AI Agent Studio. 26A.

Agent Team "Quote to Purchase Requisition Chat Assistant".

Got this Agent Team to work+some changes applied+PDF demo file provided.

Introduction/Goals.

We will try to achieve these goals:

- Get the Agent Team flow working from start to finish.
- Create a working PDF sample (provided).
- Understand the new feature "upload file" in chat.
- Understand how every node is built (this part will be up to you opening the provided original template and analyzing it).
- Make changes, as our users could ask for new requirements, we should be able to make adaptations.

Preparatory steps.

1. Make sure your user has the proper roles and data access, and is able to create purchase requisitions.

   Tip:

   In SaaS, in Setup and Maintenance in Procurement area, search "Procurement Agent".
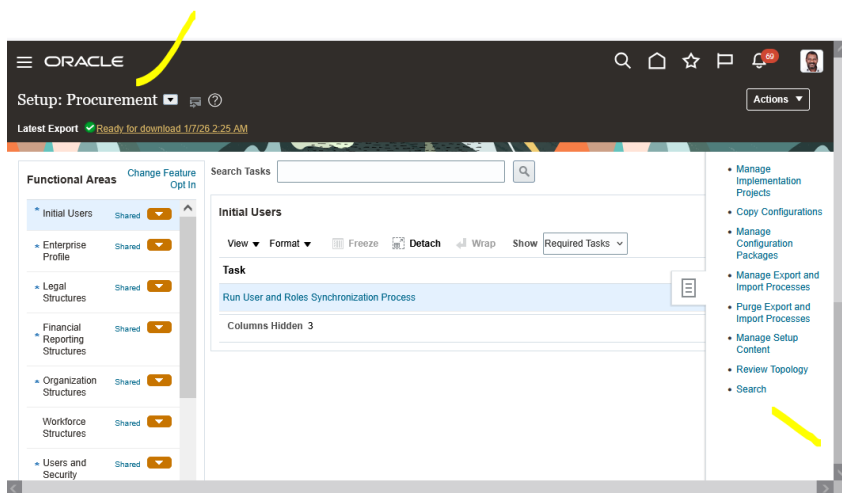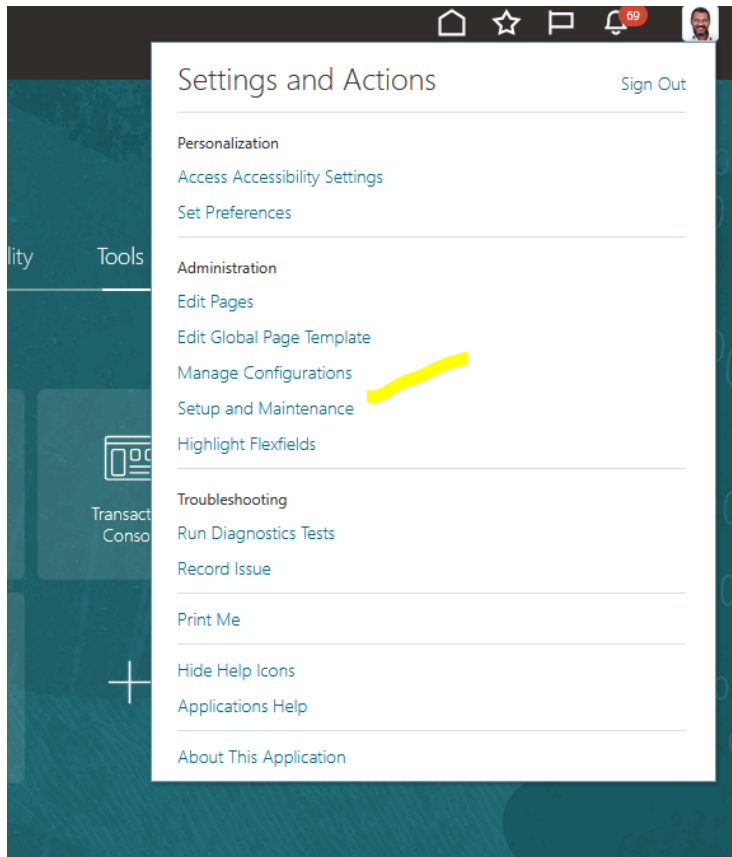
Search

procurement agent

Match With Tasks, Task Lists, Business Objects

Name	Type	Details
Configure Procurement Business Function, Procurement Agents, and Suppliers	Task	
Define Procurement Agents	Task List	Task Count: 0
Manage Procurement Agents	Task	
Procurement Agent	Business Object	Task Count: 1
Procurement Agent Access	Business Object	Task Count: 1

Then make sure your user has this setup, or ask for functional help.

Edit Procurement Agent: Feitty, Curtis ⑦

Procurement BU   US1 Business Unit
Agent   Feitty, Curtis
Default Requisitioning BU   US1 Business Unit
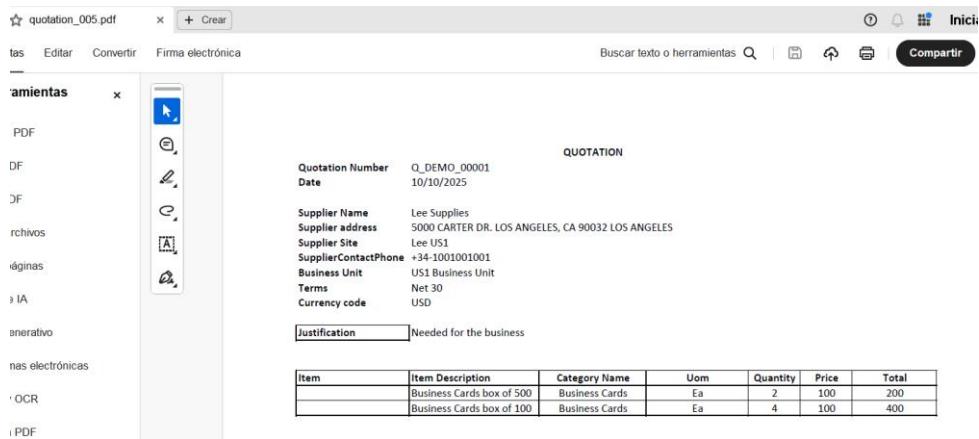Default Printer
Status   Active

