IBM Cloud Pak for Business Automation Demos and Labs 2024

Author Automation with IBM watsonx Orchestrate Automation Builder.

V 1.1

Bu Feng Hou - houbf@cn.ibm.com
Juergen Lukas - jlukas@de.ibm.com
Paul Pacholski - pacholsk@ca.ibm.coom
Zhong Tao Gao - gaozt@cn.ibm.com

Table of Contents

1 Introduction

1.1 IBM watsonx Orchestrate Automations

1.1.1 Automations

A watsonx Orchestrate (wxO) automation is a collection of components (Decisions and Workflows) that fulfill a business purpose. You create automations in the wxO Automation Builder and publish them to the wxO *Skills and apps* page as a skill. Users can then invoke the skill by entering a phrase (that matches the added skill) in the wxO chat bar. Automations created in the automation builder can also be reused in other automations.

1.1.2 Automation Builder

You create your automation and its logic in the wxO Automation Builder. Automation contains one or more automation components. An automation component fulfills a business purpose and contains business logic. The following are the types of automation components currently available:

1. **Decision**

   An automation component type that captures and automates repeatable intelligent business decisions. For example, a business decision to authorize a loan based on established rules and policies. A decision can be implemented with decision models, rule flow models, and prediction models.

   Decision models offer a straightforward and low-code approach to modeling business decisions through a structured, visual representation of a decision.

   Ruleflow models offer a more advanced way to define decisions.

   Prediction models offer insights from historical data to help you make more informed decisions.

2. **Workflow**

   An automation component type that models a business process. For example, a process to approve an upsell offer. A workflow is composed of a sequence of activities and tasks. Tasks include generated user interfaces required to complete tasks by users participating in a workflow. Activities can be other workflows, decisions, automations, or skills you can combine to create powerful automations.

Automation Builder provides all the capabilities needed to develop (no-code editors), test (unit test playback environment), version, and share automations with other authors. It also includes the capability to create operations to publish automations and expose them as skills.

1.2 Lab Scenario and Lab Overview

1.2.1 Client Onboarding Use Case

Focus Corp is a business services provider that offers various services for different industries. Focus Corp uses a fully automated client onboarding solution to provide onboarding services to its clients. Watch this video to see how client onboarding requests are completed: https://ibm.box.com/v/CLIENT-ONBORDING-USE-CASE

1.2.2 Upsell Solution

The company's quarterly services upsell initiative is critical to demand generation; however, it is increasingly difficult to manage and execute. Quarterly sales campaigns are time-consuming and ineffective. It takes over ten weeks to complete quarterly promotions. Due to time limitations, the sales team cannot include all eligible customers. Managers are looking for a faster and more effective way of launching upsell offers.

Two labs cover the Upsell Solution:

1.   Building the User Facing Solution

In this lab, you will build an IBM watsonx Orchestrate solution that re-imagines the quarterly services upsell initiative. You will be reusing the existing IT assets to author an intelligent, AI-driven solution that pulls customer data from the system of records and creates targeted emails.

2.   Building the Sales Campaign Approval Workflow, which is a component of the user facing solution

In this lab, you will use the Automation Builder to create an Automation used in the user facing solution as a skill. This Automation automates the Sales Campaign Approval Process. It is composed of two components:

- A decision that considers the customer scoreboarding information and helps to decide if the upsell offer is to be auto-approved, auto-rejected, or has to be looked at by a sales manager.
- A workflow that orchestrates the Scoreboard skill, the approval decision component, the Manager Approval User Task, and the Send Campaign Email skill.

