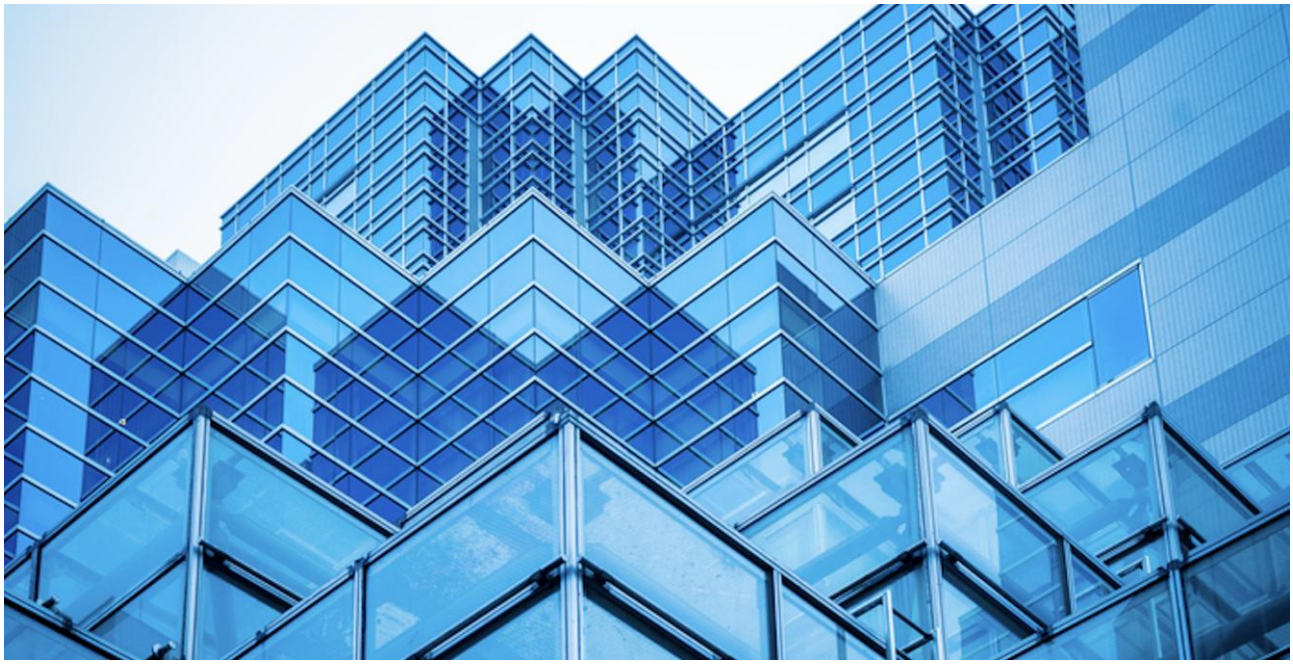


PUBLIC

SAP Build Process Automation

Direct Supplier Down Payment Request
Configuration and User Guide

Contents

Change Log

Version	Date	Description
1.0.0	June 26, 2025	Document update: Appendix updated

This document contains 3 sections. Each section covers different information about this template content package.

1. **Overview**: In this section, you will get quick overview of the use case, what are high-level components used and how the template works in nutshell. Prerequisite section provides information of different services of SAP Business Technology Platform that are required to use this template.

2. **Configuration Guide** It contain sections which will guide you to setup your (a) SAP Business Technology Platform Account with destination, cloud connector, user roles etc., (b) Integration Content via SAP Integration Suite, (c) SAPUI5 application via SAP Business Application Studio and (d) Import and configure the template using SAP Build Process Automation design studio and (e) SAP Work Zone to access the My Inbox or Task Center and other applications.

3. **User Guide**: This section provides details about different artefacts that are used in this template like process definition, decision diagram, action project details, forms, email notifications etc. to better understand how different capabilities are used in this template.

Note:

- Learning contents are published only for demo and reference purposes. We do not provide any support to learning / sample content.
- This documentation is not a detailed guide to setup SAP Business Technology Platform services. It assumes that IT admin who is setting the content is skilled with SAP Business Technology Platform environment. **Configuration section must be followed.**
- This template content is to accelerate your solution development. You must modify this template according to your requirements to achieve the desired business goal. To use this template content, you need to have basic knowledge and understanding of SAP Build Process Automation, SAP Integration Suite, and its capabilities.

Overview

This document provides technical information about what needs to be configured to make the package – Direct Supplier Down Payment Request to run. The main audience of this document are technical personas like IT Administrators or Developers.

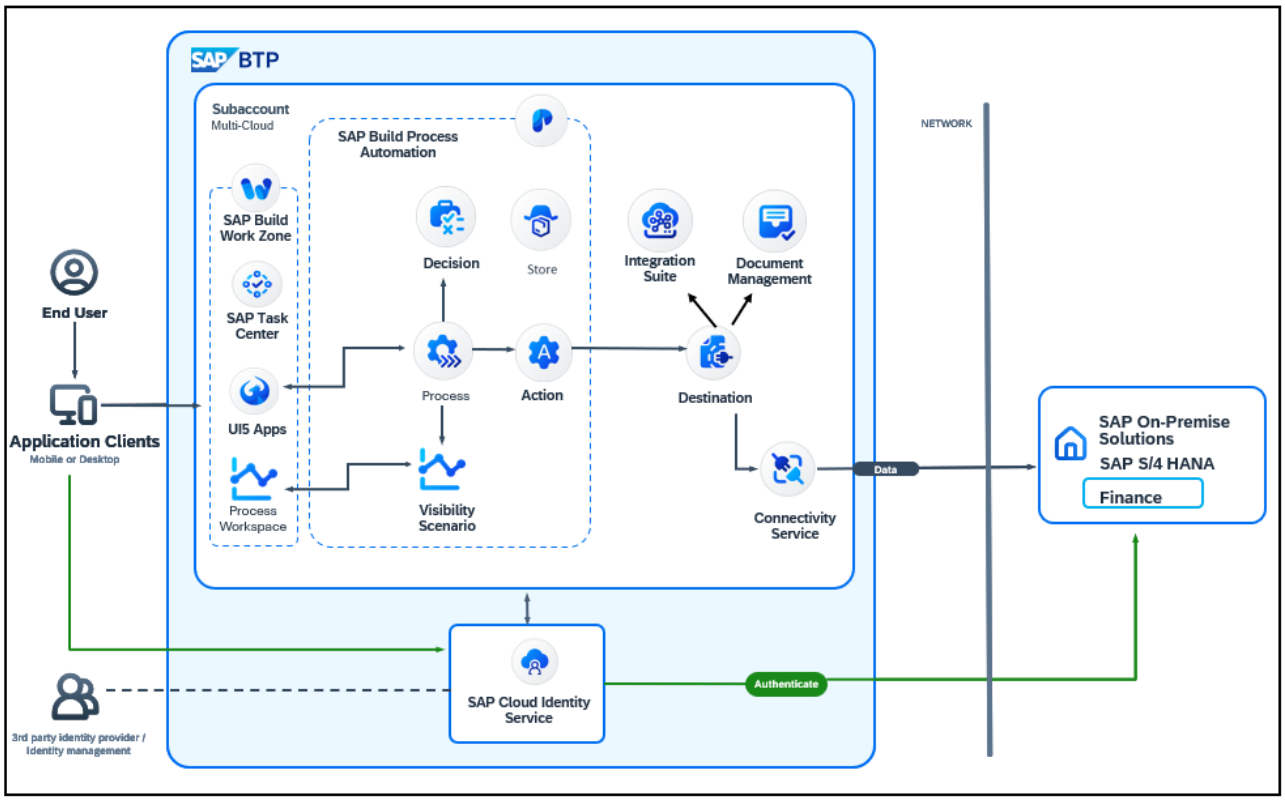
Direct Supplier Down Payment Requests content package for SAP S/4HANA automates Down Payment Approvals in a flexible and transparent approach. The down payment is created based on the approvals and validations performed in the process.

This content package can be used with both SAP S/4HANA Cloud Private Edition (2023 onwards) and S/4HANA On-Premise.

Features:

- Custom validation of down payment attributes through decisions.
- Automatic email notification to parties involved.
- Out-of-the-box visibility into key process performance indicator

Process Flow Diagram



Solution flow :

1. The requester initiates a Direct Supplier Down Payment Request by submitting details, including supplier, amount, and attachments, through the SAP UI5 application.
2. The system validates the down payment request using decisions.
3. Upon successful validation, the request is routed to the first approver, and an email notification is sent with a task assigned in SAP My Inbox.
4. Approvers evaluate the request and take one of the following actions:
    a. **Approve**: Advances the request to the next approver or triggers final creation in SAP S/4HANA.
    b. **Reject**: Ends the process, and the request status is updated to "Rejected."
    c. **Rework**: Sends the request back to the requester for updates, including approver comments.
5. If reworked, the requester makes necessary changes and resubmits the request to restart the approval process.
6. Once all approval levels are completed, the system creates the down payment request in SAP S/4HANA, and the status is updated to "Approved."
7. The requester and relevant stakeholders receive notifications about the final status and down payment request details.

Note:

- The process can be triggered via a Start UI.