Agent Access

View ▾

Action	Allowed	Access to Other Agents' Documents
Manage Requisitions	✔	Full
Manage Purchase Orders	✔	Full
Manage Purchase Agreements	✔	None
Manage Negotiations	✔	None
Manage Sourcing Programs	✔	None

2. Build an excel file with data to be used as Quotation Document.

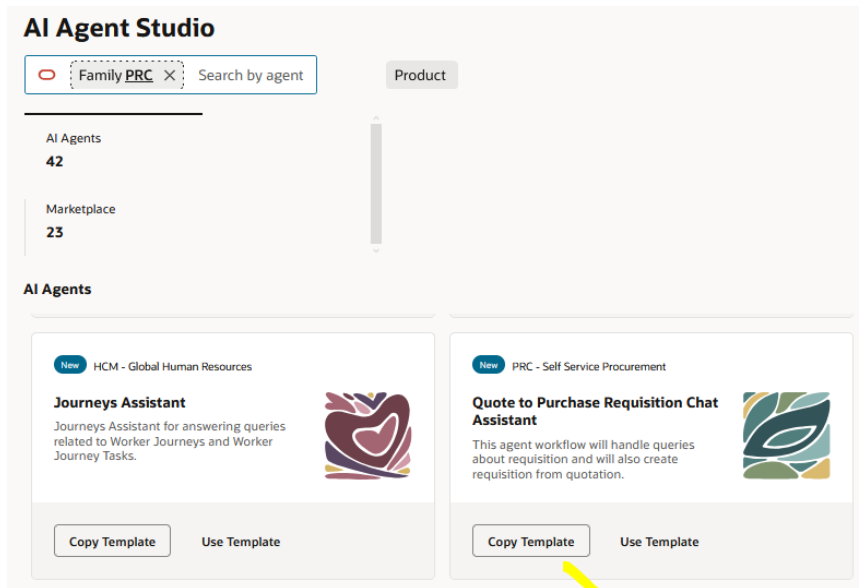|	QUOTATION					
Quotation Number	Q_DEMO_00001					
Date	10/10/2025					
Supplier Name	Lee Supplies					
Supplier address	5000 CARTER DR. LOS ANGELES, CA 90032 LOS ANGELES					
Supplier Site	Lee US1					
SupplierContactPhone	+34-1001001001					
Business Unit	US1 Business Unit					
Terms	Net 30					
Currency code	USD					
Justification	Needed for the business					
Item	**Item Description**	**Category Name**	**Uom**	**Quantity**	**Price**	**Total**
	Business Cards box of 500	Business Cards	Ea	2	100	200
	Business Cards box of 100	Business Cards	Ea	4	100	400

3. Save as PDF. It will be our file for testing.

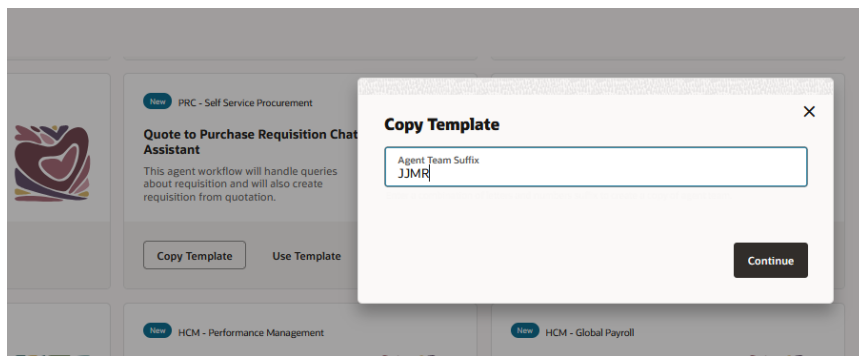As you can see, some "trial and error" was needed.

Steps/changes applied.

1. Select template and make your copy.
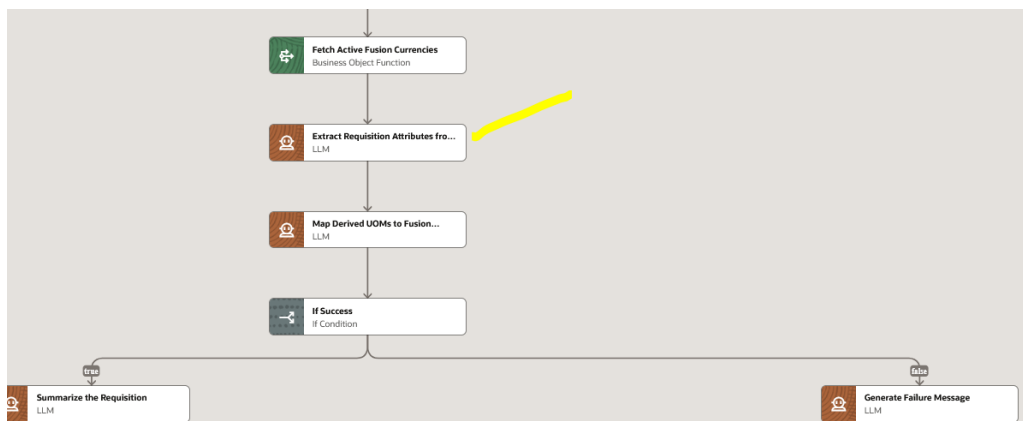
Filter, select and copy.

Add your prefix.

Next, we will provide the changes applied to get the template working and other changes for learning purposes.

2. Navigate to node "Extract Requisition Attributes from Supplier Quotation".

Extract Requisition Attributes from Supplier Quotation
LLM

Name
Extract Requisition Attributes from Supplier Quotation

Code
EXTRACT_REQUISITION_ATTRIBUTES_FROM_SUPPLIER_QUOTATION

Prompt
EXTRACTION CHALLENGE:
You're analyzing the following supplier quotation document with varied formats. {{$context.$nodes.FILE_PROCESSOR.$output.fileText}}

The document consists of header information and line information.
a. Header information can appear anywhere in the document and in any format.
b. Line information can appear in different structures:
- Formal tables with borders
- Invisible tables (aligned text without visible borders)

Changes in Prompt.

Added by us for learning purposes.