1.2.3 Sales Campaign Approval Workflow Architecture

1.2.3.1 Automation

In this lab, you will build the Sales Campaign Approval Workflow, a component of the user-facing solution.
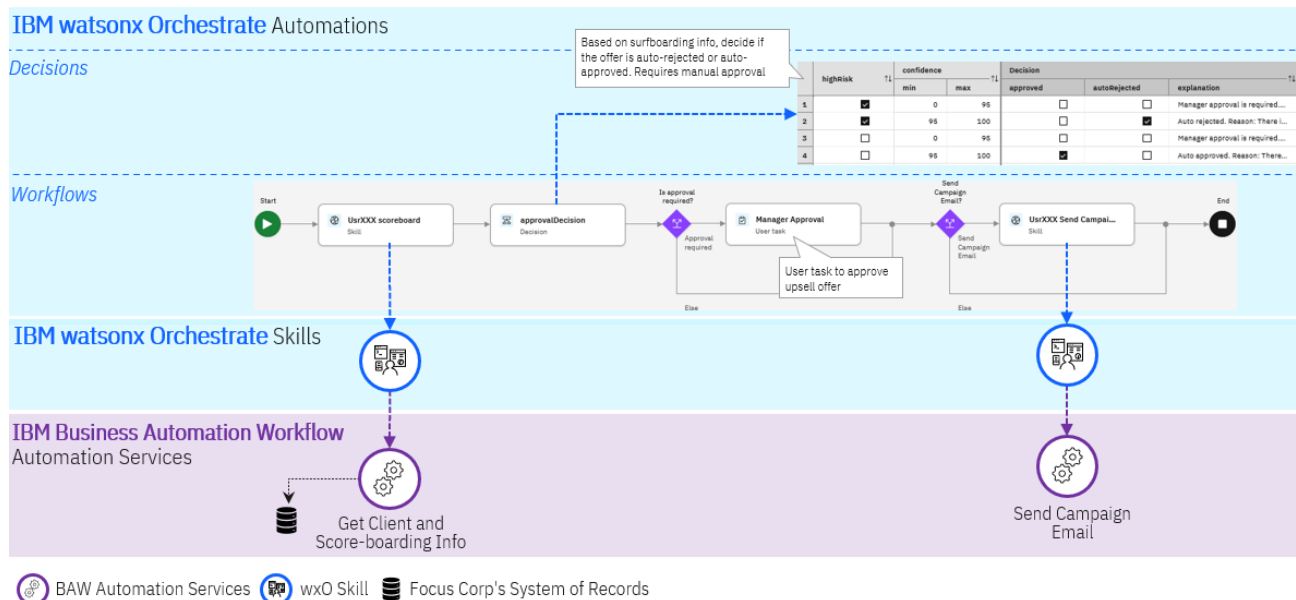

Figure 1. Sales Campaign Approval Workflow Architecture

Let's examine the Activities in this Workflow:

The Start node. The input parameters of the Workflow are client name and upsell services list.

The first Activity is a wxO skill that uses existing IT automation to retrieve customer data and the related scoreboarding information.

The second Activity is implemented in Automation Builder as Decision automation. It uses the scoreboarding information retrieved in the previous skill to determine if the sales campaign will be auto-approved, auto-rejected, or has to be looked at by a sales manager.

The third Activity is a Branch that decides based on the output of the second Activity if a sales manager's approval is required. If it is, the fourth Activity is called. Otherwise, it is skipped.

The fourth Activity is a User Task which provides the user interface with the information (client data, scoreboarding information, list of upsell services) for use by the sales manager, who can change the upsell services and approve or reject the sales campaign.

The fifth Activity is a Branch that checks if the sales campaign has been approved (auto-approved by the Decision or manually approved by the Sales Manager). If approved, it will move to the path executing the sixth Activity. Otherwise, it will be skipped.

🕸 *skill*

The sixth Activity is a wxO skill that uses an existing IT automation. The automation uses the upsell information and customer details to compose a customized upsell offer email message and sends it out to the customer.

⏹

The Workflow output includes the client's name, a list of upsell services (possibly adjusted by the sales manager), the approval decision, and the email status (sent or not sent).

1.2.3.2 Skills

The wxO skills you will use in this lab are implemented by two Service Flows authored using Business Automation Workflow: scoreboard and sendCampaignEmailUAB. These Service flows are exposed as operations (scoreboard and sendCampaignEmailUAB) in a single OpenAPI file.

See the figures below showing the services in Business Automation Workflow:

Figure 2. Service Flows that implement wxO skills

1.2.4 Lab Overview

In this lab, you will perform the following steps to author, test, and publish the Sales Campaign Automation:

1. Import and publish Skills to the Skills Catalog
2. Create the Automation and the Workflow
3. Add and configure the wxO Scoreboard Skill
4. Create and author the Decision
5. Add and configure the Decision
6. Configure the Manager Approval User Task
7. Add and configure the wxO Send Campaign Email Skill
8. Add and configure the Approval Branch and the Send Campaign Email Branch
9. Unit test the Workflow Automation
10. Expose Workflow as a wxO Skill (create a version, create an operation, and publish it as a Skill)
11. Test the Workflow Skill in the Chat

 Approximate Duration of this lab: 2-3 hours.