Required SAP Business Technology Platform Services

The following SAP Business Technology Platform services are required to consume the **Direct Supplier Down Payment Request**:

- SAP Build Process Automation
- SAP Connectivity service

- SAP Business Application Studio
- SAP Application Runtime Service
- SAP Document Management Service
- SAP Integration Suite
- SAP Work Zone, standard or advanced
- SAP Cloud Identity Services - Identity Authentication (optional)
- SAP Business Technology Platform, Cloud Foundry runtime.

Prerequisites

For this template to run successfully the following prerequisites are needed

- Get access to BTP account.
- Need valid SAP S/4HANA user.
- This template requires Integration package SAP Build Process Automation Integration with SAP S/4 HANA for Direct Supplier Down Payment Request to run. You can download this package separately from the store. The guide to set-up and configure integration package is mentioned along with the integration package itself.

Configuration Guide

Direct Supplier Down Payment Request template requires SAP Build Process Automation subscription and Destination Setup for SAP Build Process Automation. Follow the setup and configuration section

Setup Business Technology Platform Cockpit

Configure SAP Build Process Automation Destination

To Access the SAP Build process Automation, you can configure the destination. Please create destination accordingly

Destination Property	Value
Name	sap_process_automation_service
Type	HTTP
URL	<" endpoints"."api">
Client ID	<"uaa"."clientid">
Client Secret	<"uaa"."clientsecret">
Token Service URL	<"uaa"."url">/oauth/token
Proxy Type Internet	Internet
Authentication	OAuth2ClientCredentials

Add the property **sap.processautomation.enabled** as **true** to view the destination in the SAP Build Process Automation tenant. Please refer help document for more details.

Configure SAP S/4HANA Destination

This content package can be used with SAP S/4HANA On-Premise. Destination can be created accordingly.

Destination Property	Value

Name	<Name>
Type	HTTP
URL	<SAP S/4HANA endpoint> Example: https://<s4system-name>-api.xxxxxxxx
Proxy Type Internet	Internet
Authentication	Basic Authentication
Username	<Username> of the SAP S/4HANA Communication User
Password	<Password> of the SAP S/4HANA Communication User

Configure SAP Integration Suite Destination

Configure Destination to connect to Integration Suite destination cloud system.

Destination Property	Value
Name	IntegrationSuite
Type	HTTP
URL	<Integration Suite Destination endpoint> https://<CPI>.hana.ondemand.com
Proxy Type Internet	Internet
Authentication	Basic Authentication
Username	<Username> of the SAP S/4HANA Communication User
Password	<Password> of the SAP S/4HANA Communication User

Configure SAP Build Process Automation Email Destination

Mail notifications are used in this template to send notifications to requestor, approver, and admin during different stages of the process. Follow the instructions to setup mail destination in your SAP Business Technology Platform Account.

Configuring destination, Configure Mail Destination

- For more information refer to Create HTTP Destinations and OAuth User Token Exchange Authentication.

User Roles Configuration

To access this template, the business user will need these roles:

Role Collection Name	Purpose
TaskCenterAdmin	Permission to execute calls to the connector status API and monitor configured destinations and running background jobs.

Process Automation Developer	Standard Business User, can model and publish processes
ProcessAutomationParticipant	To execute (like accessing MyInbox, running the automations, etc.) this content package, you need to have this role assigned to the respective user.
ProcessAutomationAdmin	To perform process admin activities like accessing monitoring applications etc., you need to have this role assigned to the respective user.
Business_Application_Studio_Administrator	Allows administrators to manage (export and delete) user data.
Business_Application_Studio_Developer	Allows developers to load and develop applications using SAP Business Application Studio.

For more information about SAP Build Process Automation roles, see Authorizations.

Cloud Connector Configuration

For SAP S/4HANA on-premise landscape, configure cloud connector to enable secure tunnel to SAP Business Technology Platform tenant. Please refer the help documentation to configure Cloud Connector.

Services/Resources that need to be exposed from SAP S/4HANA on-premise using Cloud Connector.

Resources	Protocol	Backend-Type
/sap/opu/odata/sap/ZAPI_JE_CREATE_SRV_01/JournalEntrySet	OData	SAP S/4HANA
/sap/opu/odata/sap/ZZ1_JOURNALDETAILS_CDS	OData	SAP S/4HANA
/sap/opu/odata/sap/ZZ1_JOURNALITEMDETAILS_CDS	OData	SAP S/4HANA
/sap/opu/odata/sap/ZZ1_COMPANYCODE_CDS	OData	SAP S/4HANA
/sap/opu/odata/sap/ZZ1_PLANTS_CDS	OData	SAP S/4HANA
/sap/opu/odata/sap/ZZ1_SUPPLIER_CDS	OData	SAP S/4HANA
/sap/opu/odata/sap/ZZ1_CURRENCY2_CDS	OData	SAP S/4HANA
/sap/opu/odata/sap/ZZ1_SPECIALGLCODE_CDS	OData	SAP S/4HANA
/sap/opu/odata/sap/ZZ1_PURCHASINGDOCUMENTITEM_CDS	OData	SAP S/4HANA
/sap/opu/odata/sap/ZZ1_TAXCODES_CDS	OData	SAP S/4HANA
/sap/opu/odata/sap/ZZ1_PAYMENTBLOCK_CDS	OData	SAP S/4HANA
/sap/opu/odata/sap/ZZ1_PAYMENTMETHOD_CDS	OData	SAP S/4HANA

Setup Content Package

Import SAP Build Process Automation content

Standard automation packages released by SAP will be available at the store. To know more about the store visit SAP Build Process Automation Store .

This template can be downloaded from the Store:

1. Navigate to the Store in your SAP Build Process Automation account.
2. Select the Project Type filter as “Business Process.”
3. In the Search bar type “**Direct Supplier Down Payment Request**”
4. Select the “Create from Template” button to add the template into your lobby.
5. After successfully adding the template, navigate back to the lobby to find the “**Direct Supplier Down Payment Request”** template available for use.

Caution: Be sure to select the **Project Type** as "**Business Process**" in the filters.

Import and deploy TASK UI application

The content package requires the SAP UI5 user interface which is available in Direct Supplier Down Payment Request Process content package. Below are details of Task UI application

Task UI Application: SuppdownpayUI-DirectSupplierDownPaymentRequest
ReworkUI-DirectSupplierDownPaymentRequest

Following steps can be followed to utilize the SAP UI5 application provided with this process package:

- Download the SAPUI5 application Task UI from the imported store project.
- Open Business Application Studio and import the previously downloaded SAPUI5 project.
- Build SAPUI5 project using mta.yaml file and the generated project (generated mtar file will be available in the mta_archieves folder) file can be deployed.

Configuration & Modification

Once you have added the content to your lobby, you will find the project in your project list. Adapt the content template based on your requirements.

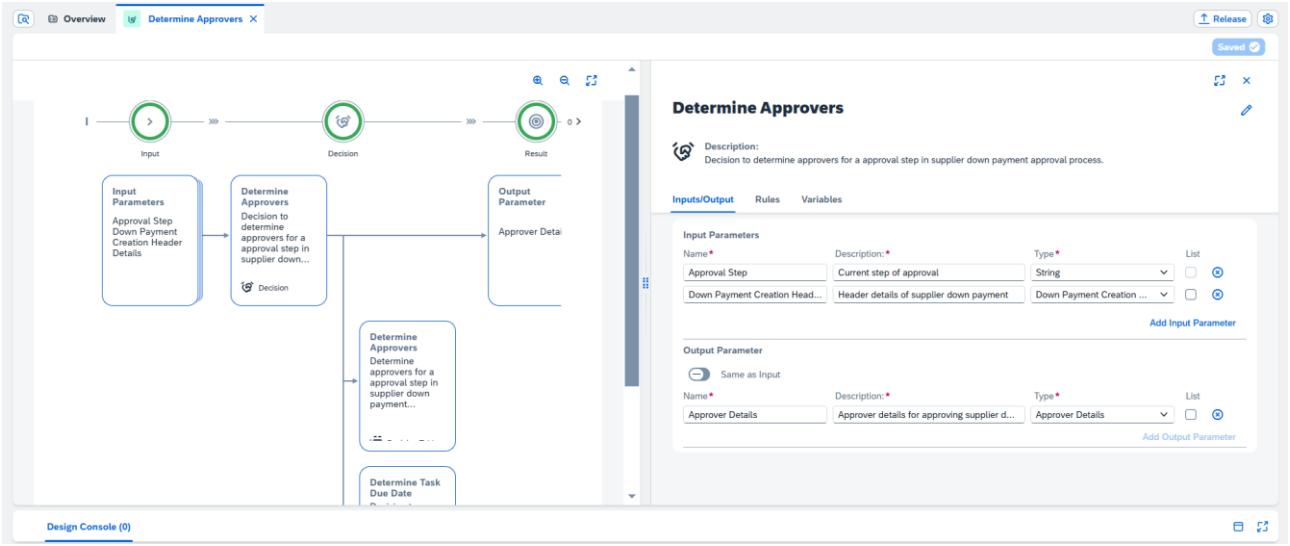
Configure Decisions

There are 3 decisions in this template. We can configure as required.