These are the attributes for the header:
1. PreparerId: {{$context.$nodes.GET_ACTIVE_REQUISITION_PREFERENCE.$output.items[0].PersonId}}
2. Description: Provide summary of the document and include the quotation number and company issuing the quotation. It must not exceed 240 characters.
3. SupplierName: Name of the company issuing the supplier quotation document. This is the company providing the goods or services. Look for this information only in cle
"Vendor," "From," or company information next to the quotation title or logo text. Never extract names or addresses from sections labeled "Customer", "Client", "Bill To", "SI
company name is the supplier, leave SupplierName blank and do not extract any supplier-related fields in either header or line information.
4. NormalizedSupplierName: Please create a shortened, standardized version of this supplier name by:
- Removing corporate designations (LLC, Inc, Corp, Corporation, Ltd, Limited, etc.)
- Removing subsidiary indicators (A division of, A subsidiary of, etc.)
- Removing regional identifiers (North America, USA, etc.) unless they are critical to identify the company
- Standardizing common abbreviations (Intl -> International, Mfg -> Manufacturing)
- Keeping the core business name that would be recognizable

5. SupplierContact: Name of the supplier contact. It must not exceed 80 characters.
6. SupplierContactEmail: contact email address of the company issuing the document.
7. SupplierContactEmailDomain: Extract the email domain from supplier contact email if it is not from the public email domain. Otherwise, set this to null.
8. SupplierSite: Location of the supplier associated with the quotation. This may refer to the supplier's physical address, branch, or facility responsible for fulfilling the order.
fulfilling the order, found under sections like "Send Orders To" or "Issue PO To".
9. SupplierContactPhone: Phone number of the company issuing the document.
10. SupplierContactFax: Fax number of the company issuing the document.
11. CurrencyCode: Steps to find currency match:
- Look for Currency field and Currency symbol from the quotation document
- If currency is not explicitly defined, determine the country from the supplier address and use the local currency of that country.
- Find the matching entry from the list of internal currencies {{$context.$nodes.FETCH_ACTIVE_FUSION_CURRENCIES.$output}}
12. QuotationNumber: A unique identifier assigned to the supplier's quotation.
13. AttachmentFileName: {{$context.$nodes.FILE_PROCESSOR.$output.fileName}}
14. Justification: Justification from the quotation document.

Text:

14. Justification: Justification from the quotation document.

Added by us for learning purposes.

6. Quantity: if the quantity is not provided, default to 1. Quantity must not be null for "Goods" and "Rate Based Services" lineType.
7. Unit Price: Number that identifies unit price of the item. It may be labeled as "Price", "Unit Price", "Customer price" or "Rate". Extract the numeric value regardless of whether the currency symbol i
line type as "Fixed Price Services", do not create a line for this item.
8. ManufacturerName: Manufacturer name of the item.Extract ManufacturerName by carefully identifying manufacturer information based on the item description or explicitly labeled fields.
9. ManufacturerPartNumber: Manufacturer part number. Only populate this if ManufacturerName is also present for that line.
10. CurrencyCode: default to "USD".
11. RequesterId: {{$context.$nodes.GET_ACTIVE_REQUISITION_PREFERENCE.$output.items[0].RequesterId}}
12. DestinationTypeCode: {{$context.$nodes.GET_ACTIVE_REQUISITION_PREFERENCE.$output.items[0].DestinationTypeCode}}
13. DestinationOrganizationId: {{$context.$nodes.GET_ACTIVE_REQUISITION_PREFERENCE.$output.items[0].DestinationOrganizationId}}
14. DeliverToLocationId: {{$context.$nodes.GET_ACTIVE_REQUISITION_PREFERENCE.$output.items[0].DeliverToLocationId}}

ⓘ Type {{ to start seeing context-sensitive expressions.

˅ **Output**

Text.

10. CurrencyCode: default to "USD".

11. RequesterId: {{$context.$nodes.GET_ACTIVE_REQUISITION_PREFERENCE.$output.items[0].RequesterId}}

12. DestinationTypeCode: {{$context.$nodes.GET_ACTIVE_REQUISITION_PREFERENCE.$output.items[0].DestinationTypeCode}}

13. DestinationOrganizationId:
{{$context.$nodes.GET_ACTIVE_REQUISITION_PREFERENCE.$output.items[0].DestinationOrganizationId}}

14. DeliverToLocationId: {{$context.$nodes.GET_ACTIVE_REQUISITION_PREFERENCE.$output.items[0].DeliverToLocationId}}

Changes in Output.


```json
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "properties": {
    "header": {
      "type": "object",
      "properties": {
        "PreparerId": { "type": "string" },
        "RequisitionDescription": { "type": "string" },
        "SupplierNameFromDocument": { "type": "string" },
        "NormalizedSupplierName": { "type": "string" },
        "SupplierContactEmail": { "type": "string", "format": "email" },
        "SupplierContact": { "type": "string" },
        "SupplierContactPhone": { "type": "string" },
        "SupplierContactFax": { "type": "string" },
        "SupplierSite": { "type": "string" },
        "SupplierContactEmailDomain": { "type": "string" },
        "CurrencyCode": { "type": "string" },
        "QuotationNumber": { "type": "string" },
        "Date": { "type": "string" },
        "Justification": { "type": "string" },
        "AttachmentFileName": { "type": "string" }
      },
      "required": [
        "PreparerId",
        "RequisitionDescription",
        "SupplierNameFromDocument",
        "NormalizedSupplierName",
        "SupplierContactEmail",
        "SupplierContact",
        "SupplierContactPhone",
        "SupplierContactFax",
        "SupplierSite",
        "SupplierContactEmailDomain",
        "CurrencyCode",
        "AttachmentFileName"
      ]
    },
    "lines": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "LineNumber": { "type": "string" },
          "CurrencyCode": { "type": "string" },
          "RequesterId": { "type": "string" },
          "DestinationTypeCode": { "type": "string" },
          "DestinationOrganizationId": { "type": "string" },
          "DeliverToLocationId": { "type": "string" },
          "CategoryName": { "type": "string" },
          "ItemDescription": { "type": "string" },
```

AI for Fusion Cookbook 2026

```
          "SupplierItemNumber": { "type": "string" },
          "Quantity": { "type": "number" },
          "UOM": { "type": "string" },
          "Price": { "type": "number" },
          "LineTypeCode": { "type": "string" },
          "ManufacturerName": { "type": "string" },
          "ManufacturerPartNumber": { "type": "string" }
        },
        "required": [
          "LineNumber",
          "ItemDescription",
          "Price",
          "LineTypeCode"
        ]
      }
    }
  },
  "required": ["header", "lines"]
}
```