1.3 Lab Setup Instructions

1.3.1 Systems, Lab Files, and Credentials

1.3.1.1 Tech Jam Event

If you are performing this lab as part of an IBM event, access the document that lists the available systems, URLs, and login instructions.

For this lab, you will need to access:

*   **IBM watsonx Orchestrate**
*   **Local Mail Client**

1.3.1.2 Jam in a Box Event

You can also do this lab as part of [Jam-in-a-Box for Business Automation](#) (starting with V23.0.1). In this case, access to an IBM watsonx Orchestrate instance is required as a prerequisite. You need a Builder or Administrator role to do all parts of the lab. If you are an IBMer or IBM Business Partner, you can look at the wxO content on Seismic to learn how to access such a wxO instance. Otherwise, you can contact your IBM representative to get access. When doing the lab as part of Jam-in-a-Box, please also carefully read the section Lab Considerations in the Jam-in-a-Box Readme.md page to guide you on the necessary access.
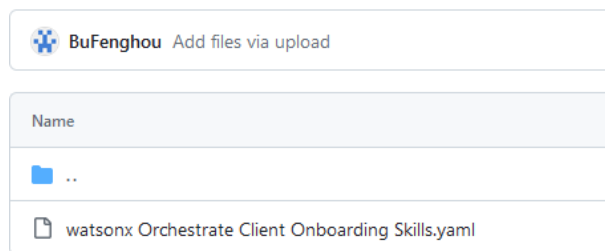
For this lab, you will need to access:

*   **IBM watsonx Orchestrate**
*   **IBM Cloudpak Dashboard**
*   **Local Mail Client**

1.3.2 Customize the OpenAPI File

Download the **watsonx Orchestrate Client Onboarding Skills - UAB.yaml** file from the Lab Data folder to your computer.

The OpenAPI file *watsonx Orchestrate Client Onboarding Skills - UAB.yaml* includes definitions of the REST call to invoke the two REST Services, which expose Service Flows authored in IBM Business Automation Workflow. You will use this file to create Skills in wxO.

Since we are using a shared wxO environment, you need to perform the steps below to ensure your Skills have unique names and to define the connection to the Server where the Service Flows are running.

_1. Use a text editor of your choice to **Open** the *watsonx Orchestrate Client Onboarding Skills - UAB.yaml* file (see 1.3.1 Systems, Lab Files, and Credentials)

_2. Replace **all six** occurrences of **UsrXXX** with **the user id you received** when registering for your CP4BA Credentials., e.g., Usr002 (use Replace all in your editor of choice to ensure you don't miss one or multiple of the six occurrences).

▌ Double-check that you replaced all six occurrences of UsrXXX.

You must change the server URL if you are doing this lab as part of Jam-in-a-Box for Business Automation. You can skip this step if you are doing the lab as part of a Tech Jam.

Replace the placeholder **<IBM Cloudpak Dashboard URL>** with the URL of the **IBM Cloudpak Dashboard** (see 1.4.1 Systems, Lab Files, and Credentials).

_3. **Save and close** *watsonx Orchestrate Client Onboarding Skills - UAB.yaml* file.

1.3.3 Login to wxO

_1. In your Web Browser, open **IBM wastsonx Orchestrate** (see 1.3.1 Systems, Lab Files, and Credentials)

_2. Enter your IBM ID and click **Continue to log in**.

Log in to IBM
Watson Orchestrate

Don't have an account? Book a demo

Continue with IBMid   Forgot ID?

pacholsk@ca.ibm.com

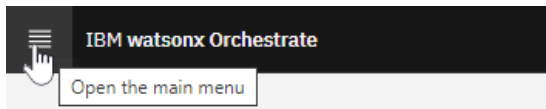Continue   →
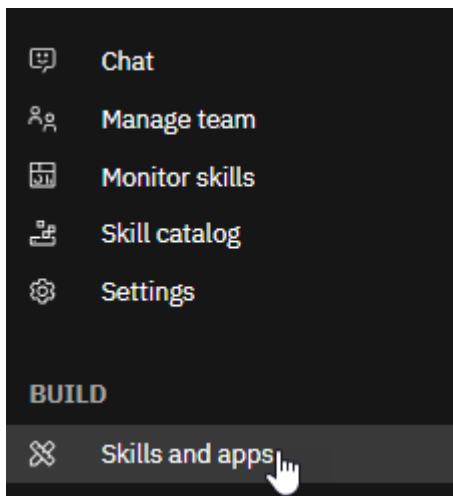
2 Exercise: Import and Publish Skills