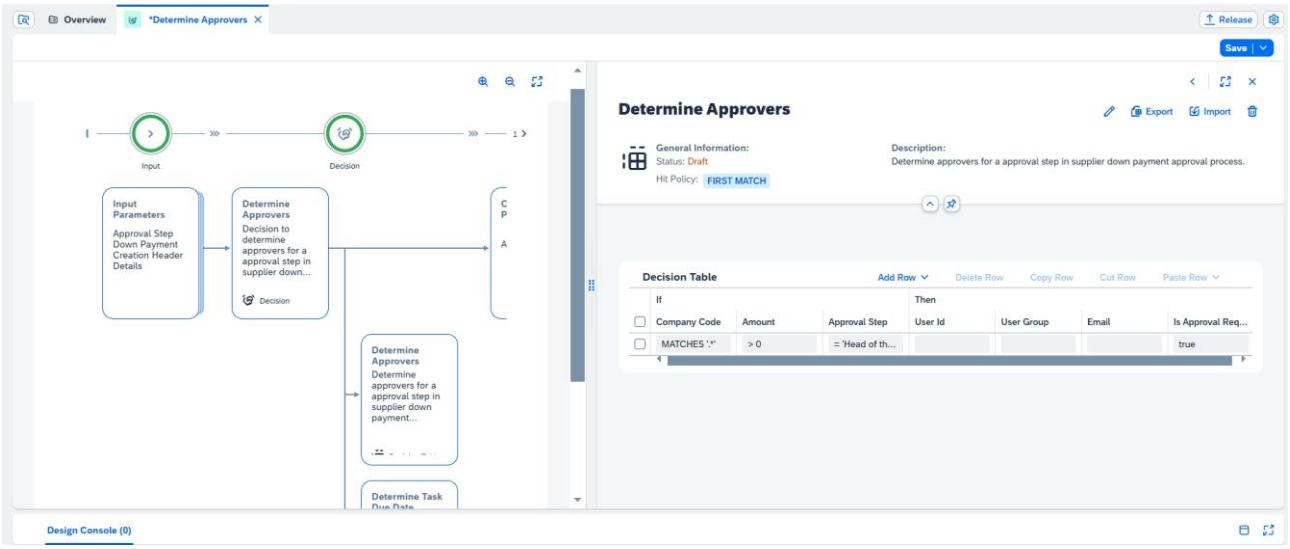
1. Determine Approvers

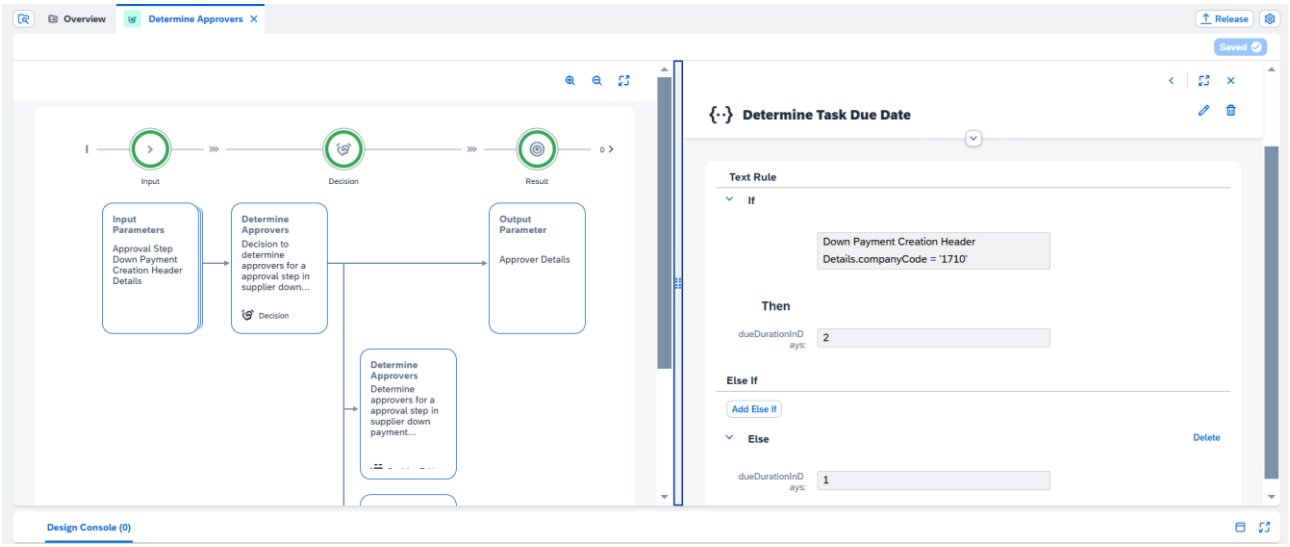
Decision to determine approvers for a approval step in supplier down payment approval process. To modify the decision:

- In process editor, double click to open the decision.
- Select the rule from the decision diagram.



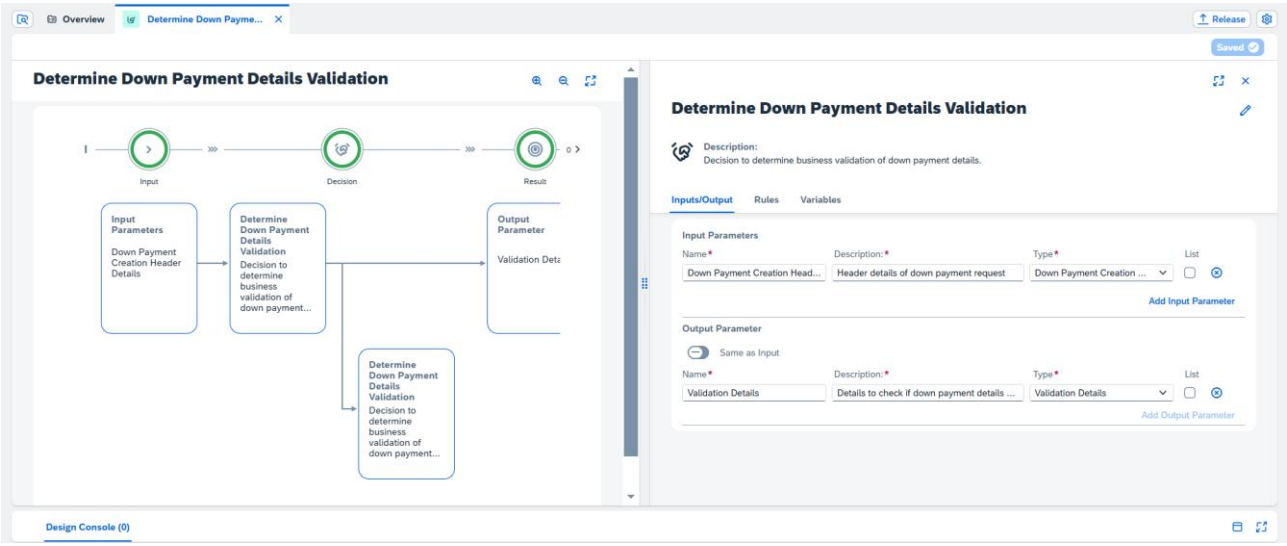
- Edit the rule to add more rows or update existing value.





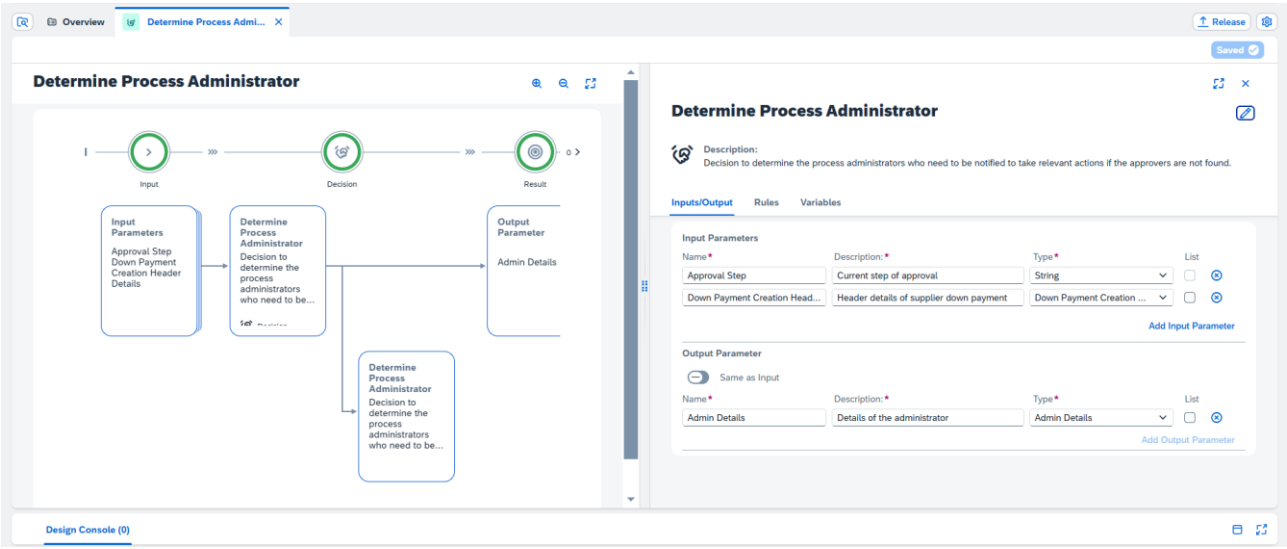
2. Determine Down Payment Details Validation

Decision to determine business validation of down payment details.



