

# IBM Business Automation and Digital Labor Hands-on Labs

## IBM watsonx Orchestrate Decision

For IBM watsonx Orchestrate Tech Jam EMEA

Last update: Tuesday, 10 April 2025  
wxO 2025.03.27

## Table of Contents

<b>1 Introduction.....</b>	<b>3</b>
1.1 IBM watsonx Orchestrate .....	3
1.2 Lab Scenario.....	3
1.2.1 Computer Accessories Procurement Automation .....	3
1.2.2 Solution Architecture.....	3
1.3 Lab Overview .....	4
<b>2 Lab Setup Instructions .....</b>	<b>5</b>
2.1 Login to IBM watsonx Orchestrate .....	5
<b>3 Exercise: Create Decision.....</b>	<b>6</b>
3.1 Decision scenario overview .....	6
3.2 Create Project.....	6
3.3 Create Decision Model .....	7
3.4 Define Data.....	8
3.5 Create Decision Model .....	11
3.5.1 Add Input nodes .....	11
3.5.2 Define the default value for input nodes.....	13
3.5.3 Add decision nodes .....	15
3.5.4 Connect the input nodes and decision nodes.....	16
3.5.5 Define a decision table for the category approval threshold .....	17
3.5.6 Define a decision table for category yearly budget .....	19
3.5.7 Define Business Rules for the Purchase Decision Node.....	21
3.6 Test the Decision.....	30
3.6.1 Test the Auto Approved Decision Path.....	30
3.6.2 Test the Manager's Approval Required Decision Path.....	32
3.6.3 Test the Request Denied Decision Path .....	33
3.7 Publish the Decision Model as a Skill .....	34
3.8 Test the Skill in Chat.....	36
3.8.1 Add the Skill to the Personal Skills .....	36
3.8.2 Invoke the Skill in the Chat.....	38
3.9 Summary .....	39

# 1 Introduction

## 1.1 IBM watsonx Orchestrate

IBM watsonx Orchestrate (wxO) is a generative AI and automation solution designed to help businesses automate tasks, simplify complex processes, and save time and effort. It provides a catalog of prebuilt apps and skills, a conversational chat experience, and a low-code builder studio to create scalable AI assistants and agents.

To learn more about IBM watsonx Orchestrate, click [here](#). To see the technical documentation, click [here](#).

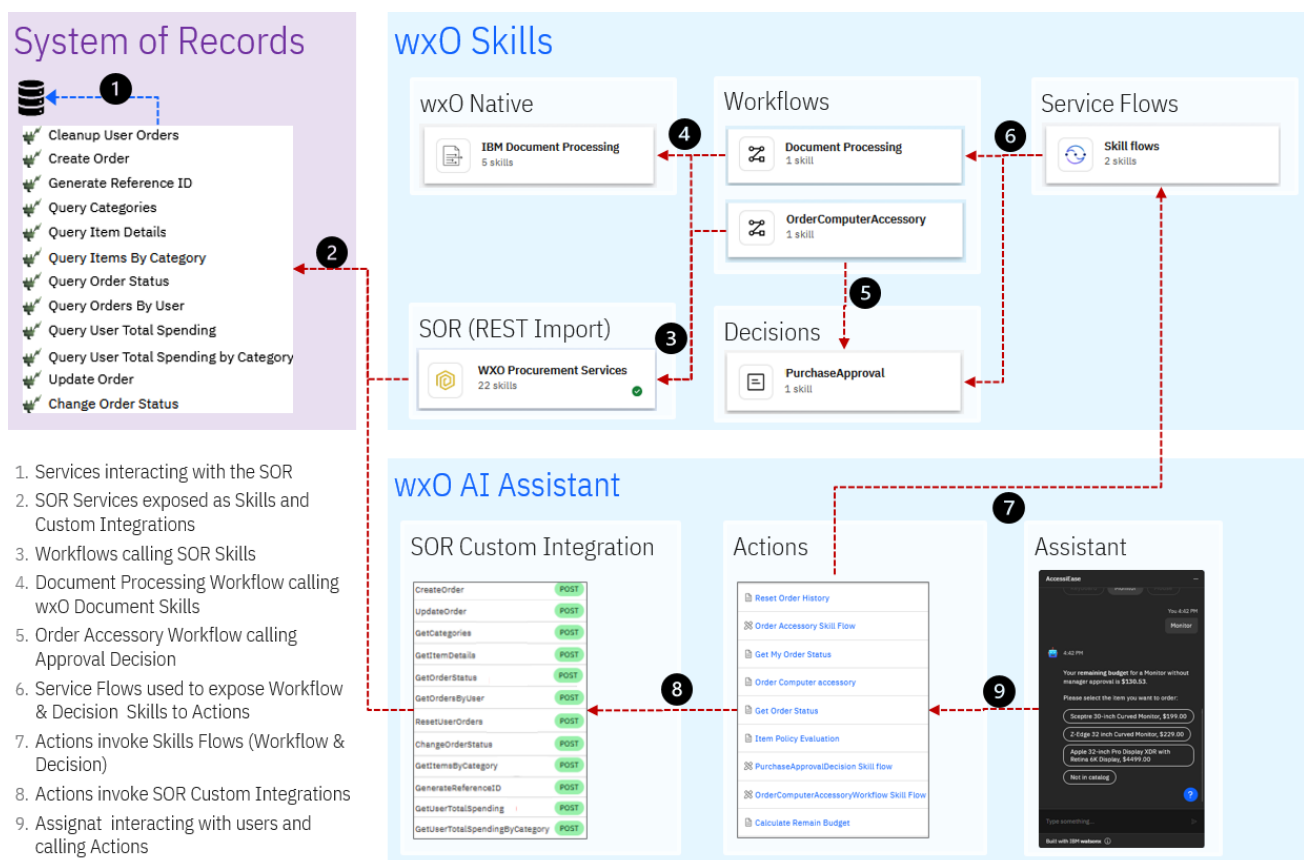
## 1.2 Lab Scenario

### 1.2.1 Computer Accessories Procurement Automation

Focus Corp provides computer accessories (e.g., mouse, keyboard, headsets, ...) for employees to order. The ordering system implemented using wxO's advanced AI-enhanced features provides a superior user experience and operational cost reductions.

- *Employees* use wxO AI Assistant. The AI Assistant guides users through the ordering process, provides advice on company policies, and helps them make informed decisions.
- *Managers* do not need to get involved with standard orders. They can focus on exception cases, making decisions based on the company's business needs and financial situation.
- *The purchasing department* utilizes an AI-assisted process to automate the procurement of catalog accessories, including the AI-assisted processing of quote documents.
- *The company* reduced operational costs by leveraging AI to automate the ordering process.

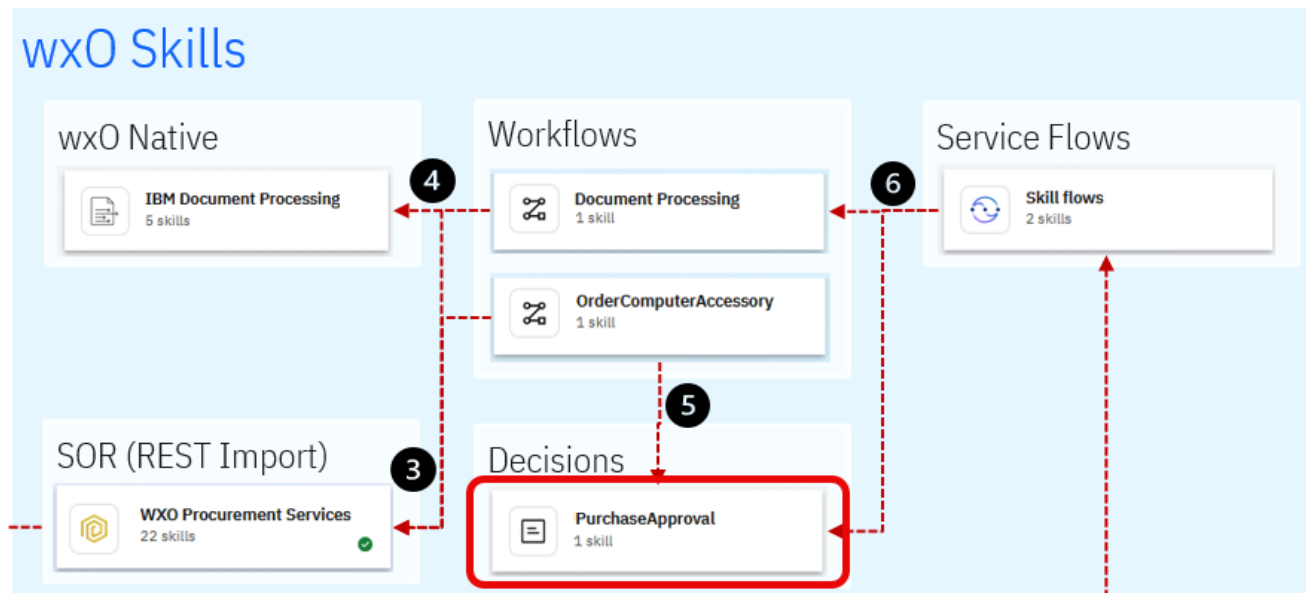
### 1.2.2 Solution Architecture



**Figure 1. Procurement of Computer Accessories Solution Architecture**

### 1.3 Lab Overview

In this lab, you will implement the Purchase Approval Decision Skill part of the Solution architecture introduced in the figure above.