Because you have a Builder role, you can create skills from OpenAPI specifications and add them to the Skill Catalog for reuse in wxO.
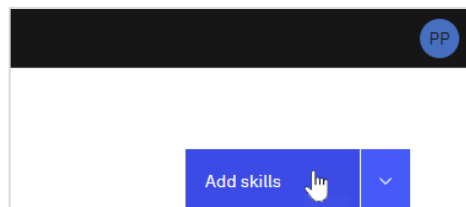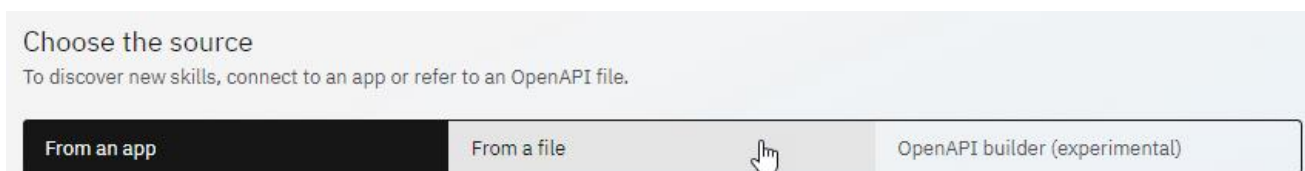
2.1 Import Skills

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

_2. Click **Skills and apps.**

_3. In the top-right, click **Add skills**.

_4. From Choose the source, click **From a file.**

_5. From your file system, **drag and drop** _watsonx Orchestrate Client Onboarding Skills - UAB.yaml_ file to the upload box or click the blue link to upload the file.

Import a skill file

Ensure your file is in the .json, or yaml format and no larger than 50 MB.

Drag and drop files here or click to upload

watsonx Orchestrate Client Onboarding Skills - UAB.yaml

Make sure you see _"...good to go!"_ message.

watsonx Orchestrate Client Onboarding   ×

The OpenAPI file or skill package is good to go! ✓

_6. In the bottom-right corner, click the **Next** button.

Cancel   Next

_7. **Select** all the skills you imported and click the **Add** button.

Choose skills

you can choose which skill to add here

2 out of 2 selected

	Skill	Description	Status
☑	UsrXXX scoreboard	Perform scoreboard to evaluate...	✓ Ready to add
☑	UsrXXX Send Campaign Email UAB	Send campaign email to client ...	✓ Ready to add

Cancel   Add

You will now see a confirmation message.

✓ 2 skills were added successfully.

17:31:03

_8. Click the **Skills** tab.

Skills and apps

Add new skills, train them to be more effective, and publish

Apps | Skills

_9. To see your skill, in the *search bar,* enter **UsrXXX** (where XXX are the digits of your CP4BA Credentials user id) and hit the **Enter key**.

🔍 UsrXXX   ✕

> Notes:
> • Your published skills will be prefixed with UsrXXX (where XXX is the user ID you used when you replaced the value in the yaml file). Also, note that the status of your skills is set to "Ready to publish."
> • If you cannot find your skill using a partial name, try entering the full skill name, i.e., UsrXXX scoreboard.

Skills

	Name	Step in the process	Status	Skill type	Author	Last edited	
⌄	UsrXXX scoreboard	Just 1 step away to be ready	✅ Ready to publish	Imported	pacholsk@ca.ibm.com	March 25 2024	⋮
⌄	UsrXXX Send Campaign Email UAB	Just 1 step away to be ready	✅ Ready to publish	Imported	pacholsk@ca.ibm.com	March 25 2024	⋮

_10. Click **X** to close the message.

✅ 2 skills were added successfully.   ✕

11:49:17

2.2 Publish Skills to the Skills Catalog

The skills in the "Ready to publish" state can be published to make them available in the Skills Catalog. Before publishing, we can make some changes and enhancements. Let's explore Skill enhancement and publishing.