3. We have removed this node.
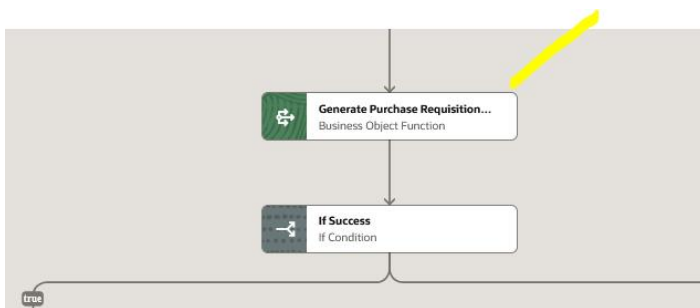
Reason: after many tests getting errors related to Category of the item, we have decided to have more control, generating our detailed payload and using the normal POST for purchase requisitions.

This drives us to make some changes in below nodes.

This removed node generated payload using this REST.

After removing, some renaming in next nodes are required.

4. New Business Object.

Name: AA_CREATE_PURCHASE_REQ.

Description: Create purchase requisition from PDF.

Function-scenario.

Name: create_purchase_requisition

Operation type: POST.

Description: Create purchase requisition.

Resource Path: https://fa-etit-saasfademo1.ds-fa.oraclepdemos.com/fscmRestApi/resources/11.13.18.05/purchaseRequisitions

Edit scenario - AA_CREATE_PURCHASE_REQ

Name	Description
create_purchase_requisition	Create purchase requisition

Operation Type	Resource Path
HTTP POST	/fscmRestApi/resources/11.13.18.05/purchaseRequisitions

Use native authentication Header

Operation Payload
```
{
    "RequisitioningBUId": {pRequisitioningBUId},
    "PrepareId": {pPrepareId},
    "ExternallyManagedFlag": false,
    "Description": "POST-ing a Requisition Header from REST",
    "InterfaceSourceCode": "TestSS1",
    "Justification": {pJustification},
        "lines": {pLines}
}
```

Operation Payload.

```
{
    "RequisitioningBUId": {pRequisitioningBUId},
    "PreparerId": {pPreparerId},
    "ExternallyManagedFlag": false,
    "Description": "POST-ing a Requisition Header from REST",
    "InterfaceSourceCode": "TestSS1",
    "Justification": "{pJustification}",
        "lines": {pLines}
}
```

Parameters.

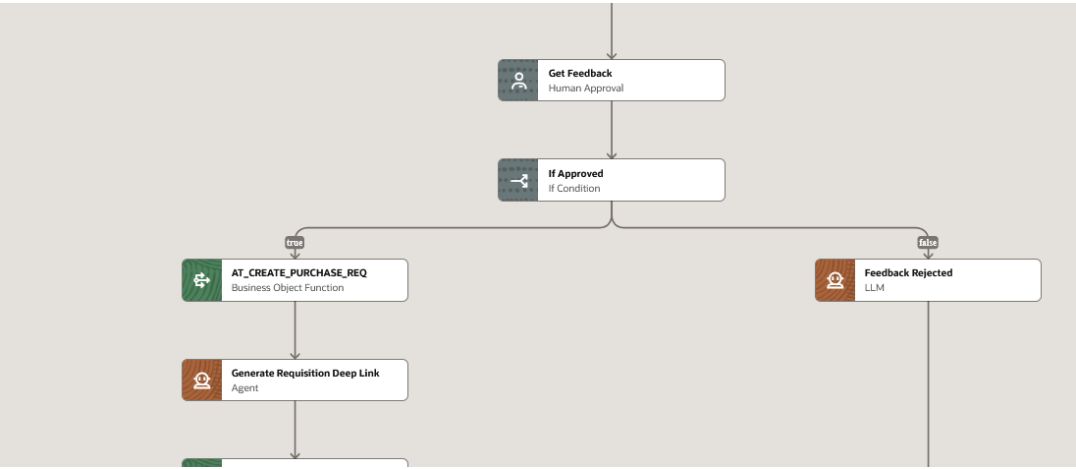### Parameters and Business Object Fields

Add Parameter   Add Field   Add Field from Specification

Name	Data Type	Description	Is Parameter
pRequisitioningBUId	string	pRequisitioningBUId	true
pPreparerId	string	pPreparerId	true
pJustification	string	pJustification	true
pLines	string	pLines	true

5. Business Object Function.

Added here.

Name: AT_CREATE_PURCHASE_REQ.

Detail.

Parameters:


```
{{$context.$nodes.GET_ACTIVE_REQUISITION_PREFERENCE.$output.items[0].RequisitioningBUId}
}
```

```
{{$context.$nodes.EXTRACT_REQUISITION_ATTRIBUTES_FROM_SUPPLIER_QUOTATION.$output.header.
PreparerId}}
```



```
{{$context.$nodes.EXTRACT_REQUISITION_ATTRIBUTES_FROM_SUPPLIER_QUOTATION.$output.lines}}
```



```
{{$context.$nodes.EXTRACT_REQUISITION_ATTRIBUTES_FROM_SUPPLIER_QUOTATION.$output.header.
Justification}}
```

Output.
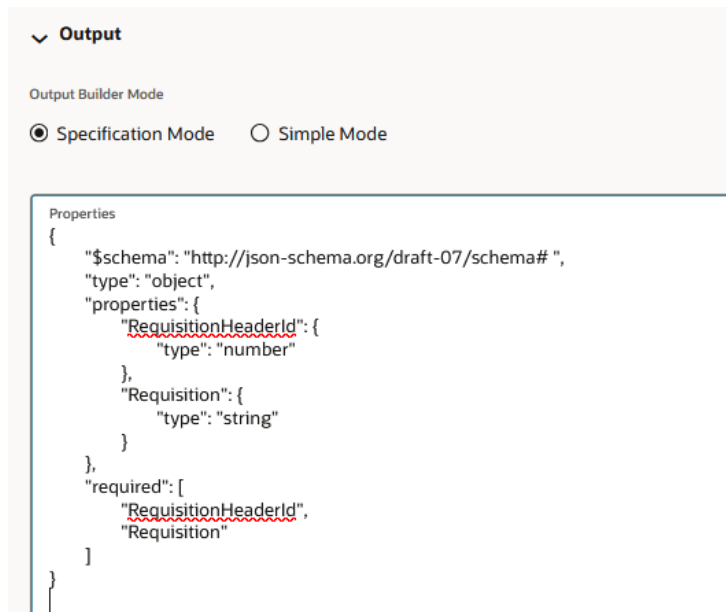

```json
{
        "$schema": "http://json-schema.org/draft-07/schema# ",
        "type": "object",
        "properties": {
                "RequisitionHeaderId": {
                        "type": "number"
                },
                "Requisition": {
                        "type": "string"
                }
        },
        "required": [
                "RequisitionHeaderId",
                "Requisition"
        ]
}
```

Note.