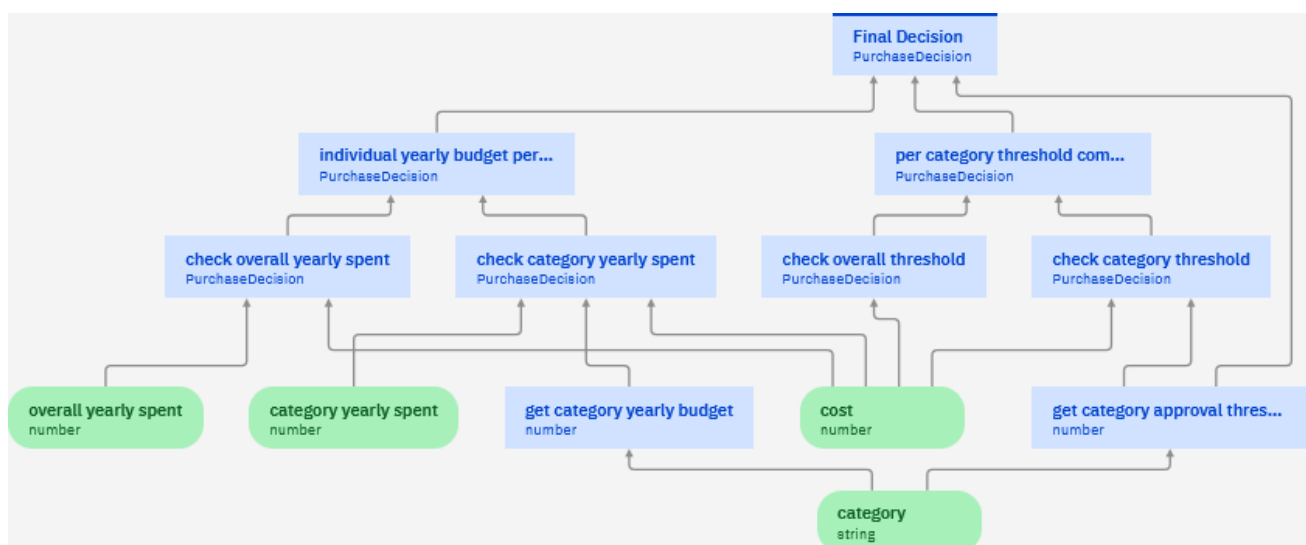


In this lab, you will learn how to create a Project that uses the decision feature of wxO.

Focus Corp's AI Assistant utilizes Purchase Approval Decision to determine whether a purchase request requires approval from managers. The decision will be based on the item cost and current yearly spending.

You will follow these steps to create a skill-based action:

- Create decision model and decision rules
- Preview the decision model
- Share and publish the decision project
- Test skill-based action



## 2 Lab Setup Instructions

### 2.1 Login to IBM watsonx Orchestrate

- \_1. In your web browser open IBM watsonx Orchestrate using the link provided to you.
- \_2. Enter your **IBM ID**, click **Continue**, enter your **password**, and click **Continue** again to log in.
- \_3. If you have access to multiple tenants, please select the tenant that was shared with you as part of the event.

Log in to IBM

Watson Orchestrate

Don't have an account? [Book a demo](#)

Continue with IBMid

[Forgot ID?](#)

pacholsk@ca.ibm.com

Continue

→

## 3 Exercise: Create Decision

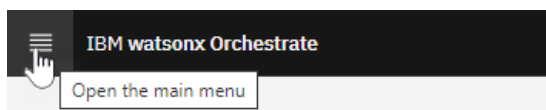
### 3.1 Decision scenario overview

You will be implementing five business rules that govern the ordering of computer accessories by Focus Corp's employees:

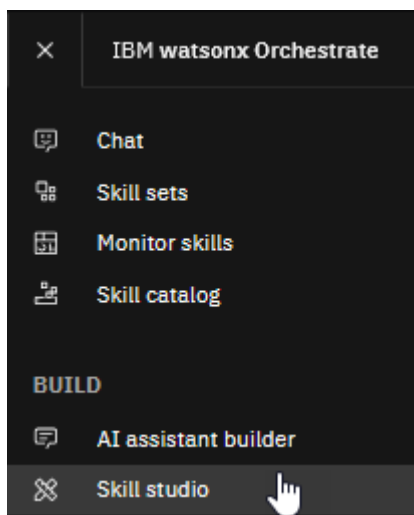
1. If the cost exceeds the limit for individual items, it will be denied
2. If the cost exceeds the limit for an individual category, it needs the manager's approval
3. If the cost exceeds the category's yearly budget, it needs the manager's approval
4. If the cost t exceeds the overall annual budget, it needs the manager's approval
5. Otherwise, the order will be approved automatically

### 3.2 Create Project

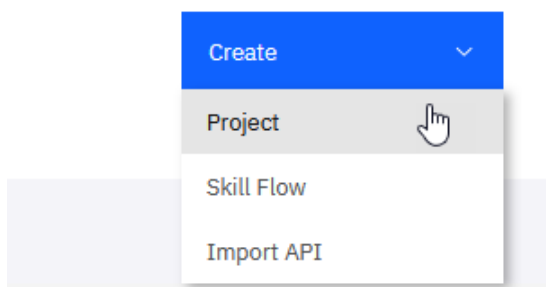
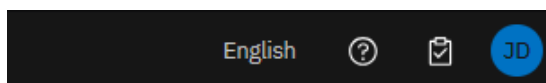
\_1. Click the **Hamburger** menu in the top-left corner.



\_2. Click **Skill studio**.



\_3. On the top-right, click **Create > Project**.



\_4. As *Name*, enter **UsrXXXPurchaseApprovalDecision** (remember to replace XXX with your back end Credentials user id) and click **Create**.

**New project**

- Create project
- Import project
- Discovery tutorials
- Industry samples

**Create project**  
Create an empty project and build it from scratch.

Name  
UsrXXXPurchaseApprovalDecision

Note: This is a symbolic name that must be unique and cannot be changed later.

Description (optional)  
Describe your project

Cancel Create

### 3.3 Create Decision Model

\_1. Click the **Decision** tile.

Decision

Automate complex business decisions with rules and decision tables.

→

\_2. Then click the **Decision Model** tile.

Decision model

Decompose your decision and define the data its depends on.

→

\_3. For *Name*, enter **UsrXXX Purchase Approval Model** (remember to replace XXX with your back end Credentials user id) and click **Create**

Create a decision model

Name

UsrXXX Purchase Approval Model

Note: This is a symbolic name that must be unique and cannot be changed later.

Description (optional)

Describe your skill

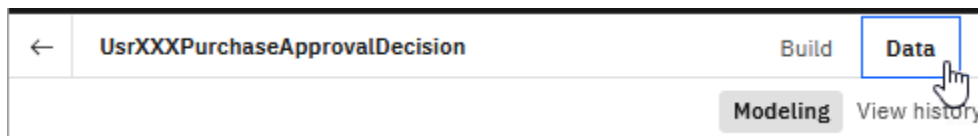
Cancel Create

### 3.4 Define Data

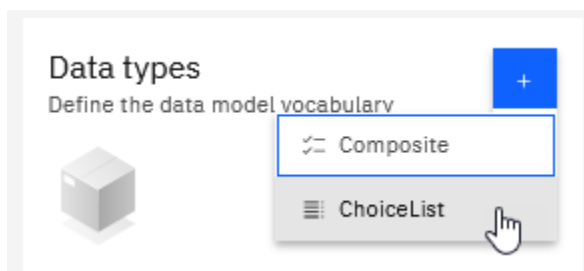
In this scenario, the decision will return the following information:

- **Approval:** can be "auto", "manager", or "denied"
- **PurchaseDetail:** contains detailed information, e.g., item cost, approval threshold, etc
- **Message:** Your purchase request has been approved or denied or needs the manager's approval

\_1. Click the **Data** tab at the top center to switch to the Data view.



\_2. On the left side, click the **blue plus button** , then select **ChoiceList**.



\_3. Set the name of the ChoiceList to **Approval**.

Approval

ChoiceList



**Note:** The Vocabulary will be created once you click outside of the ChoiceList.

### Vocabulary

Edit the default verbalization of vocabulary elements.

#### Singular

**the** Approval      **an** Approval      **this** Approval      **each** Approval

#### Plural

**the** Approvals      Approvals      **all** Approvals      **Approvals**

\_4. In the *Values* section, click the blue **Add +** button **twice** to add two more choices.

### Values

Define the list of possible values for the choice list. For example, Season can have the values Spring, Summer, Autumn, and Winter.

Add +

\_5. Modify the three choice items to **Manager**, **Denied**, and **Auto**, as shown below.

### Values

Define the list of possible values for the choice list. For example, Season can have the values Spring, Summer, Autumn, and Winter.

Add +

#### Name

Manager

Denied

Auto

\_6. On the left side, click the **blue plus button** , then select **Composite**.

### Data types

Define the data model vocabulary

Search

Approval

Composite

ChoiceList

\_7. Set Composite to **PurchaseDetail**.

PurchaseDetail

Composite

\_8. In the *Attributes* section, **click the blue Add + eight times** to create eight attributes.

### Attributes


Create a list of attributes to describe the structure of the data type. For example a Customer has a name, an address, and a date of birth.