_1. On the row with *UsrXXX scoreboard,* select the **vertical ellipses (3 dots),** then click **Enhance this Skill**.

	Name	Step in the process	Status	Skill type	Author	Last edited	
⌄	UsrXXX scoreboard	Just 1 step away to be ready	✅ Ready to publish	Imported	pacholsk@ca.ibm.com	March 25 2024	⋮
⌄	UsrXXX Send Campaign Email UAB	Just 1 step away to be ready	✅ Ready to publish	Imported	pacholsk@ca.ibm.com	Ma	Enhance this skill
							Export this skill
							Delete this skill

_2. Examine the tabs that we can use to enhance a skill before publishing.

Skills and apps / Enhance this skill

Enhance the "UsrXXX scoreboard" skill

Add details that will make people want to use this skill.

| Name | Input | Output | Security | Phrases | Next best skills |

Name and describe this skill in a way that tells users how it's used and why they would want to use it.   **Preview**

Tab	Enhancements
Name	Name and describe this skill in a way that tells users how it's used and why they would want to use it. The preview shows how a skill tile will appear in the Skill Set and the Skills Catalog.
Input	You can define how the requested inputs, inherited from the OpenAPI, are displayed to the skill users. The inputs are typically requested in a form format. For example, you can define the language that is used in the form that asks for user input on each parameter.
Output	You can configure how to display the output of the tasks to the users. Results are typically shown in table format. By default, the table headers are each output field name. You can rename any header, reorder the results, or hide fields that users are not interested in.
Security	You can review the authentication type and the server, which are inherited from the OpenAPI. For more information, see Configuring the API security scheme.
Phrases	Phrases are the texts you can use to find and use a skill in the chat bar. You can manually enter or generate new phrases based on IBM Granite Models (experimental).
Next best skill	The next best skills feature helps you find skills to complete tasks that naturally follow the current work. You can add up to five skills as the next best skills.

_3. Without making any changes, in the bottom-right corner, click the **Publish** button.

| Cancel | Publish | Save as draft |

_4. **Rerun the Search** to see only your skills.

🔍 UsrXXX   ✕

Note that the state has changed to Published.

Name	Step in the process	Status
⌄ UsrXXX scoreboard	Ready to use	Published

_5. Click **X** to close the message.
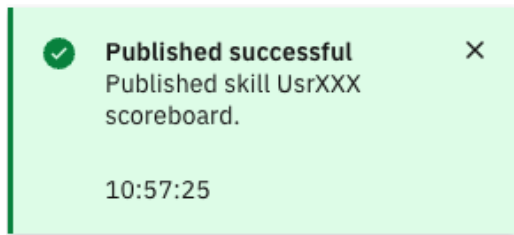
_6. Click the **Skills** tab.
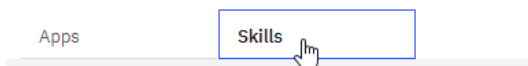
_7. Repeat the above five steps (1-5) to publish **UsrXXX Send Campaign Email UAB** skill.
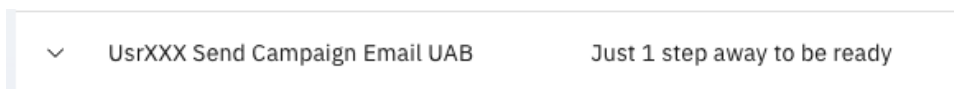
_8. Verify that all the skills you have imported are published.

	Name	Step in the process	Status
⌄	UsrXXX Send Campaign Email UAB	Ready to use	✅ Published
⌄	UsrXXX scoreboard	Ready to use	✅ Published

2.3 Add Skills from the Skill Catalog to your Personal Skills

In the previous steps (in the Builder role), you imported skills using the *watsonx Orchestrate Client Onboarding Skills - UAB.yaml* and added them to the Skills Catalog as an App.
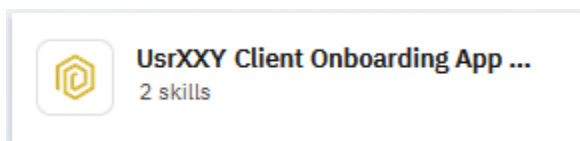
You will now (in the User role) add skills to your Personal Skills from the Skills Catalog. Recall that you have added new skills to the Skills Catalog in the previous part of the lab in the "Builder" role.