In case of new/remove/rename parameters (my case for learning purposes), you will need to remove and then add Business Object in Agent Team to refresh parameters.

Afterwards, you will need to redo the mappings.

6. Generate Requisition Deep Link.
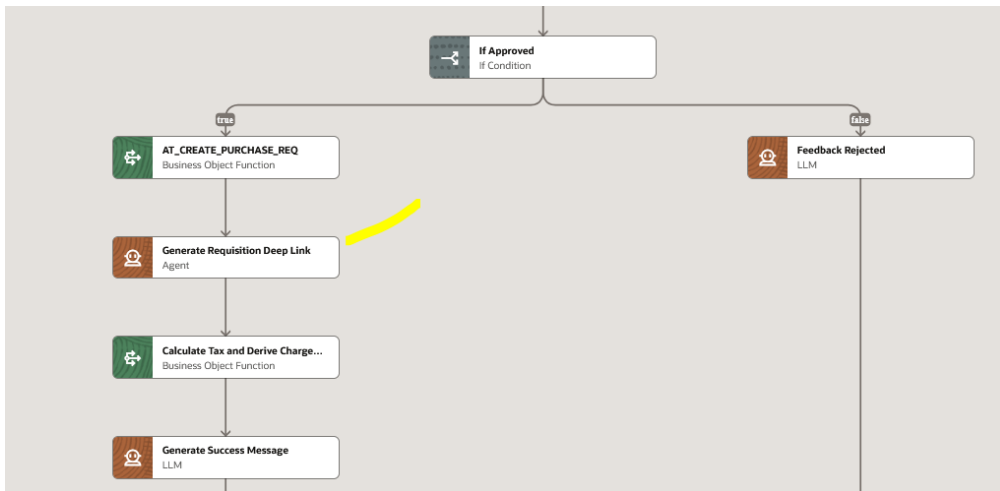
Changes.

Input Variables

Name
message

Value
{{"Create a deeplink for requisition "+ $context.$nodes.ATCREATEPURCHASEREQ.$output.Requisition}}

Input Variables

message

Name
RequisitionHeaderId

Value
{{$context.$nodes.ATCREATEPURCHASEREQ.$output.RequisitionHeaderId}}

ⓘ Type {{ to start seeing context-sensitive expressions.

7. Calculate Tax and Derive Charge Account.

We have removed this node for simplicity.

Feel free to make adaptations to get it working.

Test.

Note: as several "trial and error" actions has been required, some id's in screenshots does not match, this does not affect the explanation.

Click on here to start the test:
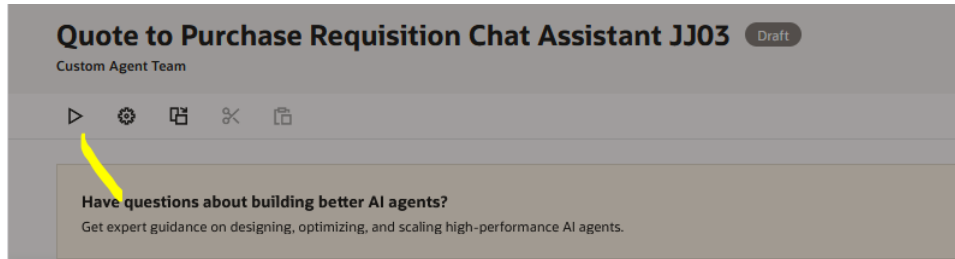
Afterwards, click on here to upload file from your laptop:

Select PDF file (provided in GitHub, see below).

Type this text and Press enter.