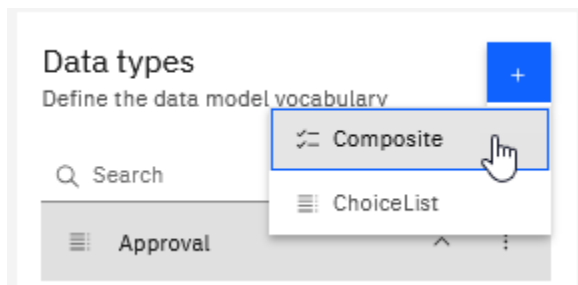
Add +

\_9. Set the names and types of these attributes as follows:

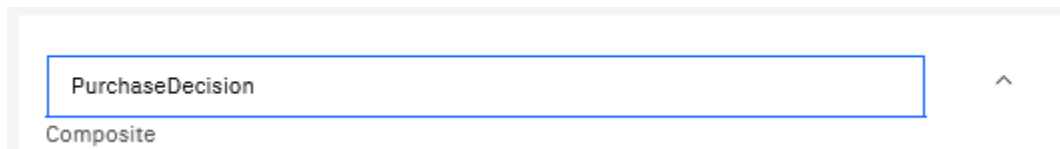
- ApprovalThreshold: Number
- CategoryApprovalThreshold: Number
- CategoryYearlyBudget: Number
- CategoryYearlySpent: Number
- ItemCategory: String
- ItemCost: Number
- OverallYearlyBudget: Number
- OverallYearlySpent: Number

Name	Type	List
ApprovalThreshold	Number	<input type="checkbox"/> ⋮
<i>the ApprovalThreshold, the ApprovalThresholds, an ApprovalThreshold, ApprovalThresholds</i>		
CategoryApprovalThreshold	Number	<input type="checkbox"/> ⋮
<i>the CategoryApprovalThreshold, the CategoryApprovalThresholds, a CategoryApprovalThreshold, CategoryApprovalThresholds</i>		
CategoryYearlyBudget	Number	<input type="checkbox"/> ⋮
<i>the CategoryYearlyBudget, the CategoryYearlyBudgets, a CategoryYearlyBudget, CategoryYearlyBudgets</i>		
CategoryYearlySpent	Number	<input type="checkbox"/> ⋮
<i>the CategoryYearlySpent, the CategoryYearlySpents, a CategoryYearlySpent, CategoryYearlySpents</i>		
ItemCategory	String	<input type="checkbox"/> ⋮
<i>the ItemCategory, the ItemCategories, an ItemCategory, ItemCategories</i>		
ItemCost	Number	<input type="checkbox"/> ⋮
<i>the ItemCost, the ItemCosts, an ItemCost, ItemCosts</i>		
OverallYearlyBudget	Number	<input type="checkbox"/> ⋮
<i>the OverallYearlyBudget, the OverallYearlyBudgets, an OverallYearlyBudget, OverallYearlyBudgets</i>		
OverallYearlySpent	Number	<input type="checkbox"/> ⋮
<i>the OverallYearlySpent, the OverallYearlySpents, an OverallYearlySpent, OverallYearlySpents</i>		

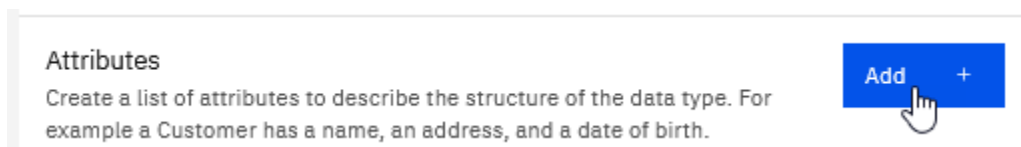
\_10. On the left side, click the blue plus button , then select **Composite**.



\_11. Set Composite to **PurchaseDecision**.



\_12. In the *Attributes* section, **click the blue Add + three times** to create three attributes.



\_13. Add three attributes to it:

- Approval: Approval
- Details: PurchaseDetail
- Message: String

Name	Type	List
Approval	Approval	<input type="checkbox"/> 
<i>the Approval, the Approvals, an Approval, Approvals</i>		
Details	PurchaseDetail	<input type="checkbox"/> 
<i>the Details, the Detailses, a Details, Detailses</i>		
Message	String	<input type="checkbox"/> 
<i>the Message, the Messages, a Message, Messages</i>		

**Note:** The Decision Editor saves the changes automatically as you type.

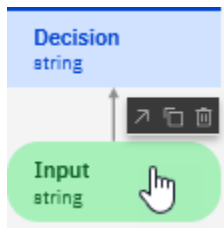
## 3.5 Create Decision Model

### 3.5.1 Add Input nodes

\_1. Switch to the **Build** view by clicking the **Build** tab on the top center.



\_2. Select the **Input** node.



\_3. Change the node name to **cost** and the output type to **Number**.

cost  
Input node

Details Default value

Node details

Node name

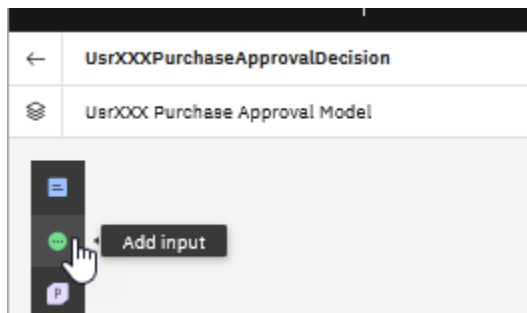
cost

Output type

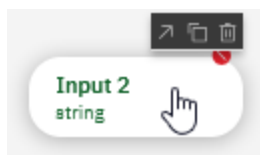
Number

A number, that can be decimal

\_5. Add a new input node by clicking the **Add input** icon  in the diagram toolbar.



\_6. Select the created input node **Input 2**.



\_7. On the property panel on the right side, change the name to **category** and leave the Output type as **String**.

category  
Input node

Details Default value

Node details

Node name

category

Output type

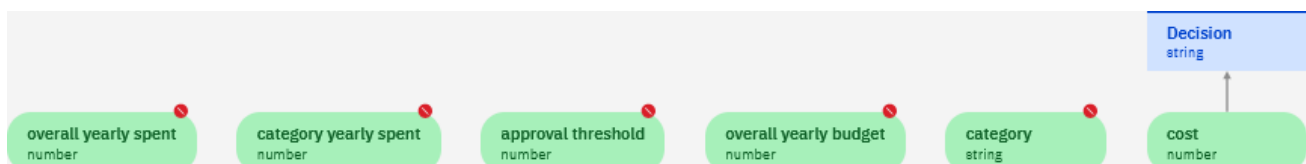
String

A string of characters

\_8. Follow steps 6-8 to add four input nodes according to the table below:

Node name	Output type
overall yearly spent	Number
category yearly spent	Number
approval threshold	Number
overall yearly budget	Number

\_9. Verify that the diagram looks as shown below:

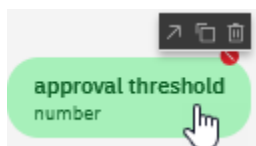


### 3.5.2 Define the default value for input nodes

In this scenario, **cost**, **category**, **overall yearly spent**, and **category yearly spent** will be provided as input by the service calling the decision model. We won't define default values for them.

We will define default values for **approval threshold** and **overall yearly budget** here.

\_1. Click the **approval threshold** input node.



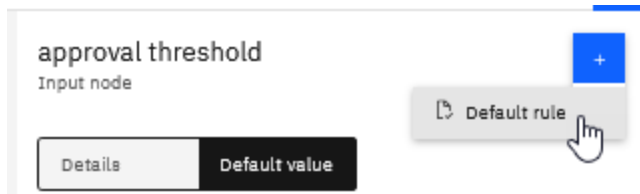
\_2. On the property panel, click **Default value** tab.

approval threshold  
Input node


Details Default value

Node details

\_10. Click the blue **plus button**  and select **Default rule**.



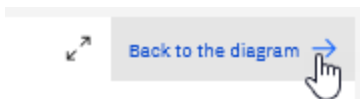
\_11. In the default-value-setting editor, change *<a number>* to **1000**.

default-value-setting 

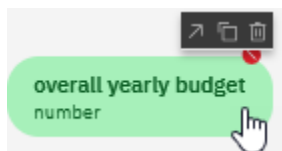
Type your rule using the list below as reference

```
1 set 'approval threshold' to 1000 ;
```

\_12. Click **Back to the diagram**  to go back to the model diagram.



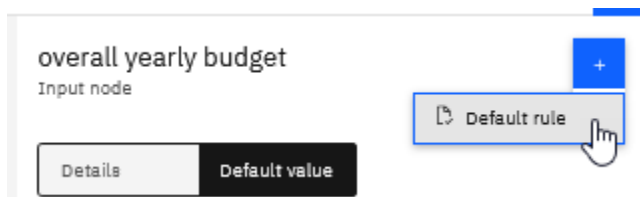
\_3. Click the **overall yearly budget** input node.



\_4. On the property panel, if not already selected, click **Default value** tab.



\_13. Click the blue **plus button**  and select **Default rule**.




\_14. In the default-value-setting editor, change *<a number>* to **330**.

default-value-setting 

Type your rule using the list below as reference

```
1 set 'overall yearly budget' to 330;
```

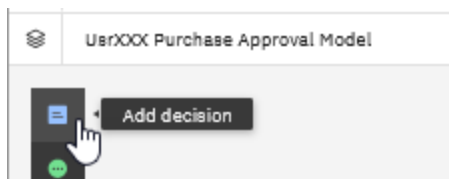
\_15. Click **Back to the diagram**  to go back to the model diagram.



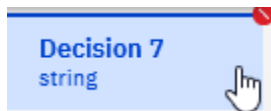
### 3.5.3 Add decision nodes

We will create two decision nodes: **category yearly budget**, and **category approval threshold**. They will use the **category** as input.

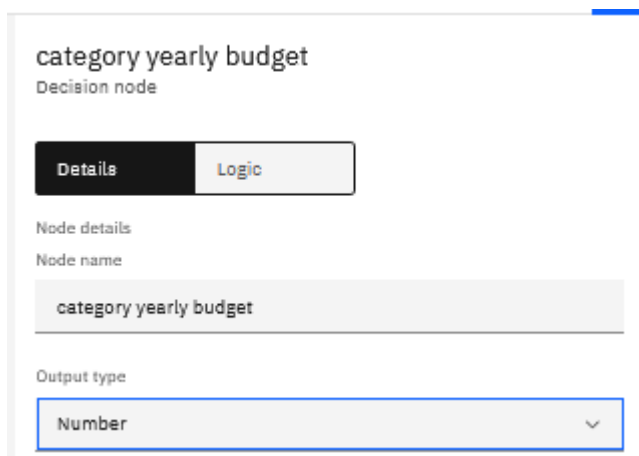
\_1. Add a decision node by clicking the **Add decision** button  in the diagram toolbar.



\_2. Select the decision node just created,

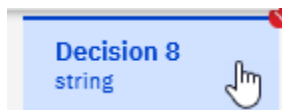


\_3. Change the *Node name* to **category yearly budget** and the *Output type* to **Number**.