3. Determine Process Administrator

Decision to determine the process administrators who need to be notified to take relevant actions if the approvers are not found.



Configure Email

Email notifications are sent to requestor and approver at different places during process and task execution. All the email artefacts in the process have default template body which can be modified.

To update the email body content, do the following:

- Select the mail artefact in the process.
- From the **Settings** in the right-side panel, check the **Mail Header** to confirm the **To** field (or email recipients) and the **Subject**. You can adjust the subject and recipient, if needed.
- You may also add someone in **CC**.

- Click to Open '**Mail Body Editor**'.

- The mail editor allows you to write and adjust text according to your needs. You can also reuse information that has been collected during the process. If you want to use additional process context data, you can simply drag the fields from the left menu to the editor on the right.

For more information on email configuration and text, refer to this link

In case any further extension needs to be made then follow the reference links below.

Create Process
Create a Business Process | SAP Help Portal

Create Forms
Create a Form | SAP Help Portal

Create Automation
Create an Automation | SAP Help Portal

Create Decision
Create a Decision | SAP Help Portal

Create Process Visibility Scenario
Create a Visibility Scenario | SAP Help Portal

Release and Deploy the Process

After configuring your process, it's time to publish, test and run it. To run a process, you must first release and deploy it.

Release - This locks the version of your process, meaning that no further edits can be made to that version.
At this stage, the process can't be run and isn't available to participants yet.
To release a version of your process:

- First ensure that the status is listed as 'Editable'.
- Then click 'Release'.

For more information please visit: SAP Build Process Automation

Note: If this is the first time you've released the process, a version labelled 1.0.0 is released. For subsequent
releases, you're prompted to select a version type before the process is released.

Import Destination
Actual destinations that are created in SAP BTP cockpit will be required when you deploy the process. This destination will be mapped to the destination environment variables during deployment.

To configure destination,

- Click on **Control Tower.**
- Select **Destinations** from the right panel.
- Click **Add**

Select the destinations created in the setup section to be added.
For more information and screen shots, please refer to the below link: Manage Destinations

Deploy – This takes a released version of a process and enables it to be actively run and monitored. A deployed version of this approval process unlocks the request form, allowing participants to submit requests and the approvers to view items in their inbox.

To deploy a released version of your process,

- First ensure that the status is listed as 'Released'.
- Then click '**Deploy'**.

For more information and screen shots, please refer to the below link: Deploy Project

To deploy your project, you will be prompted to provide a destination as an environment variable. Select the destination from the drop-down and fill other environment variables as well.

- **SAP S/4HANA**: Business Technology Platform destination to connect with SAP S/4HANA system. As shown in SAP S/4HANA destination set-up-section.

Note: If you do not map your destination environment variable with actual SAP Business Technology Platform destination during deployment, then the process will enter erroneous state when it tries to connect to the external system.

Configure and Deploy Start UI

The content package requires the SAP UI5 user interface which is available in **Direct Supplier Down Payment Request** content package. Below are details of Start UI application

Start UI Application: CreateUI-DirectSupplierDownPaymentRequest

Following steps can be followed to utilize the SAP UI5 application provided with this process package:

- Download the SAPUI5 application Start UI from the imported store project.
- Open Business Application Studio and import the previously downloaded SAPUI5 project.
- The definition id can be found in the Monitoring section, inside Process and Workflow Definitions. You need to enter the project name and go to the main process. The ID mentioned in the process would be the definition id. Refer determine the workflow definition ID for more details
- This definition id must be updated in Start UI application. It will be maintained either in the i18n file or controller.js file. Please refer below screenshot

```
startWorkflowInstance: function() {
  var model = this.getView().getModel();
  var definitionId = [redacted];
  var initialContext = model.getProperty("/initialContext");

  var data = {
    definitionId: definitionId,
    context: JSON.parse(initialContext)
  };

  $.ajax({
    url: this._getWorkflowRuntimeBaseURL() + "/workflow-instances",
    method: "POST",
    async: false,
```

- Ensure that the instance details in mta.yaml file is as below.

Configure SAP Work Zone

Configure SAP Work Zone to access the start UI for this automation.
Once SAP Work zone is configured, perform the following steps to create the tile for the start UI:

1. Open content manager and click on the HTML5 Apps.
2. Select the Import Direct Supplier Down Payment Request Start UI from the list of apps and choose Add.
3. Add the relevant roles to the user. Direct Supplier Down Payment Request Start UI will be available to the user as a tile in the SAP Work Zone.

If you plan to use SAP Work Zone service, then configure SAP Work Zone with SAP Build Process Automation Applications using help documentation.

User Guides

Capabilities

The following are the capabilities of this template:
For detailed information about different artifacts, please refer help documentation

Type	Name	Description
Decision	Determine Approvers	Decision to determine approvers for a approval step in supplier down payment approval process.

Decision	Determine Down Payment Details Validation	Decision to determine business validation of down payment details.
Decision	Determine Process Administrator	Decision to determine the process administrators who need to be notified to take relevant actions if the approvers are not found.
Process	Direct Supplier Down Payment Approval Process	Process to get approval and handle rework for supplier down payment request.
Process	Direct Supplier Down Payment Creation Process	Process to create down payment request in SAP S/4HANA system post approval.
Process	Direct Supplier Down Payment Validation Process	Process to validate supplier down payment data and notify initiator if validation failed.
Form	Admin Form	Form to assign approvers manually in case not determined from the decisions.
Form	Approval Form	Form to approve or reject supplier down payment request.
Form	Rework Form	Form to rework on down payment details in case suggested by the approver.
Data type	Admin Details	Data type for details of process administrators who will be notified to take action if there is any issue with determining the approvers.
Data type	Approver Details	Data type for approver details to approve supplier down payment request.
Data type	Down Payment Creation Details	Data type for header and item details of supplier down payment from start payload.
Data type	Down Payment Creation Header Details	Data type for header details of supplier down payment.
Data type	Down Payment Creation Item Details	Data type for item details of supplier down payment.
Data type	History Details	Data type for history details like processor, decision and comments.
Data type	Response Message Details	Data type for message array details like message type, description.
Data type	Validation Details	Data type for checking if supplier down payment details are valid.

Actions

Direct Supplier Down Payment Request Creation Action

This service is based on the OData protocol and can be used to create supplier down payment requests.

Method	API	API Path	Purpose
POST	/sap/opu/odata/sap/ZAPI_JE_CREATE_SRV_01	/sap/opu/odata/sap/ZAPI_JE_CREATE_SRV_01/JournalEntrySet	Action to create supplier down payment request

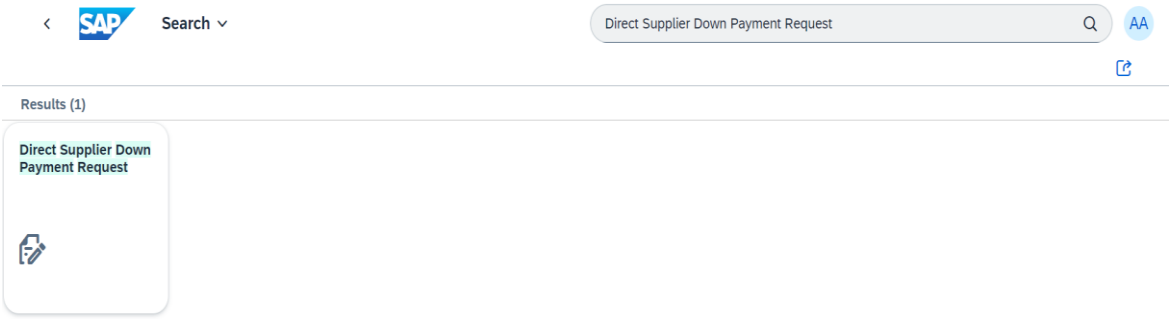
SAP UI5 SCREEN

This is the Start UI application to submit the request for Direct Supplier Down Payment Request must be entered here to trigger the process.