"Please, create purchase requisition from file uploaded".
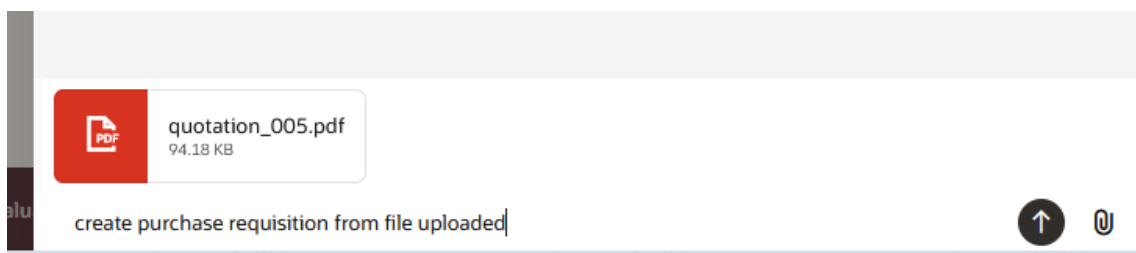
It will take a while to run all the nodes before the "human" node that ask for "Approve".
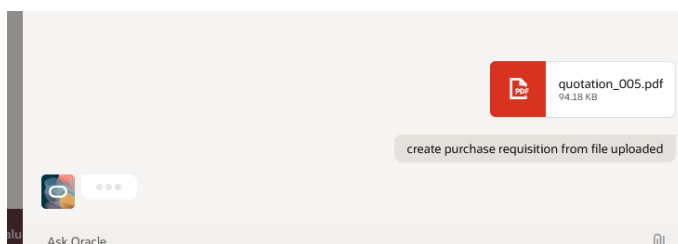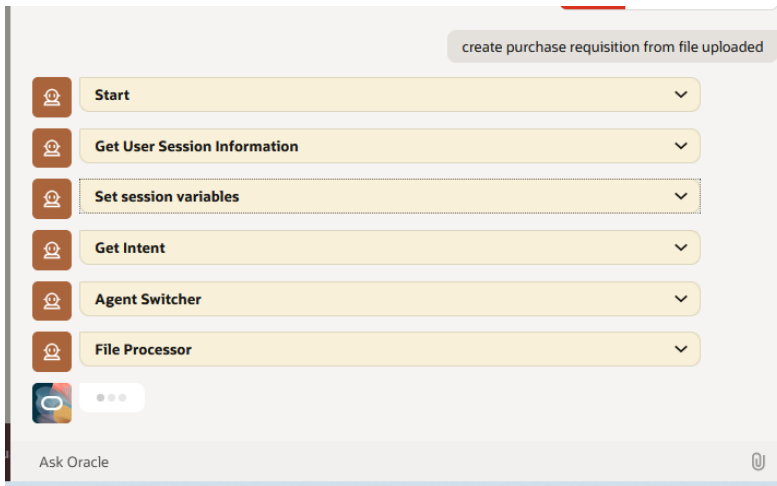
This node is quite important as it provides many fields to fill the payload (described in Prompt).
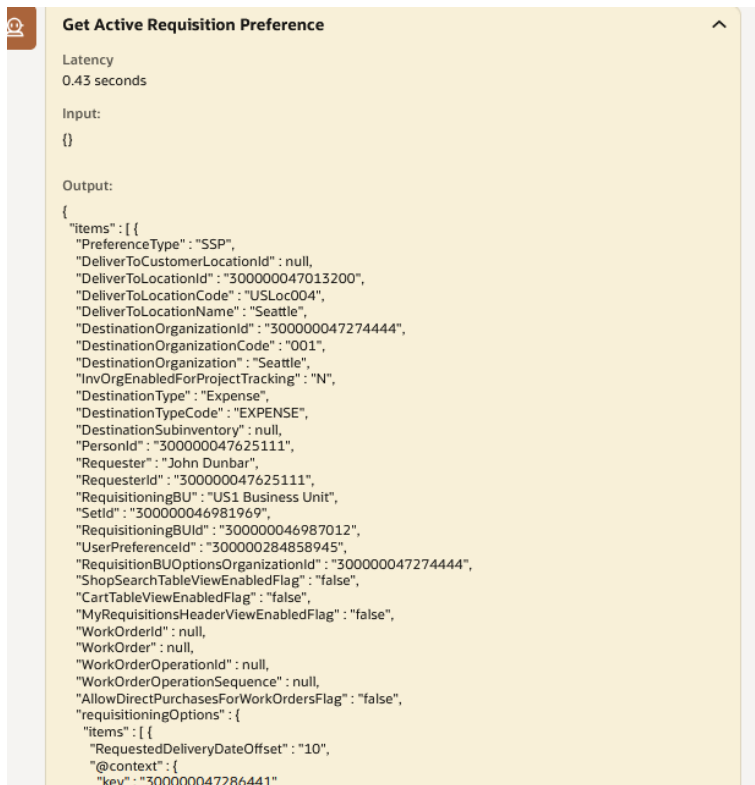
Based in this REST API.
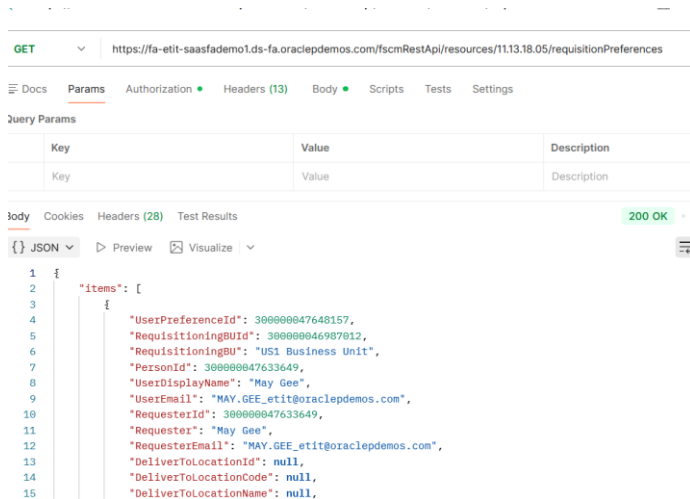
This node is the one that does the magic of reading the file.

Before calling the Creation node the user will have to "Approve".

Header part.

AT_CREATE_PURCHASE_REQ

Latency
1.97 seconds

Input:

{"pRequisitioningBUId":300000046987012,"pPreparerId":300000047625111,"pLines":
[{"CurrencyCode":"USD","LineTypeCode":"Goods","DestinationTypeCode":"EXPENSE","UOM":"Ea","Price
":100,"DestinationOrganizationId":"300000047274444","DeliverToLocationId":"300000047013200","R
equesterId":"300000047625111","CategoryName":"Business
Cards","Quantity":2,"ItemDescription":"Business Cards box of 500 Business Cards","LineNumber":"1"},
{"CurrencyCode":"USD","LineTypeCode":"Goods","DestinationTypeCode":"EXPENSE","UOM":"Ea","Price":
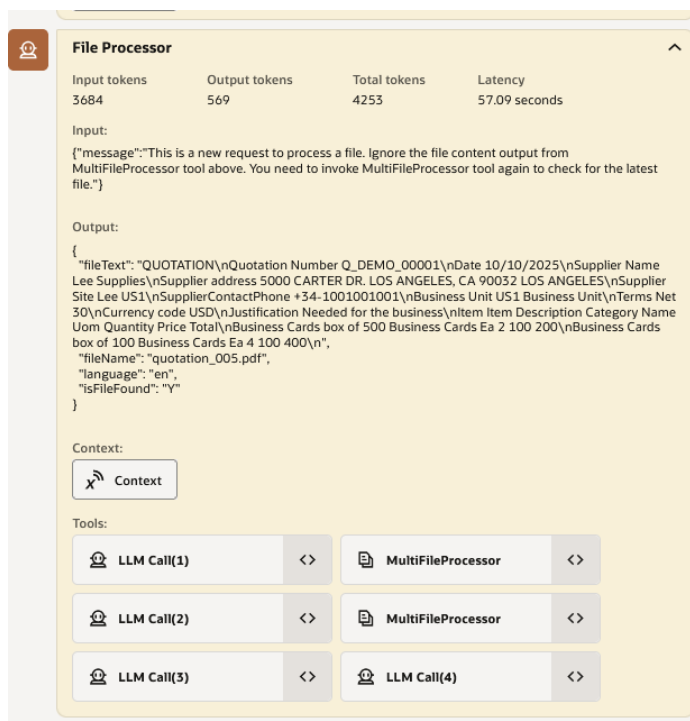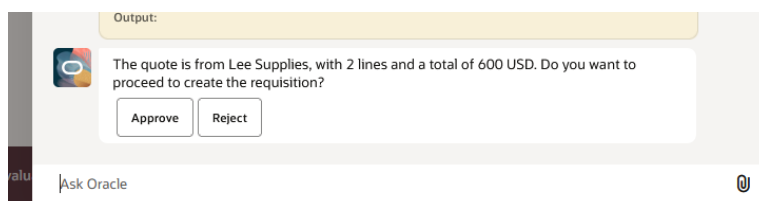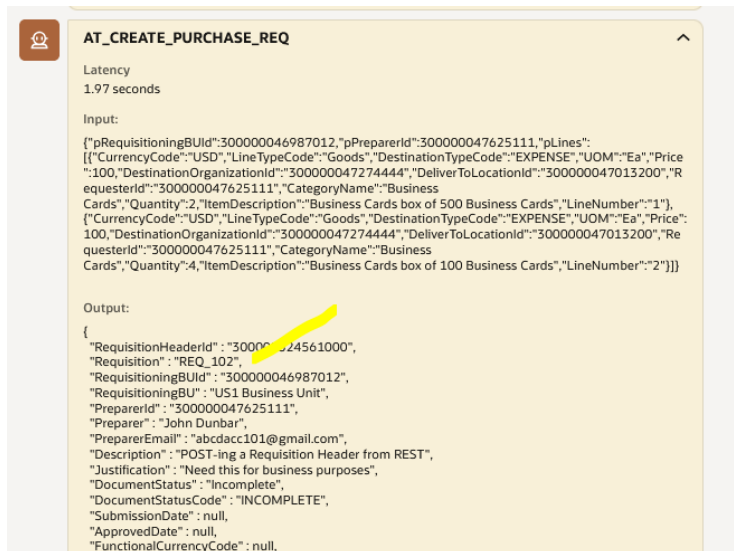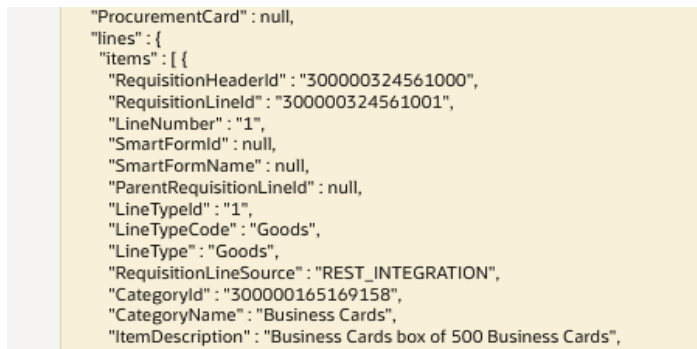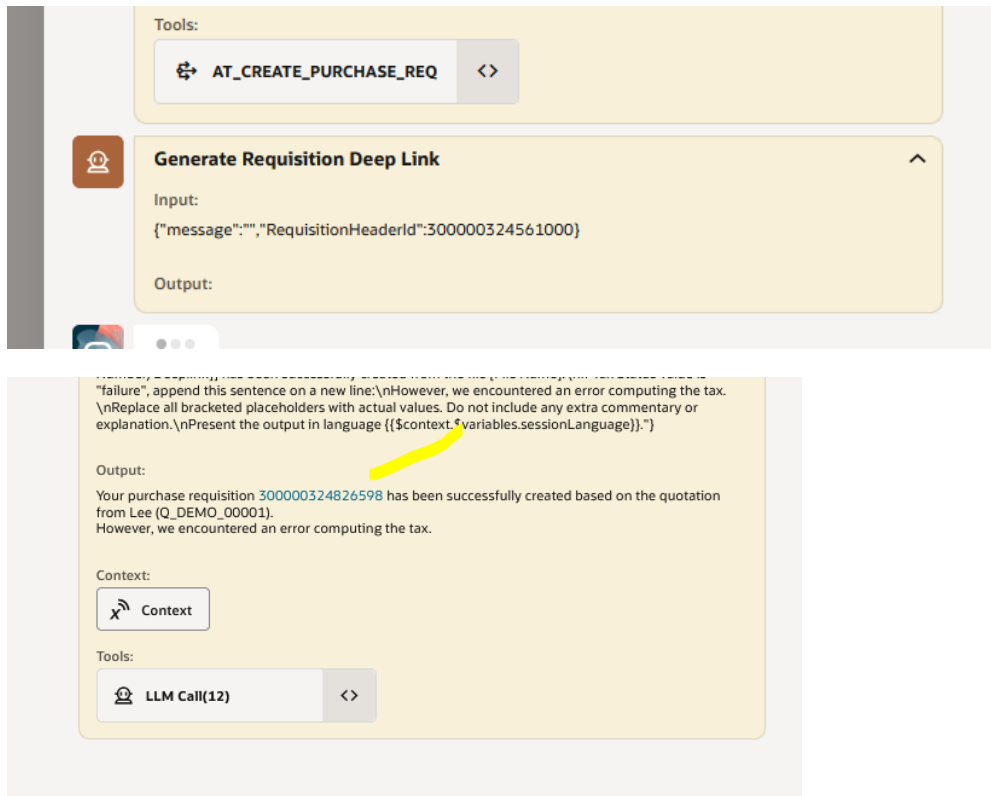100,"DestinationOrganizationId":"300000047274444","DeliverToLocationId":"300000047013200","Re
questerId":"300000047625111","CategoryName":"Business
Cards","Quantity":4,"ItemDescription":"Business Cards box of 100 Business Cards","LineNumber":"2"}]}

Output:

{
  "RequisitionHeaderId" : "300000324561000",
  "Requisition" : "REQ_102",
  "RequisitioningBUId" : "300000046987012",
  "RequisitioningBU" : "US1 Business Unit",
  "PreparerId" : "300000047625111",
  "Preparer" : "John Dunbar",
  "PreparerEmail" : "abcdacc101@gmail.com",
  "Description" : "POST-ing a Requisition Header from REST",
  "Justification" : "Need this for business purposes",
  "DocumentStatus" : "Incomplete",
  "DocumentStatusCode" : "INCOMPLETE",
  "SubmissionDate" : null,
  "ApprovedDate" : null,
  "FunctionalCurrencyCode" : null,

Line 1 part.