 A screenshot of the IBM Business Automation console showing the configuration for a decision node. The node is named 'category yearly budget' and is of type 'Decision node'. There are two tabs: 'Details' (selected) and 'Logic'. Under 'Node details', the 'Node name' field contains 'category yearly budget'. The 'Output type' dropdown menu is set to 'Number'.

\_4. Add another decision node.

\_5. Click the **Decision** node just created.



\_6. Change the *Node name* to **category approval threshold** and the *Output type* to **Number**.

category approval threshold

Decision node

Details	Logic
---------	-------

Node details

Node name

category approval threshold

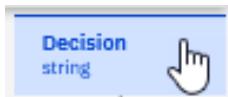
Output type

Number

A number, that can be decimal

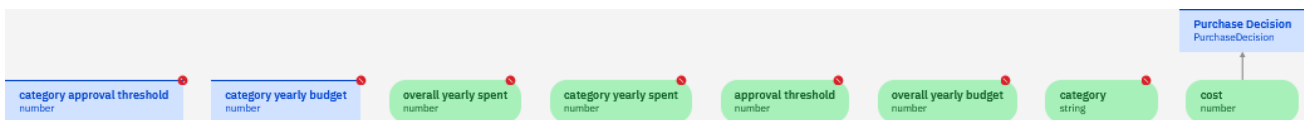
☐ Output is a list

\_7. Click the (initial) **Decision** node created by default and linked to the cost input.



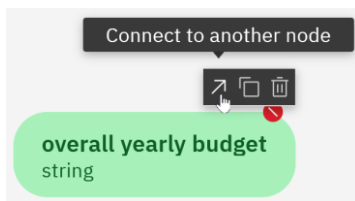
\_8. Change the *Node name* to **Purchase Decision** and the *Output type* to **PurchaseDecision**.

\_9. Verify that the Diagram looks as shown below.



### 3.5.4 Connect the input nodes and decision nodes

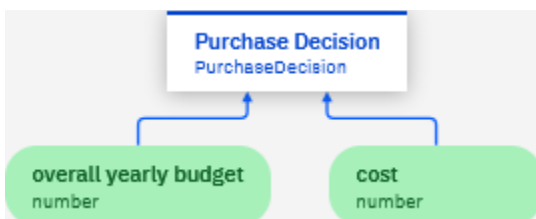
\_1. Hover the mouse over the *overall yearly budget* node and click the **Connect to another node** icon (arrow).



\_2. **Move the arrow** to the *Purchase Decision* node and click to make a permanent connection.



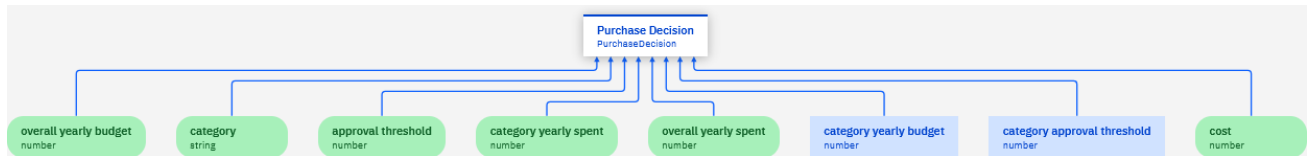
\_3. Verify that the Diagram now looks as shown below:



\_4. Repeat steps 1-3 to connect the **remaining input nodes** to the *Purchase Decision* node.

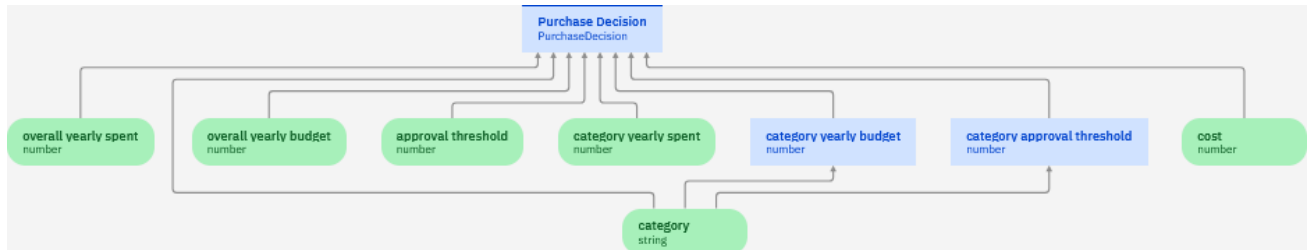


\_5. Verify that the Diagram now looks as shown below:



\_6. Repeat steps 1-2 to connect the **category** node to the **category yearly budget** and the **category approval threshold** Decision nodes.

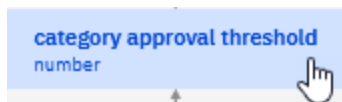
\_7. Verify that the Diagram now looks similar to the one shown below:



**Note:** The diagram may look different depending on the order in which you made the connection.

### 3.5.5 Define a decision table for the category approval threshold

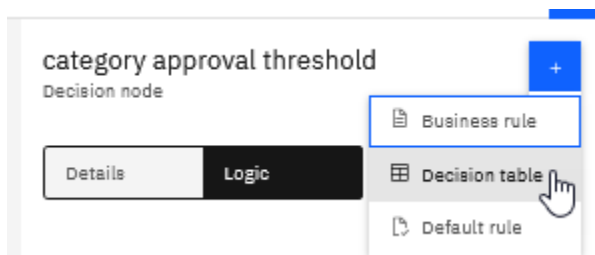
\_1. Select the **category approval threshold** Decision node.



\_2. Click the **Logic** tab.



\_3. Click the + button and select **Decision table**.




\_4. Set the *Name* to **category approval threshold**, select the **category** checkbox to include it in the table, and click the **Create** button.

category approval threshold ×


### Create decision table

Name

category approval threshold 

Select the condition columns for your decision table

Q


category 

☒ 'category' string

Preview your decision table


You can drag column headers to change the order of the condition columns.



	category	category approval threshold
1	abc	
2		
3		

Cancel Create 

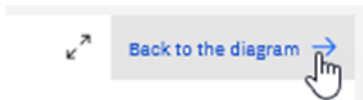
\_5. Define decision table cell contents by double-clicking the cell and setting the values from the table below:

category	category approval threshold
mouse	60
keyboard	40
monitor	200

category approval threshold 

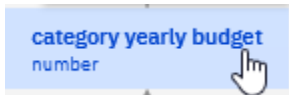
	category 	category approval threshold 
1	mouse	60
2	keybord	40
3	monitor	200
4		

\_16. Click **Back to the diagram** → to go back to the model diagram.



### 3.5.6 Define a decision table for category yearly budget

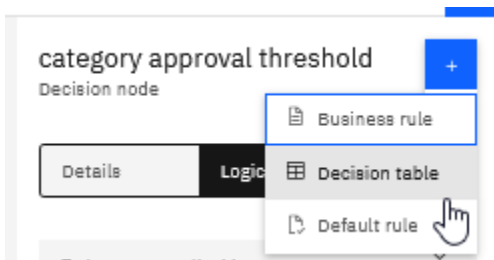
\_1. Select the **category yearly budget** Decision node.



\_2. If not already selected, click the **Logic** tab.



\_3. Click the + button and select **Decision table**.



4. Set the *Name* to **category yearly budget**, select the **category** checkbox to include it in the table, and click the **Create** button.

category approval threshold

### Create decision table

Name

category yearly budget

Select the condition columns for your decision table

category

☒ 'category' string

Preview your decision table

You can drag column headers to change the order of the condition columns.

	category	category approval threshold
1	abc	
2		
3		

Cancel Create

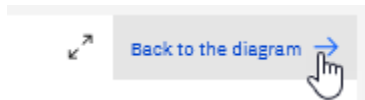
5. Define decision table cell contents by double-clicking the cell and setting the values from the table below:

category	category approval threshold
mouse	100
keyboard	100
monitor	300

category yearly budget

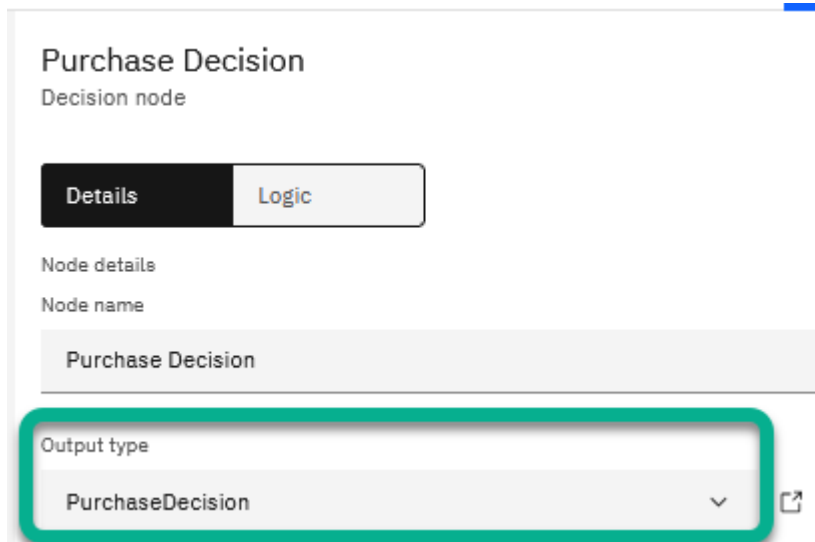
	category	category approval threshold
1	mouse	100
2	keyboard	100
3	monitor	300
4		

\_17. Click **Back to the diagram** → to go back to the model diagram.

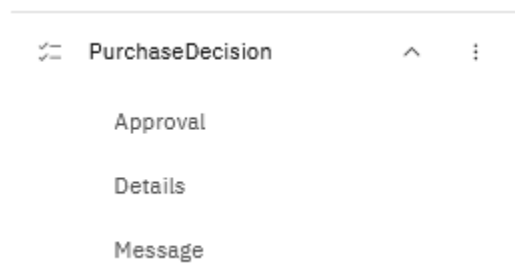


### 3.5.7 Define Business Rules for the Purchase Decision Node

Recall that the data type PurchaseDecision defines the output of the Purchase Decision decision.