> Note: You must add the skills you just added to your personal skills to make them available to Workflow.

_1. Click the **Hamburger** menu in the top-left corner.
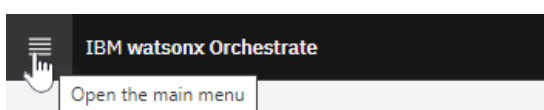
_2. Click **Chat.**
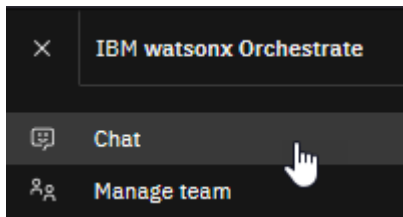
_3. Ensure **Personal skills** (NOT team skills) are selected in the top drop-down.
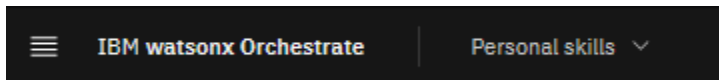
_4. Click **Add skills from the catalog**.
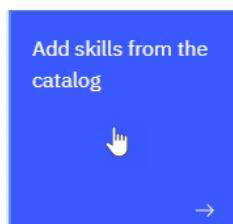
You should now see all the Apps available in the Skills Catalog.

_5. Locate the **UsrXXX Client Onboarding...** (remember that XXX are the last three digits of your CP4BA Credentials user id) by using the search and **click on the tile** to open.
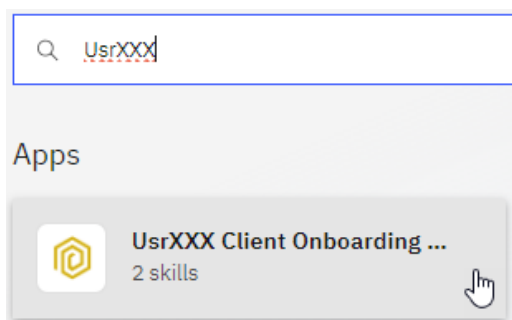
_6. Locate the *UsrXXX scoreboard* skill and click the **Add skill +** button.
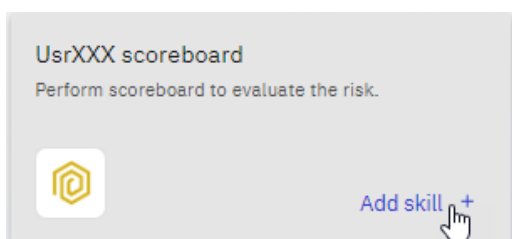
You should now see the *Added check mark for* your skill.
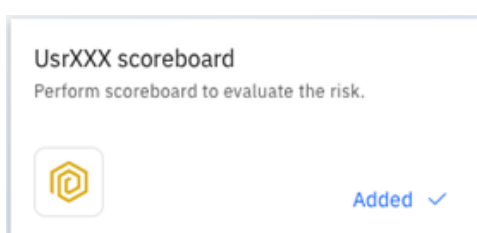
_7. Repeat the above step for the other skill.

_8. Verify that all skills have the *Added check mark*.
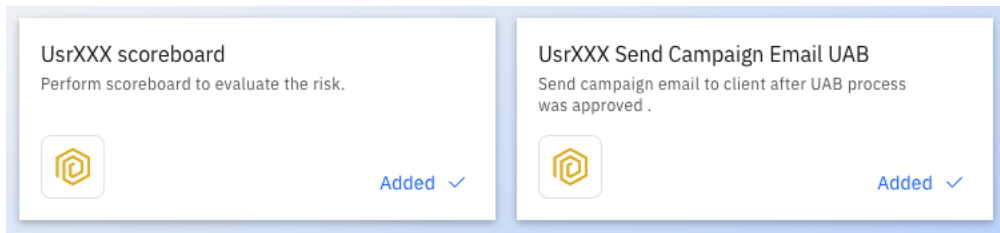
_9. Click **X** on all *Add skill information messages (if any)*.

_10. In the top-right corner, click **Connect app** to specify the security credentials for your skills.
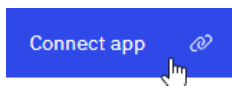
_11. For *username* and *password,* enter your Credentials (see 1.3.1 Systems, Lab Files, and Credentials).