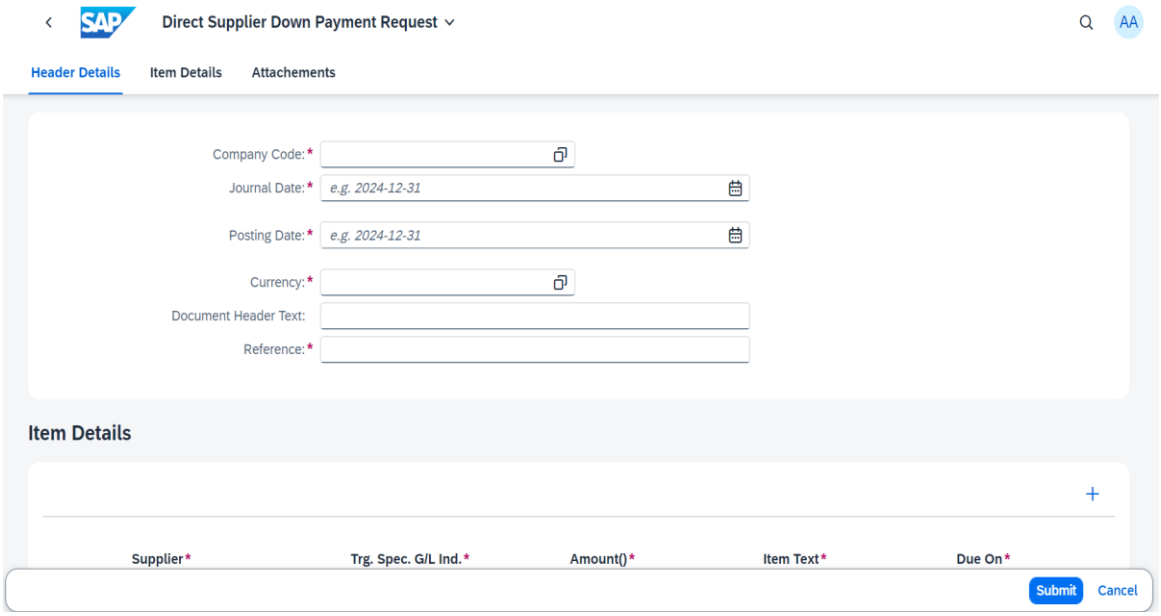
The tile name that must be chosen in the Work Zone is: '**Direct Supplier Down Payment Request**'.

Once navigated to the tile, User can start the process.

1. Access app Direct Supplier Down Payment Request.



2. The user can initiate the creation of a new Supplier Down Payment Request by clicking the "Create" button.



In 'Header Details', we have 6 fields:

Company Code (Mandatory): Identifier for the specific company within a corporate group.

Journal Date (Mandatory): Date when the journal entry is recorded.

Posting Date (Mandatory): Date the transaction is posted in the accounting system.

Currency (Mandatory): Type of currency used for the transaction.

Document Header Text: Brief description of the document or entry.

Reference (Mandatory): External identifier or reference number for the document.

In 'Item Details', we have 10

Supplier (Mandatory): Vendor or entity from whom goods or services are procured.

G/L Indicator (Mandatory): Flag specifying if the entry pertains to a General Ledger account.

Amount (Mandatory): Monetary value of the transaction or entry.

Item Text (Mandatory): Description or details related to the specific line item.

Due On (Mandatory): Date the payment or obligation is due.

Purchasing Document: Identifier for the procurement-related document.

Purchasing Document Item: Specific line item within the purchasing document.

Tax Code: Code defining the applicable tax category or rate.

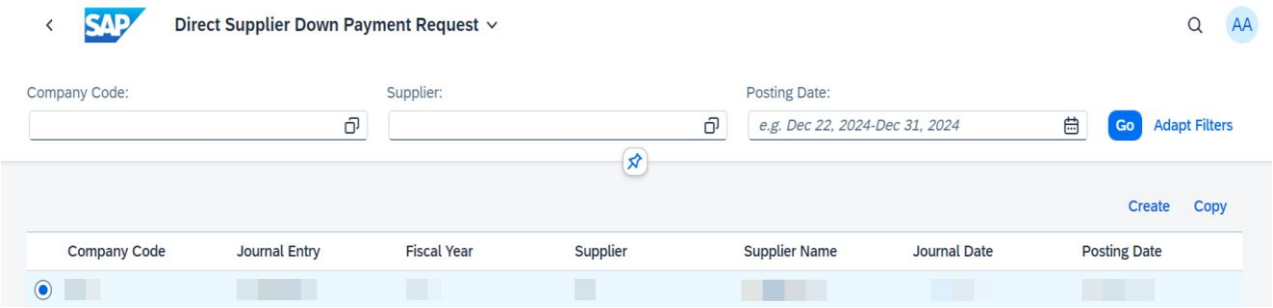
Payment Block: Indicator to prevent or delay payment processing.

Payment Method: Means or channel through which the payment will be made.

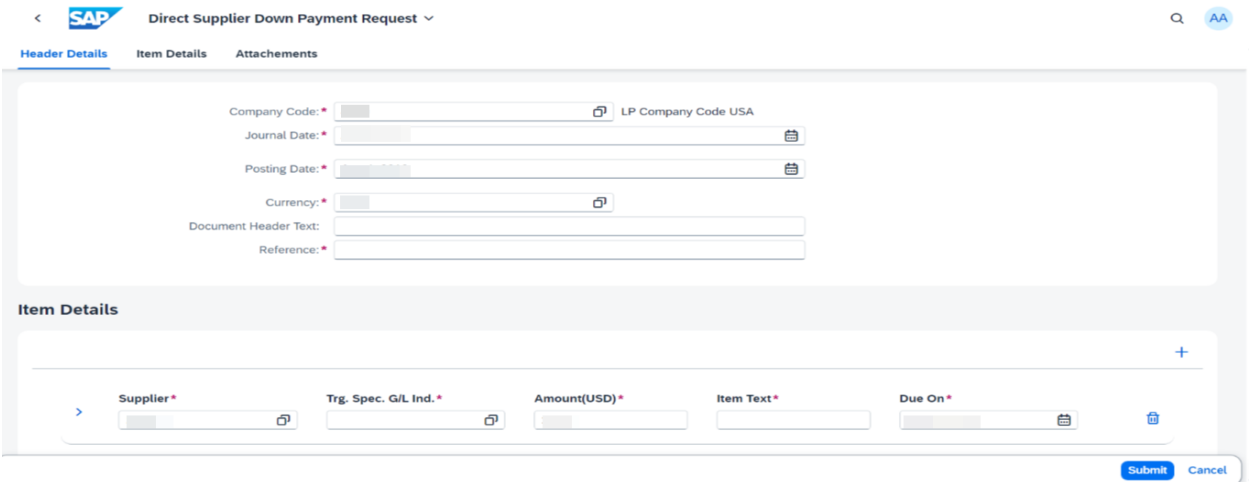
In 'Attachements' tab, we have 1 field:

Attachments (Mandatory): Additional files or documents associated with the entry.

3. The user will also be able to create a new Supplier Down Payment Request using the 'Copy' button. When the radio button of a supplier down payment that needs to be copied is selected the 'Copy' button will be available to initiate a request.



In the 'Header Details' and 'Item Details' tab, there are 5 mandatory fields that will be automatically filled after the 'Copy' operation:



Once all the mandatory/required details are filled, click on the Submit button in SAP UI5 screen to start the process

Appendix

Custom CDS View and Custom communication scenario

To call the APIs related to Direct Supplier Down Payment Request, the following CDS Views and custom communication scenarios along with communication arrangement should be enabled for the communication user. The SAP S/4HANA custom communication scenario consists of the external API details which was used to display the Supplier Down payment details and value help used in Direct Supplier Down Payment Request application. This application is developed based on CDS as external ODATA API and creation of custom communication scenario.

The creation of Custom CDS View as external API's and custom communication scenario provide below with screen shots below.

Creation of CDS View Journal details (ZZ1_JournalDetails)

Package Name: SAP S/4HANA Cloud for Finance

Label: JournalDetails

Name: ZZ1_JournalDetails

1. Login to SAP S/4HANA cloud Fiori Launchpad with required credentials.
2. Go to Extensibility business group and click on Custom CDS Views tile.
3. Click on create button it will open the pop-up, provide label as JournalDetails and choose External API option from scenario dropdown and click on button create in pop-up window.
4. Select primary data source, a popup will open with list of CDS Views. Provide the Standard CDS view name I_JournalEntry in search field and press enter. Select the required CDS to continue.
5. In case to use multiple CDS in Data sources tab, click on Add button and select associated data source which opens pop-up. Provide the Standard CDS view name ZZ1_JournalItemDetail in search field and Click "OK".
6. After selection of primary and secondary data sources, click on Join condition icon which open Define Join Conditions pop-up, click Add button and make relationship between two data sources by selection of similar fields in Field of Associated Data Source popup and change cardinality as per requirement of CDS Views.

Define Join Conditions: Perform the join condition between these data sources as listed below:

Name: I_JournalEntry

Alias: _ZZ1_JournalItemDetail

Cardinality: Zero or More[0..*]

Data Source Field	Operator	Value Type	Value
CompanyCode	Equal	Field	I_JournalEntry.CompanyCode
FiscalYear	Equal	Field	I_JournalEntry.FiscalYear
AccountingDocument	Equal	Field	I_JournalEntry.AccountingDocument

7. Go to Elements tab, click add button and select Elements from submenu and select required fields from multiple CDS click Ok button.
8. Click on Publish button present on Footer.

Creation of CDS Journal Item Detail (ZZ1_JournalItemDetail)

Package Name: SAP S/4HANA Cloud for Finance

Label: JournalItemDetail

Name: ZZ1_JournalItemDetail

Repeat step 1 to 3 as mentioned above in Journal details CDS

For 4th step select primary data source as I_JournalEntryItem shown in Step 4 above.

For step 5, select I_Supplier_VH as Associated data source.

For step 6, use below join condition.