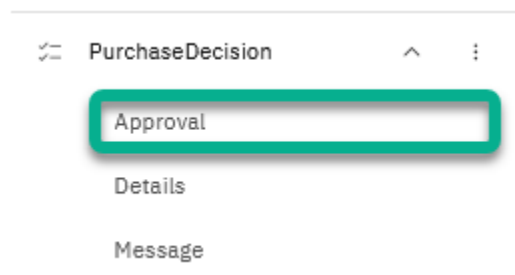
Note that this data type comprises three elements: Approval, Details, and Message.



In the steps below you will create business logic to define the three output elements of the **Purchase Decision** decision.

#### 3.5.7.1 Define Four Auto Approval Business Rules

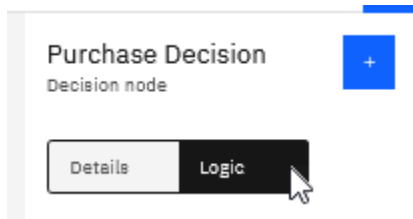
You will now author the business logic to define the Auto Approval part of the Approval output.



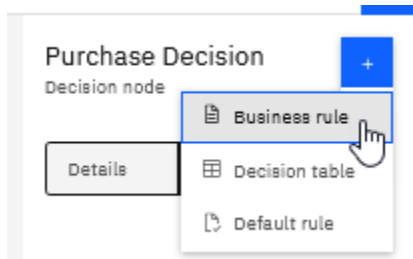
\_1. Select the **Purchase Decision** Decision node.



\_2. If not already selected, click the **Logic** tab.



\_3. Click the + button and select **Business rule**.



\_4. Set the *Name* to **Overall threshold auto approval** and click **Create**.

 A screenshot of the 'Create business rule' dialog box. The title is 'Purchase Decision' and 'Create business rule'. The 'Name' field contains the text 'Overall threshold auto approval'. Below this, there is a section 'Select the criteria for your rule' with a search bar and three expandable sections: 'cost', 'category', and 'overall yearly budget'. Each section has a checkbox and a data type (number, string, and overall yearly budget respectively). At the bottom, there is a 'Preview your rule' section showing a code snippet:
 

```
set decision to a new PurchaseDecision where
  the Approval is <Approval> ,
  the Details is <Details> ,
  the Message is <Message> ;
```

 At the very bottom, there are two buttons: 'Cancel' and 'Create'. A mouse cursor is pointing at the 'Create' button.

\_5. **Replace** the generated rule with the expression below.

```
if
  cost ≤ 'approval threshold'
then
  set the Approval of decision to Auto;
```

The business rule logic states that the purchase request will be automatically approved if the item cost does not exceed the approval threshold.

#### Overall threshold auto approval

Type your rule using the list below as reference

```
1 if
2   cost ≤ 'approval threshold'
3 then
4   set the Approval of decision to Auto;
```

**Note:** You can also create the rule manually. Type **if** and press space. The smart editor will list the possible values / actions.

#### Overall threshold auto approval

Type your rule using the list below as reference

```
1 if
# false
# true
'approval threshold'
'category approval threshold'
'category yearly budget'
'category yearly spent'
'overall yearly budget'
'overall yearly spent'
category
cost
decision
lg ( <a number> )
lg ( <a number> )
```

Variable:  
category yearly budget: number

smart mode

Select **cost** from the list and wait for one second. The business rule editor will list the possible actions.

#### Overall threshold auto approval

Type your rule using the list below as reference

```
1 if cost
  < <a number>
  = <a number>
  ≠ <a number>
  > <a number>
  ≤ <a number>
  ≥ <a number>
  does not equal <a number>
  equals <a number>
  is <an object>
  is at least <a number>
  is at least <min> and less than <max>
  is at most <a number>
```

Phrase:  
<a number> < <a number>; boolean

Documentation:  
Returns whether the first numeric value is smaller than the second

smart mode

\_6. Follow steps 3-5 to add a new Business rule, set the *Name* to **Category threshold auto approval**, and replace the default Business rule logic with the following:

```
if
  cost ≤ 'category approval threshold'
then
  set the Approval of decision to Auto;
```

The business rule logic states that the purchase request will be automatically approved if the item cost does not exceed the category approval threshold.

\_7. Follow steps 3-5 to add a new Business rule, set the *Name* to **Overall yearly spent auto approval**, and replace the default Business rule logic with the following:

```
if
  cost + 'overall yearly spent' ≤ 'overall yearly budget'
then
  set the Approval of decision to Auto;
```

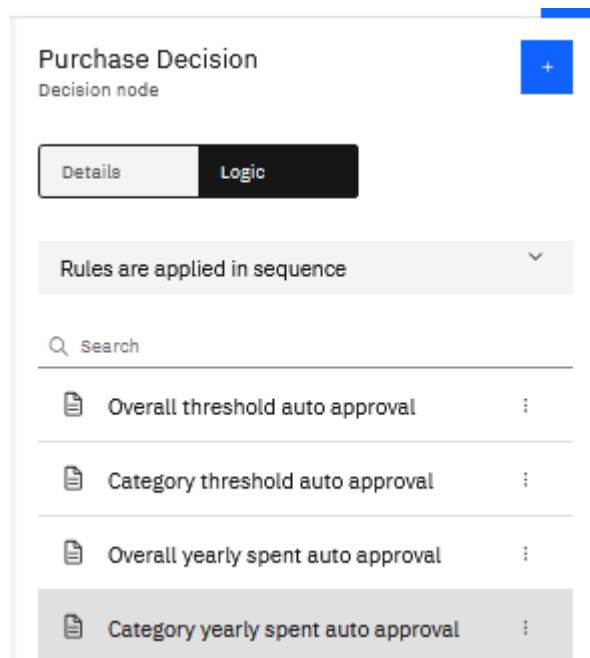
The business rule logic states that the purchase request will be automatically approved if the item cost plus the annual amount spent does not exceed the employee's yearly budget.

\_8. Follow steps 3-5 to add a new Business rule, set the *Name* to **Category yearly spent auto approval**, and replace the default Business rule logic with the following:

```
if
  cost + 'category yearly spent' ≤ category yearly budget'
then
  set the Approval of decision to Auto;
```

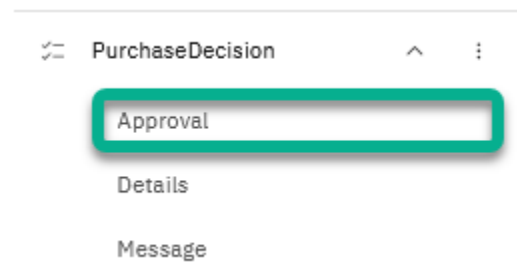
The business rule logic states that the purchase request will be automatically approved if the item cost plus the annual amount spent does not exceed the employee's yearly budget.

\_9. Verify that you have four Business rules defined.



### 3.5.7.2 Define Three Manager Approval Business Rules

You will now author the business logic to define the Manager Approval part of the Approval output.



In each step, follow steps 3-5 in [3.5.7.1 Define Auto Approval Business Rules](#).



\_1. Add a new Business rule, set the *Name* to **Category threshold manager approval**, and replace the default Business rule logic with the following:

```
if
  cost > 'category approval threshold'
then
  set the Approval of decision to Manager;
```

The business rule logic states that the purchase request will need manager approval if the item cost exceeds the category approval threshold.

\_2. Add a new Business rule, set the *Name* to **Category yearly spent manager approval**, and replace the default Business rule logic with the following:

```
if
  (cost + 'category yearly spent') > 'category yearly budget'
then
  set the Approval of decision to Manager;
```

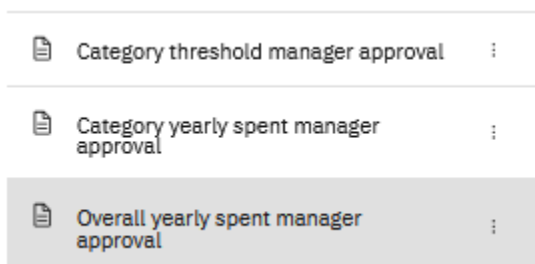
The business rule logic states that the purchase request will require manager approval if the item cost plus the annual amount spent exceeds the employee's category yearly budget.

\_3. Add a new Business rule, set the *Name* to **Overall yearly spent manager approval**, and replace the default Business rule logic with the following:

```
if
  (cost + 'overall yearly spent') > 'overall yearly budget'
then
  set the Approval of decision to Manager;
```

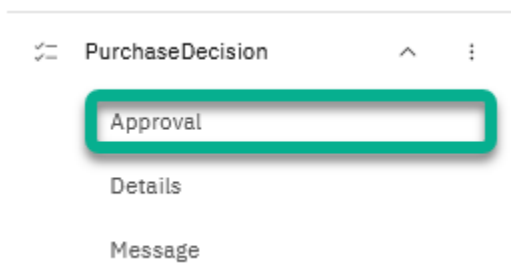
The business rule logic states that the purchase request will require manager approval if the item cost plus the annual amount spent exceeds the employee's yearly budget.

\_4. Verify that you have three more Business rules.



### 3.5.7.3 Define a Denied Business Rule

You will now author the business logic to define the Denied part of the Approval output.

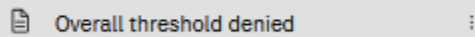


\_1. Follow steps 3-5 in [3.5.7.1 Define Auto Approval Business Rules](#) to add a new Business Rule. Set the *Name* to **Overall threshold denied**, and replace the default Business rule logic with the following code snippet:

```
if
  cost > 'approval threshold'
then
  set the Approval of decision to Denied;
```

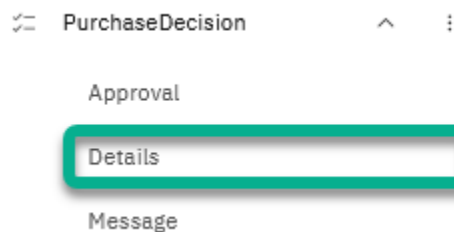
The business rule logic states that the purchase request will be denied if the item's cost exceeds the pre-defined threshold.