You should now see Connected in the top-right corner.
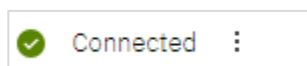
Note that wxO does not verify the login. If you enter an incorrect password or credentials, no error message will be displayed in this step.

2.4 Test the Skill Connectivity

Since wxO does not verify the login when connecting your app, you can test the connectivity using the Chat.

_1. Click the **Hamburger** menu in the top-left corner.
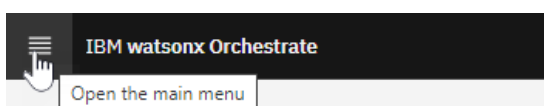
_2. Click **Chat.**
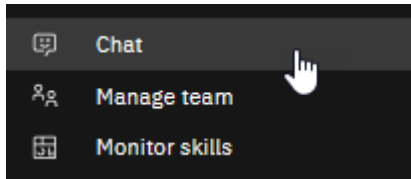
_3. Enter **perform scoreboard to evaluate the risk** in the Chat entry field and click **Send.**
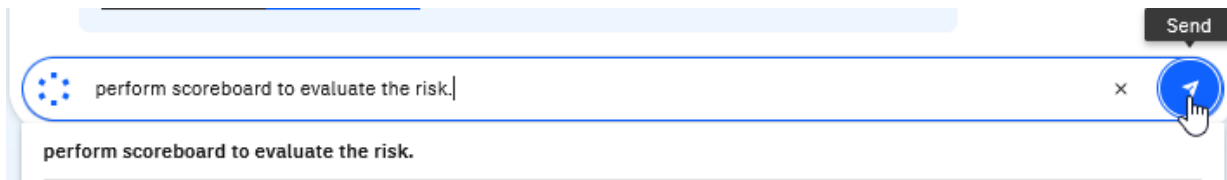
Note: You might see the "Did you mean:" prompt.
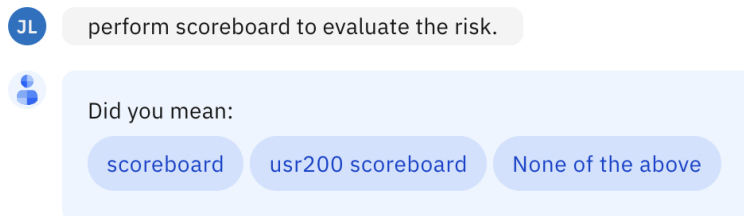
You will only see the prompt if you have completed the Introduction to watsonx Orchestrate lab first. You already loaded the scoreboard skill to your personal skill set as part of the other lab. When doing this lab and adding the UsrXXX scoreboard skill to your personal skill set, you will have two versions of the same skill. wxO needs to clarify which of the two versions you want to call. In this case, choose the one with UsrXXX in front that you created for this lab.

To avoid the prompt, you can use the keyboard interface: click **UsrXXX Client Onboarding App UAB** and then click **the UsrXXX scoreboard.**
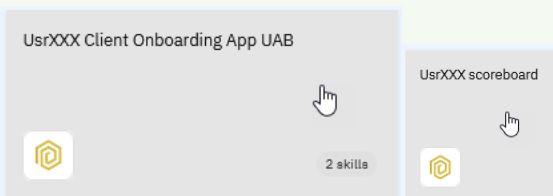
_4. *For ClientName* enter **Automation Elite** and then click **Apply**.

You just need to complete this form first.

UsrXXX scoreboard

ClientName

Automation Elite

Cancel　　Apply

If the connection credentials are correct, you should see a response as shown below:

75　　　　　　　　　　　　　　　　　　　　　　−　　+

Companyage

30　　　　　　　　　　　　　　　　　　　　　　−　　+

Segment

　Segment 1

Servicesrequested　　　　　　　　　　　　　　　　　　　∧

Servicesrequested

　External Audit　　　　　　　　　　　　　　　　　　🗑

Servicesrequested

　Tax Consultation　　　　　　　　　　　　　　　　🗑

　Add　+

Servicefee

25000　　　　　　　　　　　　　　　　　　　　　−　　+

☐ Defaultpayment

If you do not see the response above the first time, try again several times.