Cardinality: Zero or One[0..1]

Data Source Field	Operator	Value Type	Value
Supplier	Equal	Field	I_JournalEntryItem.Supplier

Repeat step 7 by selecting required fields in Elements tab as shown in below table

Key	Alias	Path	Label
ON	SourceLedger	I_JournalEntryItem.SourceLedger	Source Ledger
ON	CompanyCode	I_JournalEntryItem.CompanyCode	Company Code
ON	FiscalYear	I_JournalEntryItem.FiscalYear	Fiscal Year
ON	AccountingDocument	I_JournalEntryItem.AccountingDocument	Journal Entry
ON	LedgerGLLineItem	I_JournalEntryItem.LedgerGLLineItem	Journal Entry Item
ON	Ledger	I_JournalEntryItem.Ledger	Ledger
OFF	Supplier	I_JournalEntryItem.Supplier	Supplier
OFF	SupplierName	_I_Supplier_VH.SupplierName	Name
OFF	SpecialGLCode	I_JournalEntryItem.SpecialGLCode	Special G/L
OFF	TaxCode	I_JournalEntryItem.TaxCode	Tax Code
OFF	FinancialAccountType	I_JournalEntryItem.FinancialAccountType	Account Type
OFF	CreationDateTime	I_JournalEntryItem.CreationDateTime	Creation Time
OFF	_SourceLedger	I_JournalEntryItem._SourceLedger	_SourceLedger
OFF	_SourceLedgerText	I_JournalEntryItem._SourceLedgerText	_SourceLedgerText
OFF	_CompanyCode	I_JournalEntryItem._CompanyCode	_CompanyCode
OFF	_FiscalYear	I_JournalEntryItem._FiscalYear	_FiscalYear

OFF	_JournalEntry	I_JournalEntryItem._JournalEntry	_JournalEntry
OFF	_Ledger	I_JournalEntryItem._Ledger	_Ledger
OFF	_LedgerText	I_JournalEntryItem._LedgerText	_LedgerText
OFF	_Supplier	I_JournalEntryItem._Supplier	_Supplier
OFF	_SupplierText	I_JournalEntryItem._SupplierText	_SupplierText
OFF	_SpecialGLCode	I_JournalEntryItem._SpecialGLCode	_SpecialGLCode
OFF	_FinancialAccountType	I_JournalEntryItem._FinancialAccountType	_FinancialAccountType
OFF	_FinancialAccountTypeText	I_JournalEntryItem._FinancialAccountTypeText	_FinancialAccountTypeText

Repeat step 8 by publishing CDS View

Creation of CDS view Company Code (ZZ1_CompanyCode)

Package Name: SAP S/4HANA Cloud for Finance

Label: ComapnyCode

Name: ZZ1_CompanyCode

Repeat step 1 to 3 as mentioned above in Journal Detail CDS

For 4th step select primary data source as I_CompanyCode as shown in step 4 above

Steps 5 & 6 are not required as it is single data source

Repeat step 7 by selecting required fields in Element's tab as shown in below table

Key	Alias	Path	Label
ON	CompanyCode	I_CompanyCode.CompanyCode	Company Code
OFF	CompanyCodeName	I_CompanyCode.CompanyCodeName	Company Name
OFF	Currency	I_CompanyCode.Currency	Currency
OFF	Language	I_CompanyCode.Language	Language Key

Repeat step 8 by publish company code CDS View

Creation of CDS View plant (ZZ1_Plants)

Package Name: SAP S/4HANA Cloud for Manufacturing

Label: Plant

Name: ZZ1_Plants

Repeat step 1 to 3 as mentioned above in Journal Details with Scenario as Extenal API

For 4th step select primary data source as I_PlantStdVH as shown in step 4 above

Steps 5 & 6 are not required as it is single data source

Repeat step 7 by selecting required fields in Element's tab as shown in below table

Key	Alias	Path	Label
ON	Plant	I_PlantStdVH.Plant	Plant
OFF	PlantName	I_PlantStdVH.PlantName	Plant Name

Repeat step 8 by publish CDS View

Creation of CDS View supplier (ZZ1_Supplier)

Package Name: SAP S/4HANA Cloud for Database and Data Management

Label: Supplier

Name: ZZ1_Supplier

Repeat step 1 to 3 as mentioned above in Journal Details with Scenario as Extenal API

For 4th step select primary data source as I_Supplier as shown in step 4 above

Steps 5 & 6 are not required as it is single data source

Repeat step 7 by selecting required fields in Element's tab as shown in table.

Key	Alias	Path	Label
ON	Supplier	I_Supplier.Supplier	Supplier
OFF	SupplierName	I_Supplier.SupplierName	Name of Supplier
OFF	SupplierAccountGroup	I_Supplier.SupplierAccountGroup	Account Group

Repeat step 8 by publish CDS View

Creation of CDS View Currency (ZZ1_Currency2)

Package Name: SAP S/4HANA Cloud for Finance

Label: Currency

Name: ZZ1_Currency2

Repeat step 1 to 3 as mentioned above in Journal details with Scenario as Extenal API

For 4th step select primary data source as I_CurrencyText as shown in step 4 above

Steps 5 & 6 are not required as it is single data source

Repeat step 7 by selecting required fields in Element's tab as shown in below table

Key	Alias	Path	Label
ON	Language	I_CurrencyText.Language	Language Key

ON	Currency	I_CurrencyText.Currency	Currency
OFF	CurrencyName	I_CurrencyText.CurrencyName	Description

Repeat step 8 by publishing the CDS View

Creation of CDS View SpecialGLCode (ZZ1_SpecialGLCode)

Package Name: SAP S/4HANA Cloud for Finance

Label: SpecialGLCode

Name: ZZ1_SpecialGLCode

Repeat step 1 to 3 as mentioned above in Journal details with Scenario as Extenal API

For 4th step select primary data source as I_SpecialGLCodeText as shown in step 4 above

Steps 5 & 6 are not required as it is single data source

Repeat step 7 by selecting required fields in Element's tab as shown in below table

Key	Alias	Path	Label
ON	FinancialAccountType	I_SpecialGLCodeText.FinancialAccountType	Account Type
ON	SpecialGLCode	I_SpecialGLCodeText.SpecialGLCode	Special G/L ind.
ON	Language	I_SpecialGLCodeText.Language	Language Key
OFF	SpecialGLCodeName	I_SpecialGLCodeText.SpecialGLCodeName	Name
OFF	SpecialGLCodeLongName	I_SpecialGLCodeText.SpecialGLCodeLongName	Description

Repeat step 8 by publishing the CDS View

Creation of CDS View PurchasingDocumentItem (ZZ1_PurchasingDocumentItem)

Package Name: SAP S/4HANA Cloud for Sourcing and Procurement

Label: PurchasingDocumentItem

Name: ZZ1_PurchasingDocumentItem

Repeat step 1 to 3 as mentioned above in Journal details with Scenario as Extenal API

For 4th step select primary data source as I_PurchasingDocumentItemStdVH as shown in step 4 above

Steps 5 & 6 are not required as it is single data source

Repeat step 7 by selecting required fields in Element's tab as shown in below table

Key	Alias	Path	Label
ON	PurchasingDocument	I_PurchasingDocumentItemStdVH.PurchasingDocument	Purchasing Document
ON	PurchasingDocumentItem	I_PurchasingDocumentItemStdVH.PurchasingDocumentItem	Item

Repeat step 8 by publishing the CDS View

Creation of CDS View taxcode(ZZ1_taxcode)

Package Name: SAP S/4HANA Cloud for Finance

Label: taxcode

Name: ZZ1_taxcode

Repeat step 1 to 3 as mentioned above in Journal details with Scenario as Extenal API

For 4th step select primary data source as I_TaxCodeText as shown in step 4 above

Steps 5 & 6 are not required as it is single data source

Repeat step 7 by selecting required fields in Element's tab as shown in below table

Key	Alias	Path	Label
ON	TaxCalculationProcedure	I_TaxCodeText.TaxCalculationProcedure	Procedure
ON	TaxCode	I_TaxCodeText.TaxCode	Tax Code
ON	Language	I_TaxCodeText.Language	Language Key
OFF	TaxCodeName	I_TaxCodeText.TaxCodeName	Tax Code Name

Repeat step 8 by publishing the CDS View

Creation of CDS View Paymentblock(ZZ1_Paymentblock)

Package Name: SAP S/4HANA Cloud for Finance

Label: Paymentblock

Name: ZZ1_Paymentblock

Repeat step 1 to 3 as mentioned above in Journal details with Scenario as Extenal API

For 4th step select primary data source as I_PaymentBlockingReasonText as shown in step 4 above

Steps 5 & 6 are not required as it is single data source

Repeat step 7 by selecting required fields in Element's tab as shown in below table

Key	Alias	Path	Label
ON	Language	I_PaymentBlockingReasonText.Language	Language Key
ON	PaymentBlockingReason	I_PaymentBlockingReasonText.PaymentBlockingReason	Item Payment Block
OFF	PaymentBlockingReasonName	I_PaymentBlockingReasonText.PaymentBlockingReasonName	Payment Block Reason

Repeat step 8 by publishing the CDS View

Creation of CDS View PaymentMethod(ZZ1_PaymentMethod)

Package Name: SAP S/4HANA Cloud for Finance

Label: PaymentMethod

Name: ZZ1_PaymentMethod

Repeat step 1 to 3 as mentioned above in Journal details with Scenario as Extenal API

For 4th step select primary data source as I_PaymentMethodText as shown in step 4 above

Steps 5 & 6 are not required as it is single data source

Repeat step 7 by selecting required fields in Element's tab as shown in below table

Key	Alias	Path	Label
ON	Country	I_PaymentMethodText.Country	Country/Region Key
ON	PaymentMethod	I_PaymentMethodText.PaymentMethod	Payment Method
ON	Language	I_PaymentMethodText.Language	Language Key
OFF	PaymentMethodDescription	I_PaymentMethodText.PaymentMethodDescription	Description

Repeat step 8 by publishing the CDS View

Creation of Custom communication scenario

1. Login to SAP S/4HANA cloud Fiori Launchpad with required credentials.
2. Go to Extensibility in toolbar and click Custom Communication Scenarios tile.
3. Click on New button which open pop-up window provide communication scenario ID as Journal and description as Journal Details and click New on pop-up button.
4. Click on Add button in Inbound services and select required CDS from Add services pop-up and click on OK button.