\_2. Verify that you have one more Business rule.



### 3.5.7.4 Define Business Rule to Set the Decision Details

You will now author the business logic to define the decision Details output.

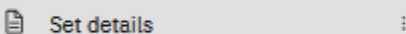


\_1. Follow steps 3-5 in [3.5.7.1 Define Auto Approval Business Rules](#) to add a new Business Rule. Set the *Name* to **Set details**, and replace the default Business rule logic with the following code snippet:

```
set the Details of decision to a new PurchaseDetail where
  the ItemCategory is 'category',
  the ItemCost is cost,
  the CategoryApprovalThreshold is 'category approval threshold',
  the ApprovalThreshold is 'approval threshold',
  the CategoryYearlySpent is 'category yearly spent',
  the CategoryYearlyBudget is 'category yearly budget',
  the OverallYearlySpent is 'overall yearly spent',
  the OverallYearlyBudget is 'overall yearly budget';
```

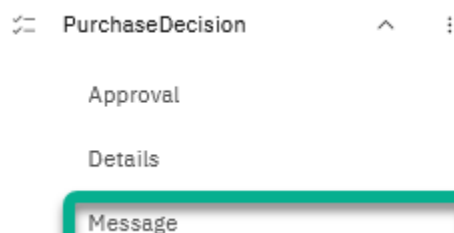
This logic will set the decision details in the decision output.

\_2. Verify that you have one more Business rule.



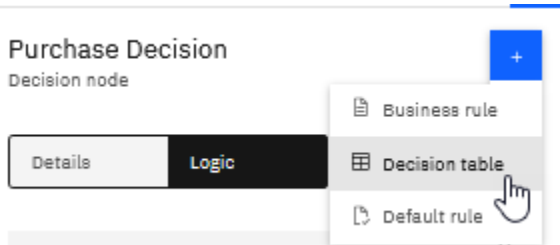
### 3.5.7.5 Define the Decision Table to Set the Decision Message

You will now author the business logic to define the decision Message output.



Specifically, you will now define a decision table to set a message visible to the user requesting a computer accessory. You associate the variable Approval with the table's first (input) column. You associate the variable Message with the table's second (output) column. The table will enable us to set the decision message based on approval type. For example: "When Approval Type is xxx, set the Message to yyy".

\_1. Click the **+** button and select **Decision table**.



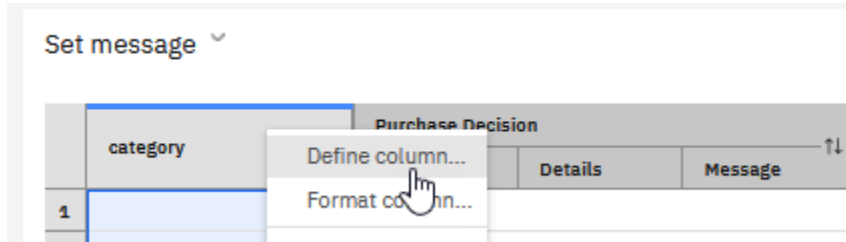
\_2. Set the *Name* to **Set message**, select the **category** checkbox to include it in the table, and click the **Create** button.

The screenshot shows the 'Create decision table' dialog box. The 'Name' field is set to 'Set message' and has a green checkmark. Below it, there's a section 'Select the condition columns for your decision table' with a search bar and a list of columns: 'cost', 'category', and 'overall yearly budget'. The 'category' column is selected with a checkbox and has a green checkmark. The 'cost' column is not selected. The 'overall yearly budget' column is not selected. Below this, there's a 'Preview your decision table' section with a table. The table has two columns: 'category' and 'Purchase Decision'. The 'category' column has three rows with values 'abc', '1', and '2'. The 'Purchase Decision' column is empty. At the bottom, there are 'Cancel' and 'Create' buttons. A mouse cursor is pointing at the 'Create' button.

	category	Purchase Decision
1	abc	
2		
3		

**Note:** We don't need the category variable in the table. We must select it to get a condition column.

\_3. Right-click on the **category** column header and select **Define column**.



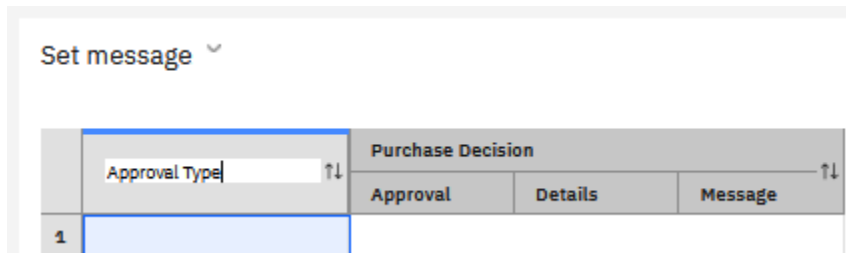
\_4. Set the **Condition Definition** to the code snippet below and then click OK

the Approval of decision is one of <Approvals>



This code snippet will change the first column to use the Approval type.

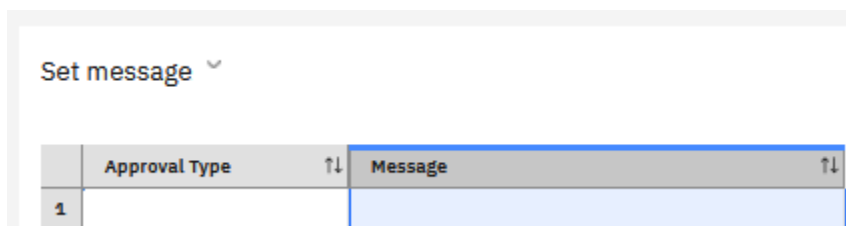
\_5. Click the first column title and modify its name to **Approval Type**.



\_6. Use steps 2-4 to modify the second column's name to **Message** and set its Action Definition to the following:

set the Message of decision to <a string>

\_7. Verify that the columns look as shown below.



\_8. Double-click the first cell in the **Approval Type** column, a drop-down list will appear; select **Auto**.

Set message ▾

	Approval Type ↑↓	Message ↑↓
1	Auto	
2	Filter domain values	
3	<input checked="" type="checkbox"/> Auto <input type="checkbox"/> Denied <input type="checkbox"/> Manager	
4		

\_9. Double-click the first cell in the **Message** column and enter: **Your purchase request has been approved automatically.**

	Approval Type ↑↓	Message ↑↓
1	Auto	Your purchase request has been approved...
2		

\_10. Repeat steps 7-8 to complete the table. Use the data in the table below.

Approval Type	Message
Denied	Your purchase request has been denied.
Manager	Your purchase request would need the manager's approval.

	Approval Type ↑↓	Message ↑↓
1	Auto	Your purchase request has been approved automatically.
2	Denied	Your purchase request has been denied.
3	Manager	Your purchase request would need the manager's approval.

\_11. Verify that you have a Decision Table.

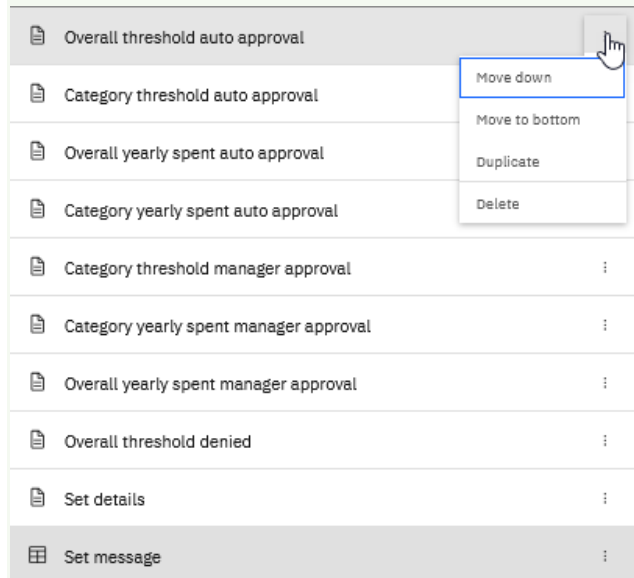
 Set message

Note that the 10 rules you defined in the Purchase Decision node will execute in the sequence you have added them: top to bottom.



In our case, this is the correct sequence.


You can change the execution order using the popup menu options on each rule.

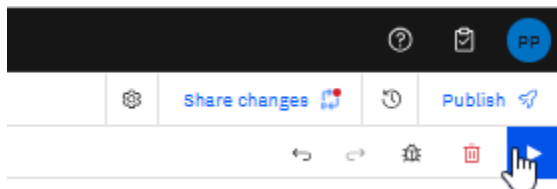


### 3.6 Test the Decision

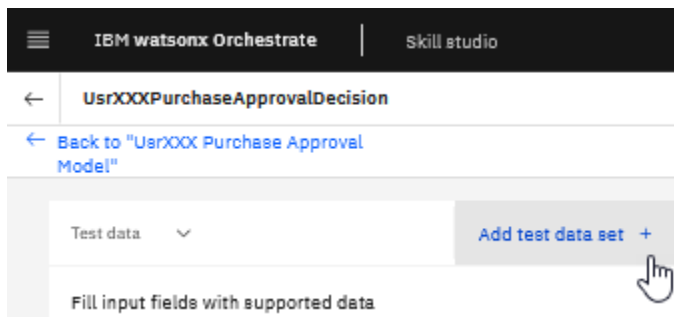
You can test the Decision you have just authored using the Preview capability.

#### 3.6.1 Test the Auto Approved Decision Path

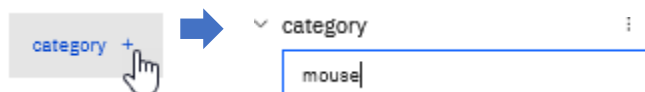
\_1. Click the **Preview** button  on the top right.