  "ProcurementCard" : null,
  "lines" : {
    "items" : [{
      "RequisitionHeaderId" : "300000324561000",
      "RequisitionLineId" : "300000324561001",
      "LineNumber" : "1",
      "SmartFormId" : null,
      "SmartFormName" : null,
      "ParentRequisitionLineId" : null,
      "LineTypeId" : "1",
      "LineTypeCode" : "Goods",
      "LineType" : "Goods",
      "RequisitionLineSource" : "REST_INTEGRATION",
      "CategoryId" : "300000165169158",
      "CategoryName" : "Business Cards",
      "ItemDescription" : "Business Cards box of 500 Business Cards",

Line 2 part.

      }]
    }
  }, {
    "RequisitionHeaderId" : "300000324561000",
    "RequisitionLineId" : "300000324561002",
    "LineNumber" : "2",
    "SmartFormId" : null,
    "SmartFormName" : null,
    "ParentRequisitionLineId" : null,
    "LineTypeId" : "1",
    "LineTypeCode" : "Goods",
    "LineType" : "Goods",
    "RequisitionLineSource" : "REST_INTEGRATION",
    "CategoryId" : "300000165169158",
    "CategoryName" : "Business Cards",
    "ItemDescription" : "Business Cards box of 100 Business Cards",
    "ItemId" : null,
    "Item" : null,
    "ItemRevision" : null,
    "Quantity" : "4",
    "SecondaryQuantity" : null,
    "UnitPrice" : "100",
    "CurrencyCode" : "USD",
    "Currency" : "US Dollar",
    "UOMCode" : "zzu",
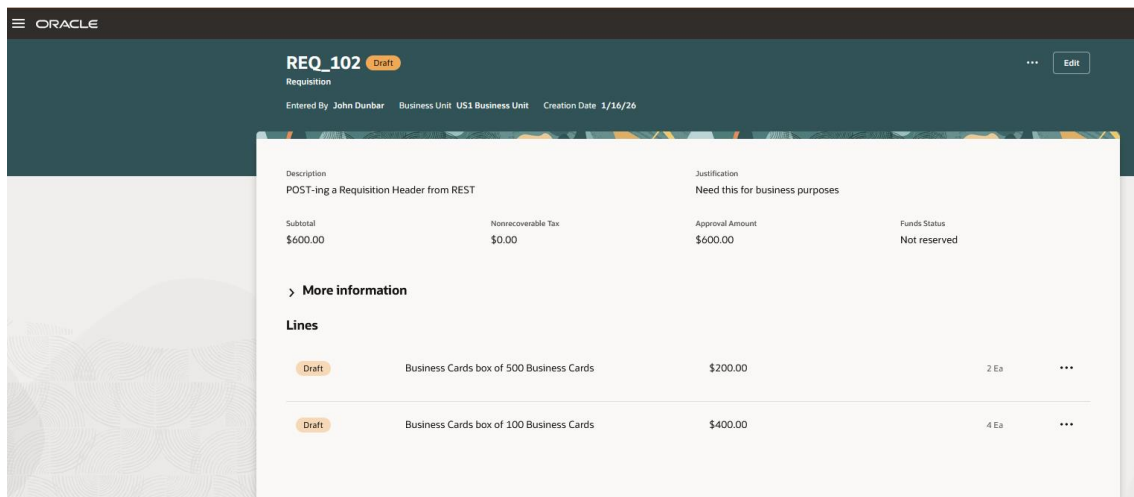   "UOM" : "Ea",
    "SecondaryUOMCode" : null

In our tests, It takes a while to generate the deep link.

Sample of URL generated for deeplink.

https://fa-etit-saasfademo1.ds-fa.oraclepdemos.com/fscmUI/redwood/procurement/requisitions/tabbar/shopping/requisitiondetails?requisitionHeaderId=300000324561000

We would get this:

Line 1.

Business Cards box of 500 Business Cards
Requisition REQ_102

Line

Line 1	Status Draft	Line Type Goods	Category Business Cards
Quantity 2 Ea	Price $100.00	Line Amount $200.00	Funds Status Not reserved

Source

Source Type Supplier	Agreement Type	Agreement	Agreement Line
Supplier	Supplier Site	Supplier Contact	Additional Contact Email
Negotiated No	Negotiation Required No	Buyer	Purchase Order
Negotiation			

Delivery

Requester John Dunbar	Requested Delivery Date	Destination Type Expense	Deliver-to Location Type Internal
Deliver-to Location Seattle	Deliver-to Address 411 University St, Seattle, WA 98101, King, United States		Urgent No

Additional information

Attribute Value

Line 2.

Business Cards box of 100 Business Cards
Requisition REQ_102

Line

Line 2	Status Draft	Line Type Goods	Category Business Cards
Quantity 4 Ea	Price $100.00	Line Amount $400.00	Funds Status Not reserved

Source

Source Type Supplier	Agreement Type	Agreement	Agreement Line
Supplier	Supplier Site	Supplier Contact	Additional Contact Email
Negotiated No	Negotiation Required No	Buyer	Purchase Order
Negotiation			

Delivery

Requester John Dunbar	Requested Delivery Date	Destination Type Expense	Deliver-to Location Type Internal
Deliver-to Location Seattle	Deliver-to Address 411 University St, Seattle, WA 98101, King, United States		Urgent No

Additional information

Attribute Value

Conclusion.

We have achieved these goals:

- Get the Agent Team flow working from start to finish.
- Create a working PDF sample (provided).
- Understand the new feature "upload file" in chat.
- Understand how every node is built (this part will be up to you opening the provided original template and analyzing it).
- Make changes, as our users could ask for new requirements, we should be able to make adaptations.

Technical

Code example	Comments
quotation_005.xlsx	To be used as reference.
quotation_005.pdf	To be used as reference.
QUOTE_TO_PURCHASE_REQUISITION_CHAT_ASSISTANT_JJ04.json	To be used as reference.

Note to import the Agent Team.

We have imported the JSON file in another instance and we need to create previously our Business Object AA_CREATE_PURCHASE_REQ.

After this manual creation we could open the Agent Team and the tests were ok.