   Select CDS Views as mentioned below:

   ZZ1_COMPANYCODE_CDS

   ZZ1_CURRENCY2_CDS

   ZZ1_JOURNALDETAILS_CDS

   ZZ1_PAYMENTBLOCK_CDS

   ZZ1_PAYMENTMETHOD_CDS

   ZZ1_PURCHASINGDOCUMENTITEM_CDS

   ZZ1_SPECIALGLCODE_CDS

   ZZ1_SUPPLIER_CDS

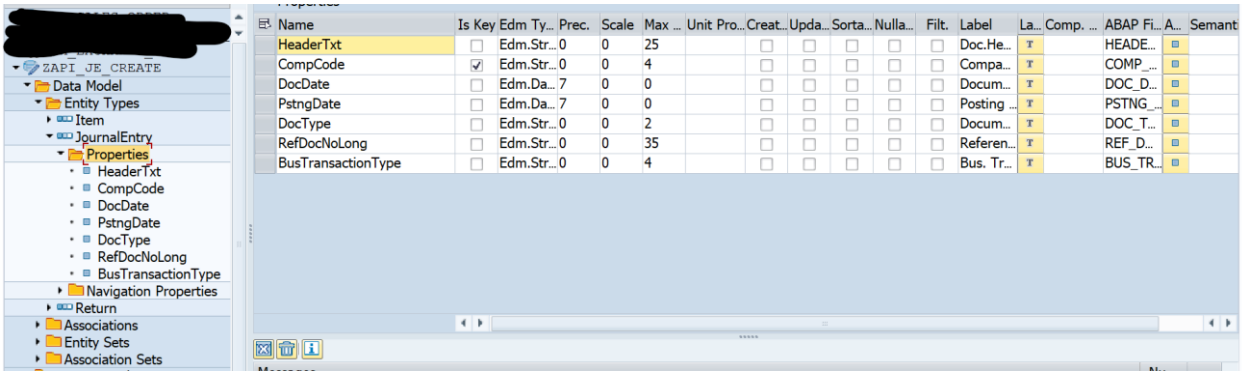
   ZZ1_TAXCODE_CDS

   Once you click on Add, below popup appears where you can select the above mentioned cds
5. Click on Publish button present on Footer, it will enable the Create Arrangement button after successful publish.
6. Click on Create Arrangement button, it will navigate to new Fiori app Communication arrangement by opening a pop-up of new communication arrangement. Click on Create button.
7. Enter Communication system and Username created in SAP S/4HANA system.
8. Save the Communication Arrangement and status should change to Active.

Setup Service: /sap/opu/odata/sap/ZAPI_JE_CREATE_SRV_01 /JournalEntrySet

Note: Action project used with this package have path configured as **"/sap/opu/odata/sap/ZAPI_JE_CREATE_SRV_01"** so make sure that below service is created as **ZAPI_JE_CREATE_SRV_01** or else you might have to update the action project in the package accordingly.

1. Go to the **SAP GUI** and enter **SEGW** in the command field to open the **Gateway Service Builder**.
2. In the **Service Builder** screen, click on the **"Create Project"** button.
3. Enter a name 'ZAPI_JE_CREATE' for your project in the **Project Name** field. This name should be unique and descriptive.
4. Provide a relevant short description of your project.
5. In the left-hand tree structure, right-click on **Data Model** and choose **Create** -> **Entity Type**.
6. Give the entity type a name as 'JournalEntry'
7. After creating the entity type, you can add **properties** (fields/attributes) that correspond to the attributes of your data model.



Name	Is Key	Edm Ty...	Prec.	Scale	Max ...	Unit Pro...	Creat...	Upda...	Sorta...	Nulla...	Filt.	Label	La...	Comp. ...	ABAP Fi...	A...	Semant
HeaderTxt	☐	Edm.Str...	0	0	25		☐	☐	☐	☐	☐	Doc.He...	T		HEADE...		
CompCode	☑	Edm.Str...	0	0	4		☐	☐	☐	☐	☐	Compa...	T		COMP_...		
DocDate	☐	Edm.Da...	7	0	0		☐	☐	☐	☐	☐	Docum...	T		DOC_D...		
PstngDate	☐	Edm.Da...	7	0	0		☐	☐	☐	☐	☐	Posting ...	T		PSTNG_...		
DocType	☐	Edm.Str...	0	0	2		☐	☐	☐	☐	☐	Docum...	T		DOC_T...		
RefDocNoLong	☐	Edm.Str...	0	0	35		☐	☐	☐	☐	☐	Referen...	T		REF_D...		
BusTransactionType	☐	Edm.Str...	0	0	4		☐	☐	☐	☐	☐	Bus. Tr...	T		BUS_TR...		

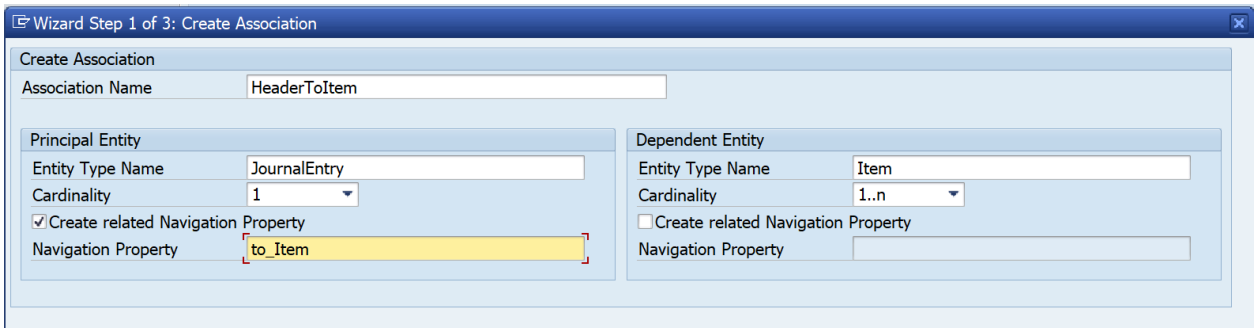
8. In the left-hand tree structure, right-click on **Data Model** and choose **Create** -> **Entity Type**.
9. Give the entity type a name as 'Item'
10. After creating the entity type, you can add **properties** (fields/attributes) that correspond to the attributes of your data model.

Name	Is Key	Edm Ty...	Prec.	Scale	Max ...	Unit Pro...	Creat...	Upda...	Sorta...	Nulla...	Filt.	Label	La...	Comp. ...	ABAP Fi...	A...	Ser
ReferenceDocumentItem	☑	Edm.Str...	0	0	10		☐	☐	☐	☐	☐	Position	T		ITEMN...		
Supplier	☐	Edm.Str...	0	0	10		☐	☐	☐	☐	☐	Supplier	T		VENDO...		
CompanyCode	☑	Edm.Str...	0	0	4		☐	☐	☐	☐	☐	Compa...	T		COMP_...		
PaymentMethod	☐	Edm.Str...	0	0	1		☐	☐	☐	☐	☐	Pymt m...	T		PYMT_...		
PaymentBlock	☐	Edm.Str...	0	0	1		☐	☐	☐	☐	☐	Pmnt bl...	T		PMNT_...		
Assignment	☐	Edm.Str...	0	0	18		☐	☐	☐	☐	☐	Assign...	T		ALLOC_...		
Currency	☐	Edm.Str...	0	0	5		☐	☐	☐	☐	☐	Paymen...	T		PYMT_...		
Amount	☐	Edm.De...	23	4	0		☐	☐	☐	☐	☐	Pmnt/c ...	T		PYMT_...		
SpecialGLIndicator	☐	Edm.Str...	0	0	1		☐	☐	☐	☐	☐	Special ...	T		SP_GL_...		
TaxCode	☐	Edm.Str...	0	0	2		☐	☐	☐	☐	☐	Tax Code	T		TAX_C...		
DueOn	☐	Edm.Da...	7	0	0		☐	☐	☐	☐	☐	Tax date	T		TAX_DA...		
WbsElement	☐	Edm.Str...	0	0	24		☐	☐	☐	☐	☐	WBS El...	T		WBS_E...		

11. In the left-hand tree structure, right-click on **Data Model** and choose **Create** -> **Entity Type**.
12. Give the entity type a name as 'Return'
13. After creating the entity type, you can add **properties** (fields/attributes) that correspond to the attributes of your data model.