\_2. Click **Add test data set +**.



\_3. For each Field defined in the table below, click the + to expand it and enter the data shown in the table below.



Field	Value
category	mouse
categoryYearlySpent	30
cost	29
overallYearlySpent	150

\_4. Verify that the dataset looks as shown below:

Test data **dataset** ✎ 📄 🗑️ [Add test data set +](#)

**dataset** 📄 </>

Fill input fields with supported data ℹ️

category  
 mouse

categoryYearlySpent  
 30

cost  
 29

overallYearlySpent  
 150

\_5. Click the **Preview** button.

? 📄 PP

⚙️ [Share changes](#) 🕒 [Publish](#)

[Preview](#)

\_6. Click **Show more** to see the full JSON output of the decision.

**dataset** [Show JSON output](#)

2/22/2025, 3:26:58 PM

^ Decision output

Node Name	Result
Purchase Decision	<pre>{   "Approval": "Auto",   "Details": {     "ApprovalThreshold": 1000.0,     "CategoryApprovalT</pre>

[Show more](#)

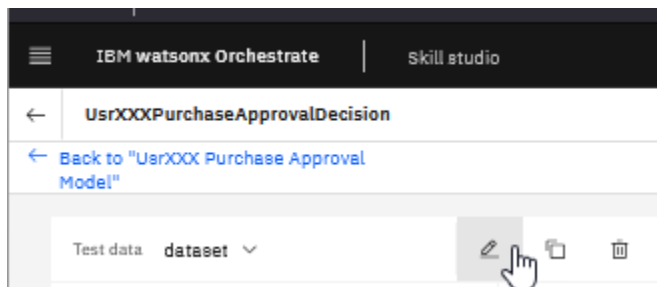
\_7. Verify that the output looks as shown below.

Result

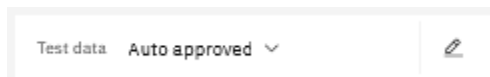
```
{
  "Approval": "Auto",
  "Details": {
    "ApprovalThreshold": 1000,
    "CategoryApprovalThreshold": 60,
    "CategoryYearlyBudget": 100,
    "CategoryYearlySpent": 30,
    "ItemCategory": "mouse",
    "ItemCost": 29,
    "OverallYearlyBudget": 330,
    "OverallYearlySpent": 150
  },
  "Message": "Your purchase request
has been approved automatically."
}
```

### 3.6.2 Test the Manager's Approval Required Decision Path

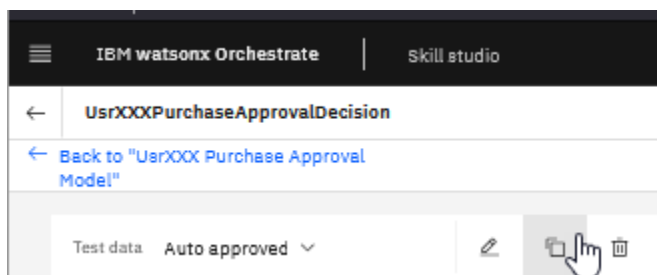
\_1. Click the **Rename test dataset** icon  .




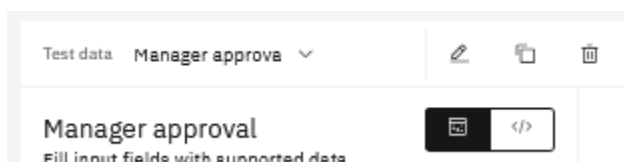
\_2. Set the *Test data* to **Auto approved**.



\_3. Click the **Duplicate test data set** button  .



\_4. Click the **Rename test dataset** icon  to rename the dataset from *Auto approved*\_copy to **Manager approval**.



\_5. Change the **cost** to 100





\_6. Click the **Preview** button.



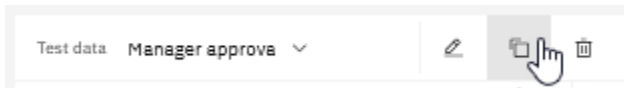
\_7. Examine the Result.


```
{
  "Approval": "Manager",
  "Details": {
    "ApprovalThreshold": 1000,
    "CategoryApprovalThreshold": 60,
    "CategoryYearlyBudget": 100,
    "CategoryYearlySpent": 30,
    "ItemCategory": "mouse",
    "ItemCost": 100,
    "OverallYearlyBudget": 330,
    "OverallYearlySpent": 150
  },
  "Message": "Your purchase request would need the manager's approval."
}
```

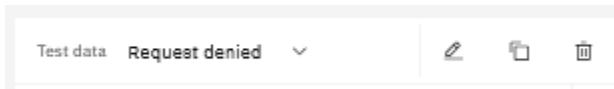
The manager's approval is required because the item cost plus the yearly spent for the mouse is 130, which exceeds the category annual budget of 100.

### 3.6.3 Test the Request Denied Decision Path

\_1. Click the **Duplicate test data set** button  .



\_2. Click the **Rename test dataset** icon  and rename the data set from *Manager approval\_copy* to **Request denied**.



\_3. Change the *category* to **monitor** and the *cost* to **1200**.

\_4. Click the **Preview** button.



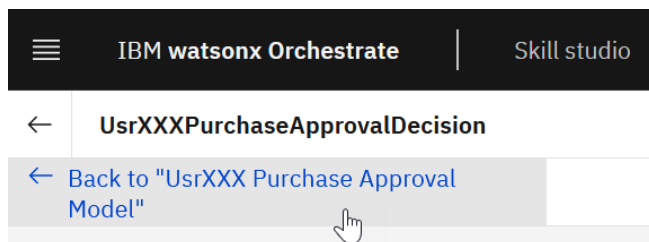
\_5. Examine the Result.

## Result

```
{
  "Approval": "Denied",
  "Details": {
    "ApprovalThreshold": 1000,
    "CategoryApprovalThreshold": 200,
    "CategoryYearlyBudget": 300,
    "CategoryYearlySpent": 200,
    "ItemCategory": "monitor",
    "ItemCost": 1200,
    "OverallYearlyBudget": 330,
    "OverallYearlySpent": 700
  },
  "Message": "Your purchase request has been denied."
}
```

The request is denied because the item cost is 1200, exceeding the approval threshold of 1000.

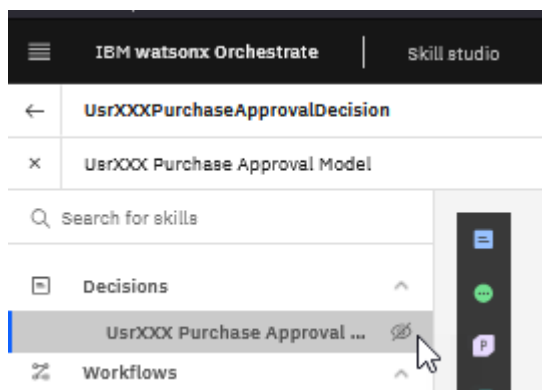
\_6. Click on **Back to UstrXXX Purchase Approval Model** to exit the test capability.



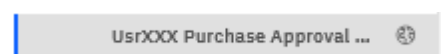
### 3.7 Publish the Decision Model as a Skill

Now that we have tested the decision model, we can publish it as a skill. The decision model is set to **private** by default. Before we publish it, we need to change it for the **public**.

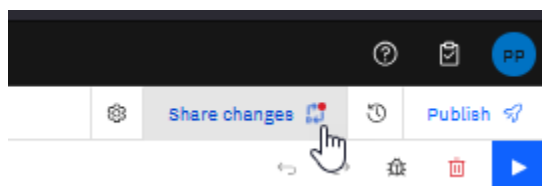
\_1. Click the **crossed eye icon** located to the right of our decision model.



\_2. Verify that the icon now shows public skill access.



\_3. Click **Share changes**.



\_4. Review the change log and click the **Share** button.

Change log

Load **Share**

Share changes  
Share changes that you made locally with your collaborators.

Revert changes **Share**

<input checked="" type="checkbox"/>	Skill name	Type	Kind	Details	Last updated
<input checked="" type="checkbox"/>	UsrXXX Purchase Approval Model	Decision model	Skill updated	<a href="#">View details</a>	2/24/2025, 8:57:07 AM

Note: This saves the decision artifacts in an underlying GitHub repository from which you will later publish it as a Skill.

\_5. Optionally provide a description and then click the **Share** button.

Share

Share changes to make them available to collaborators.

Describe your changes (optional)

Why did you make these changes?

Cancel **Share**

\_6. Click **x** to close the *Success* message box.

Success

Your changes were successfully shared.

2/24/2025, 9:06:10 AM

\_7. Click the **Back to UsrXXXPurchaseApprovalDecision** link to go back to the decision model.

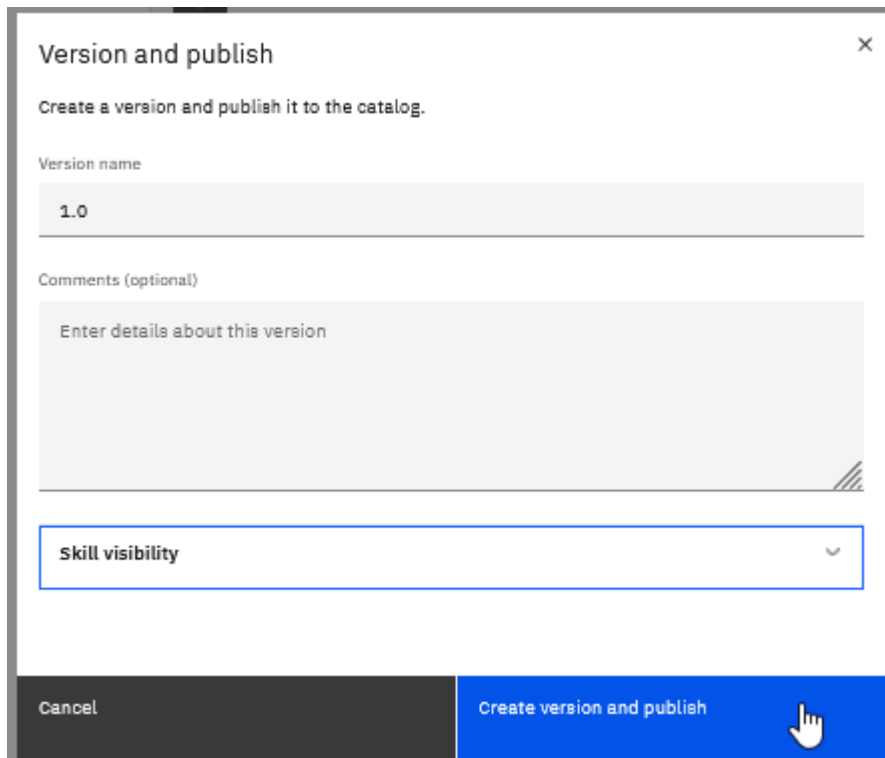
IBM watsonx Orchestrate | Skill studio

[Back to UsrXXXPurchaseApprovalDecision](#)

\_8. Click **Publish**.

