Organization and Organization User are automatically created in Ariba Guided Sourcing via API	New Bidder Organization and Organization User are created in Ariba Guided Sourcing
12	Awarded Bid sheet in Keelvar is automatically sent to event in Ariba Guided Sourcing as Surrogate Bid via API	Awarded Bid sheet is added for awarded suppliers /bidder in Ariba Guided Sourcing
13	Event status is changed to "Stop" in Ariba Guided Sourcing via API	Event is automatically stopped in Ariba Guided Sourcing
14	Award Scenario in Keelvar is integrated to the event in Ariba Guided Sourcing via API	Award Scenario is automatically created in Ariba Guided Sourcing
15	Award Supplier/Bidder in Ariba Guided Sourcing	Supplier/Bidder is awarded in Ariba Guided Sourcing
16	Event status is changed to "Completed" in Ariba Guided Sourcing	Event is completed in Ariba Guided Sourcing

## Test Considerations/Dependencies

- Depends on the test condition of [ERP - 108 Sourcing event creation and update integrated between Ariba and Keelvar](#)
- Depends on the test condition of [ERP - 137 Award Bid integrated from Keelvar to Ariba](#)

## Other Information

Not Applicable

## Development Details

### Package

Package Name	Parent Package

## Other Objects

Object Type	Object Name	Configured Parameters				Comments
		Name	Value-Development	Value - Test	Value - Production	
IFlow	Ariba_Surrogate_Bidding_API_Connector	Ariba Base URL	eu.openapi.ariba.com/api/sourcing-event-bid/v1/prod			Base URL path for Surrogate Bidding API
		Credential Name	OAuth_Ariba_Sourcing_SurrogateBiddingAPI			Security Material Containing Client ID and Secret
		API Delay	2			Delay when Rate Limit is reached
		Ariba API key	APIKEY_AribaSurrogateBidding_CPI_Dev			Security Material containing API Key for application
		Ariba Password Adapter	ThirdPartyUser			
		Ariba Realm	745255310-SS-T	744368466-T	744368466	
		Ariba User	R_BTP_ARB_ADMIN			
IFlow	Ariba_Event_Management_API_Connector	See <a href="#">ERP-108</a>				
IFlow	JMS_to_JMS_GetKeelvarAwardData	Ariba Event API Process Direct Path	/ariba/events/connector			Ariba Event Management API Connector IFlow
		Ariba JMS Event Queue name	aribaEventMgtQueue			JMS Queue Name
		Keelvar Connector Process Direct Path	/keelvar/connector			Keelvar API Connector IFlow
		JMS Process Direct Path	/ariba/generate-bid-sheet			Process Direct path for the next IFlow to be executed
		Max Delay Thread	5			Delay for Event status to change
IFlow	JMS_to_JMS_GenerateBidSheets	Ariba Surrogate Bid API Connector Process Direct	/ariba/surrogate-bidding/connector			Ariba Surrogate Bid API Connector defined in IFlow Ariba_Surrogate_Bidding_API_Connector
		JMS Queue Name	aribaEventMgtQueue			Name of the JMS Queue
		Delay in Seconds for Status Check	5			Pause between each Status Check iteration
		JMS Process Direct Path	/ariba/create-award			Process Direct path for the next IFlow to be executed
		Status Check Iterations	10			The count of attempts to receive Job Completion status.
IFlow	JMS_to_Ariba_CreateAwardScenarioToEvent	Ariba Event Management API Connector Process Direct	/ariba/events/connector			Ariba Event Management Connector IFlow. See NRF=2007 for details
		Maximum Delay for Status Check	2			Delay for Event status to change

## Appendix

### See also

File	Modified
File IFlow Steps draw.io diagram	Apr 23, 2026 by EPASINGHE-ext, Kapila
File -IFlow Steps.tmp draw.io Draft	Apr 23, 2026 by EPASINGHE-ext, Kapila

File Create Keelvar Bid Sheet draw.io diagram	Feb 24, 2026 by EPASINGHE-ext, Kapila
File ~Create Keelvar Bid Sheet.tmp draw.io Draft	Feb 24, 2026 by EPASINGHE-ext, Kapila
File Ariba Keelvar Award Interface draw.io diagram	Feb 24, 2026 by EPASINGHE-ext, Kapila
File ~Ariba Keelvar Award Interface.tmp draw.io Draft	Feb 24, 2026 by EPASINGHE-ext, Kapila
File Create Award Scenario draw.io diagram	Dec 12, 2025 by EPASINGHE-ext, Kapila
File ~Create Award Scenario.tmp draw.io Draft	Dec 12, 2025 by EPASINGHE-ext, Kapila
File drawio-backup-IFlow Steps-rev-3 draw.io diagram backup	Dec 10, 2025 by EPASINGHE-ext, Kapila

[Download All](#)



## Change log

Version	Published	Changed By	Comment
<b>CURRENT (v. 44)</b>	<b>Apr 23, 2026 06:59</b>	<b>EPASINGHE-ext, Kapila</b>	
v. 43	Mar 18, 2026 14:22	WENNINGER-ext, Sascha	
v. 42	Feb 24, 2026 13:39	EPASINGHE-ext, Kapila	CR273 and cleaning up
v. 41	Feb 24, 2026 13:13	EPASINGHE-ext, Kapila	
v. 40	Feb 24, 2026 12:52	EPASINGHE-ext, Kapila	
v. 39	Feb 19, 2026 12:06	MUKHOPADHYAY-ext, Aakshita	
v. 38	Feb 19, 2026 08:16	MUKHOPADHYAY-ext, Aakshita	
v. 37	Feb 19, 2026 08:08	MUKHOPADHYAY-ext, Aakshita	
v. 36	Feb 18, 2026 10:28	MUKHOPADHYAY-ext, Aakshita	
v. 35	Feb 12, 2026 12:10	MUKHOPADHYAY-ext, Aakshita	

[Go to Page History](#)

## Workflow history

This view shows the 5 most recent entries. The complete workflow log is available from the 'Document Activity' menu item.

From	Actor	Type	Activity	Version
Mar 18, 2026 to Apr 23, 2026				
Approved	EPASINGHE-ext, Kapila and WENNINGER-ext, Sascha	Edit	multiple updates from  EPASINGHE-ext, Kapila and WENNINGER-ext, Sascha	
	WENNINGER-ext, Sascha	State	changed state to <b>Approved</b> at 1:23 pm	v43
Revision under Review	WENNINGER-ext, Sascha	State	gave <i>Minor change</i> approval at 1:23 pm	
		State	changed state to <b>Revision under Review</b> at 1:23 pm	v43
Revision in progress	WENNINGER-ext, Sascha	State	changed state to <b>Revision in progress</b> at 1:22 pm	v43
Mar 06, 2026				
Approved	 TILBEE-ext, Amanda	State	changed state to <b>Approved</b> at 9:35 am	v42