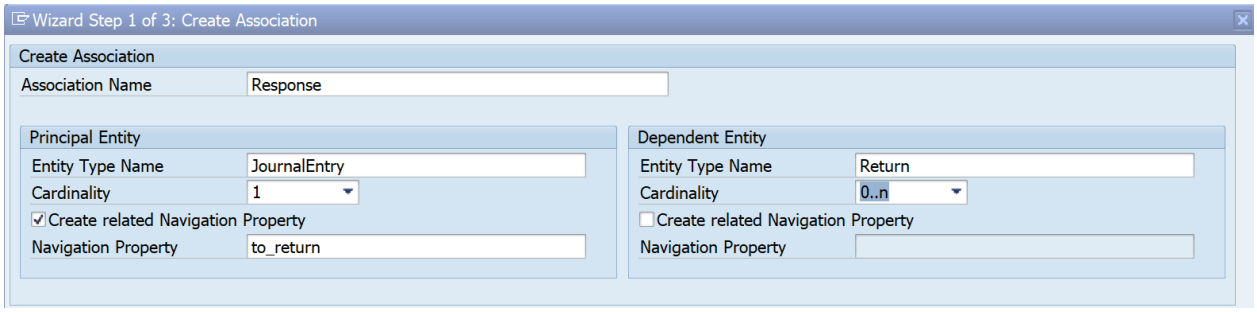
Name	Is Key	Edm Ty...	Prec.	Scale	Max ...	Unit Pro...	Creat...	Upda...	Sorta...	Nulla...	Filt.	Label	La...	Comp. ...	ABAP Fi...	A...	Semanti...
Type	☑	Edm.Str...	0	0	1		☐	☐	☐	☐	☐	Messag...	T		TYPE		
Id	☑	Edm.Str...	0	0	20		☐	☐	☐	☐	☐	Messag...	T		ID		
Number	☑	Edm.Str...	0	0	3		☐	☐	☐	☐	☐	Messag...	T		NUMBER		
Message	☐	Edm.Str...	0	0	220		☐	☐	☐	☐	☐	Messag...	T		MESSA...		

14. Create an association by right clicking on Associations→ create and enter the details as given below.



15. After entering the details click on next and select the field to establish the relationship.

16. Create an association by right clicking on Associations→ create and enter the details as given below.



17. After entering the details click on next and select the field to establish the relationship.

Principal Entity	Principal Key	Dependent Entity	Dependent Property
JournalEntry	CompCode	Return	Number

18. After defining entity types and services, generate runtime objects using **Service Builder**, this will generate the necessary classes in the backend.

Runtime Artifacts

Name	Generated Artifact Type	Program ID	Object Type	Object Name
ZAPI_JE_CREATE_MDL	Registered Model	R3TR	IWMO	ZAPI_JE_CREATE_MDL
ZAPI_JE_CREATE_SRV	Registered Service	R3TR	IWSV	ZAPI_JE_CREATE_SRV
ZCL_ZAPI_JE_CREATE_DPC	Data Provider Base Cla	R3TR	CLAS	ZCL_ZAPI_JE_CREATE...
ZCL_ZAPI_JE_CREATE_DPC_EXT	Data Provider Extensio	R3TR	CLAS	ZCL_ZAPI_JE_CREATE...
ZCL_ZAPI_JE_CREATE_MPC	Model Provider Base Cl	R3TR	CLAS	ZCL_ZAPI_JE_CREATE...
ZCL_ZAPI_JE_CREATE_MPC_EXT	Model Provider Extensi	R3TR	CLAS	ZCL_ZAPI_JE_CREATE...

19. Redefine the method "/iwbep/if_mgw_appl_srv_runtime~create_deep_entity" and add the code within the Method & Endmethod given as below.

20. METHOD /iwbep/if_mgw_appl_srv_runtime~create_deep_entity.

```
DATA: BEGIN OF ls_payload.
      INCLUDE TYPE zcl_zapi_je_create_mpc=>ts_journalentry.
DATA :  to_accpay TYPE TABLE OF zcl_zapi_je_create_mpc=>ts_item,
       to_return TYPE TABLE OF zcl_zapi_je_create_mpc=>ts_return,
     END OF ls_payload.

DATA : ls_doc_header     TYPE bapiache09,
       lv_obj_type       TYPE bapiache09-obj_type,
       lv_obj_key        TYPE bapiache09-obj_key,
       lv_obj_sys        TYPE bapiache09-obj_sys,
       lt_gl_acc         TYPE TABLE OF bapiacgl09,
       lt_acc_pay        TYPE TABLE OF bapiacap09,
       lt_curr_amnt      TYPE TABLE OF bapiaccr09,
       lt_return         TYPE TABLE OF bapiret2,
       ls_return         TYPE bapiret2,
       lv_entity_set_name TYPE /iwbep/mgw_tech_name.

DATA(exception) = NEW /iwbep/cx_mgw_busi_exception( ).
DATA(msg_container) = exception->get_msg_container( ).

* Get Entity Set Name
lv_entity_set_name    = io_tech_request_context->get_entity_set_name( ).

CASE lv_entity_set_name.
 WHEN 'JournalEntrySet'.

  io_data_provider->read_entry_data(
 IMPORTING
  es_data = ls_payload ).

  ls_doc_header-username    = sy-uname.
  ls_doc_header-header_txt  = ls_payload-header_txt.
  ls_doc_header-comp_code   = ls_payload-comp_code.
  ls_doc_header-doc_date    = sy-datum.
  ls_doc_header-pstng_date  = sy-datum.
  ls_doc_header-doc_type    = ls_payload-doc_type.
  ls_doc_header-ref_doc_no  = ls_payload-ref_doc_no_long.
  ls_doc_header-bus_transaction_type  = ls_payload-bus_transaction_type.

  LOOP AT ls_payload-to_accpay ASSIGNING FIELD-SYMBOL(<fs_accpay>).
   APPEND VALUE #( itemno_acc  = <fs_accpay>-itemno_acc
                   vendor_no   = <fs_accpay>-vendor_no
                   comp_code   = <fs_accpay>-comp_code
                   pymt_meth   = <fs_accpay>-pymt_meth
                   pmnt_block  = <fs_accpay>-pmnt_block
                   alloc_nmbr  = <fs_accpay>-alloc_nmbr
                   sp_gl_ind   = <fs_accpay>-sp_gl_ind
                   tax_code    = <fs_accpay>-tax_code
                   tax_date    = <fs_accpay>-tax_date
                   wbs_element = <fs_accpay>-wbs_element
```

```
            ) TO lt_acc_pay.
ENDLOOP.

LOOP AT ls_payload-to_accpay ASSIGNING FIELD-SYMBOL(<fs_curramnt>).
 APPEND VALUE #( itemno_acc  = <fs_curramnt>-itemno_acc
              currency    = <fs_curramnt>-pymt_cur
              amt_doccur  = <fs_curramnt>-pymt_amt
             ) TO lt_curr_amnt.
ENDLOOP.

CALL FUNCTION 'BAPI_ACC_DOCUMENT_POST'
 EXPORTING
  documentheader = ls_doc_header
 IMPORTING
  obj_type      = lv_obj_type
  obj_key       = lv_obj_key
  obj_sys       = lv_obj_sys
 TABLES
  accountpayable = lt_acc_pay
  currencyamount = lt_curr_amnt
  return        = lt_return.

CLEAR ls_payload-to_return.
IF line_exists( lt_return[ type = 'E' ] ).
 CALL FUNCTION 'BAPI_TRANSACTION_ROLLBACK'.

 LOOP AT lt_return ASSIGNING FIELD-SYMBOL(<fs_return>).
  APPEND VALUE #( type       = <fs_return>-type
               id         = <fs_return>-id
               number     = <fs_return>-number
               message    = <fs_return>-message
              ) TO ls_payload-to_return.
 ENDLOOP.

ELSE.
 CALL FUNCTION 'BAPI_TRANSACTION_COMMIT'
  EXPORTING
   wait = abap_true.

 LOOP AT lt_return ASSIGNING FIELD-SYMBOL(<fs_ret>).
  APPEND VALUE #( type       = <fs_ret>-type
               id         = <fs_ret>-id
               number     = <fs_ret>-number
               message    = <fs_ret>-message
              ) TO ls_payload-to_return.

 ENDLOOP.

ENDIF.
copy_data_to_ref(
  EXPORTING
```

```
        is_data = ls_payload
      CHANGING
        cr_data = er_deep_entity
        ).
  ENDCASE.

 ENDMETHOD.
```

21. After implementing the logic, register and activate the service using the **/IWFND/MAINT_SERVICE** transaction and test the service in the SAP Gateway Client (/IWFND/GW_CLIENT).

Support

There is no support available for template content. If you experience general issues with SAP Build Process Automation, please follow the links below:

- Troubleshoot Support Guided Answer
- Test an Automation Help portal Article
- Test an Application Help portal Article
- Get traces Support Guided Answer

If the issue persists, please raise an incident via SAP Support Portal on **BPI-PA-BPC** component.

This template can be modified using SAP Process Automation. For example: add decisions to determine the approver based on a certain business logic or add an action to post collected information to backend SAP system. Refer help portal for more details.

www.sap.com