Share changes **Publish**

\_9. Set the Version name to **1.0** and click the **Create version and publish** button.



Version and publish

Create a version and publish it to the catalog.

Version name

1.0

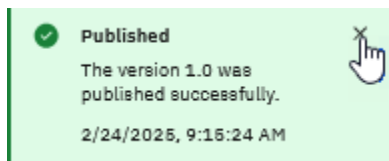
Comments (optional)

Enter details about this version

Skill visibility

Cancel Create version and publish

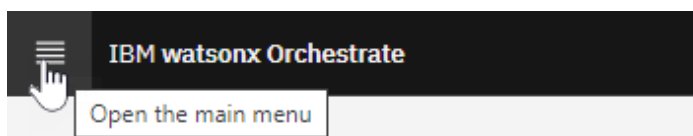
\_10. Click **x** to close the *Published* message box.



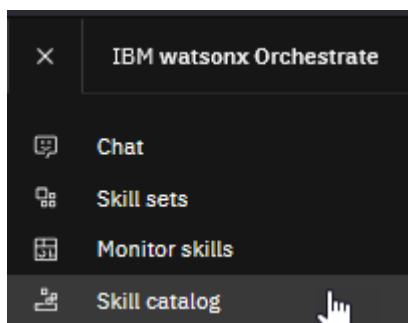
### 3.8 Test the Skill in Chat

#### 3.8.1 Add the Skill to the Personal Skills

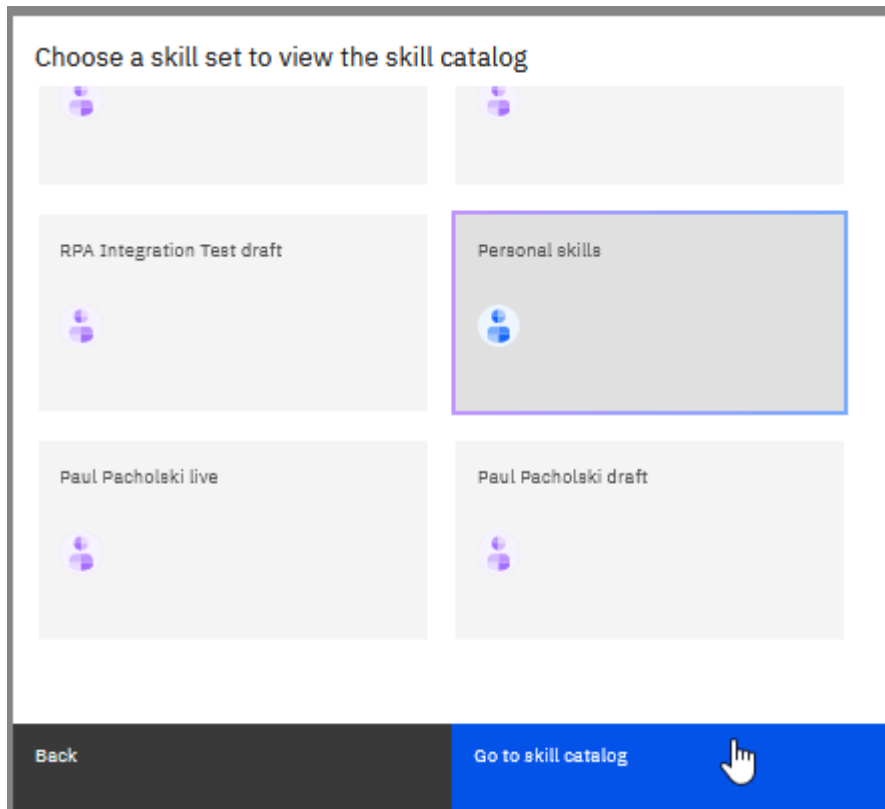
\_1. Click the **Hamburger** menu.



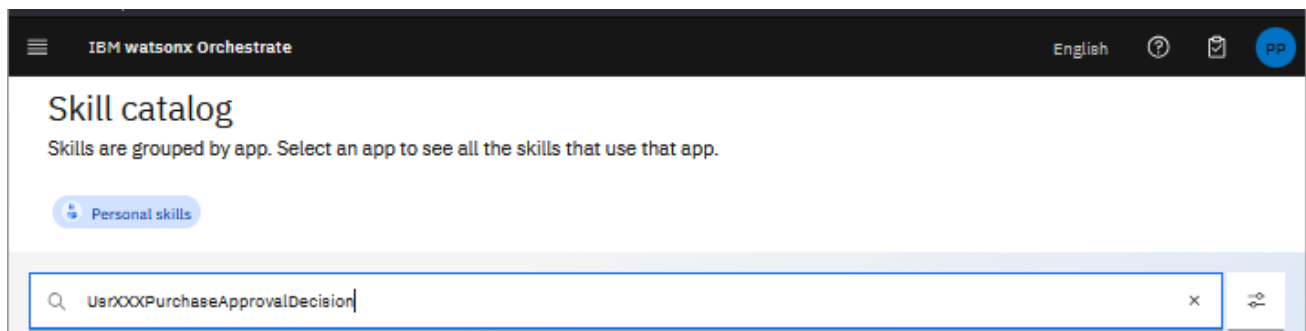
\_2. Click **Skill catalog**.



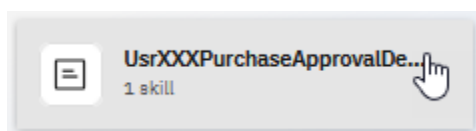
\_3. If you see the *Choose a skill set to view the skill catalog* panel, select **Personal skills**, and click **Go to skill catalog**.



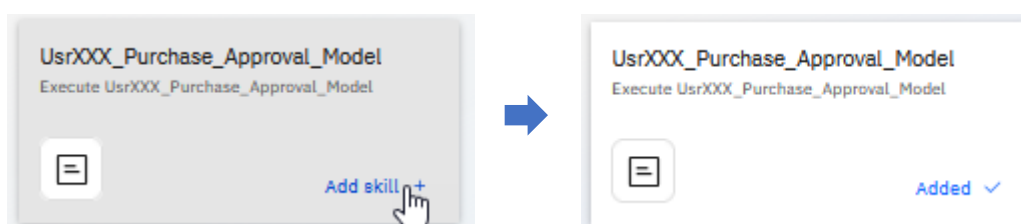
\_6. Type **UsrXXXPurchaseApprovalDecision** (remember to replace XXX with the three-digit number in the user id you have received) in the Skill catalog search bar and press the **enter** key.



\_7. Click the **UsrXXXPurchaseApprovalDecision** tile.

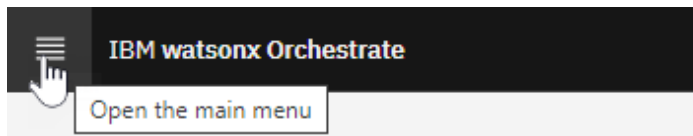


\_8. Click **Add skill +** and verify the status changed to **Added**

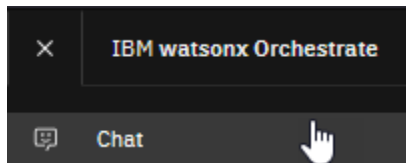


### 3.8.2 Invoke the Skill in the Chat

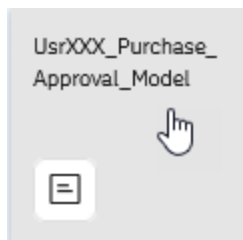
\_1. Click the **Hamburger** menu.



\_2. Click **Chat**.




\_3. Click the skill we just added,



\_4. Enter the input values below, click **Apply**, and verify that the result of the skill invocation is as shown below.

Name	Value
cost	20
category	mouse
overallYearlySpent	150
categoryYearlySpent	40

You just need to complete this form first.

 UstrXXX\_Purchase\_Approval\_Model

cost

20 - | +

category

mouse

approvalThreshold

- | +

overallYearlySpent

150 - | +

categoryYearlySpent

40 - | +

overallYearlyBudget

- | +

Cancel Apply



ItemCategory

mouse

ApprovalThreshold

1000 - | +

OverallYearlySpent

150 - | +

CategoryYearlySpent

40 - | +

OverallYearlyBudget

330 - | +

CategoryYearlyBudget

100 - | +

CategoryApprovalThreshold

60 - | +

Message

Your purchase request has been approved automatically.

Approval

Auto

### 3.9 Summary

In this lab, you learned how to use watsonx Orchestrate to build a decision model for Focus Corp's purchasing assistant. You also learned how to create and configure a decision model, Business Rule, and Decision Table and preview and publish it as a skill.

**Congratulations! You have now completed the watsonx Orchestrate Decision Lab!**

## Notices and disclaimers

© 2024 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

**U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.**

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information.

**This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided. The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

IBM products are manufactured from new parts or new and used parts.

In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

**Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.**

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to ensure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

## Notices and disclaimers (Continued)

Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).