If you do not see the response consistently, return to Skill Catalog and enter the correct credentials.

3 Exercise: Author the Sales Campaign Automation

3.1 Create the Automation and the Workflow

3.1.1 Create the Automation

_1. Click the **Hamburger** menu in the top-left corner.
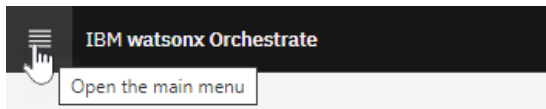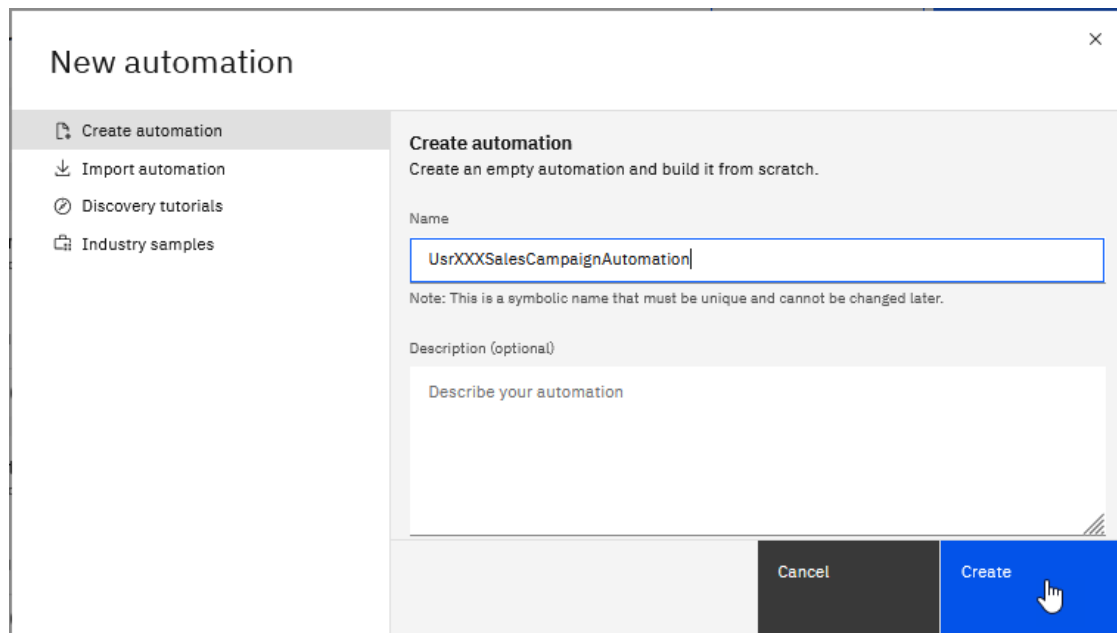
_2. Click **Automations.**
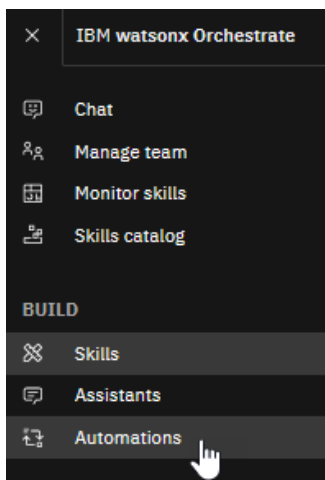
_3. On the top-right, click **Create automation +.**
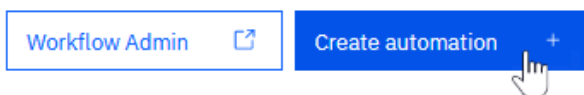
_4. For *Name*, enter **UsrXXXSalesCampaignAutomation** (remember to replace XXX with your CP4BA Credentials user id) and click **Create**.

3.1.2 Create the Workflow

_1. Click the **Workflow** tile.
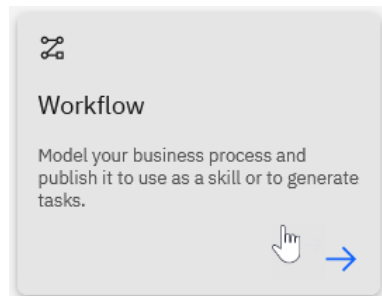
_2. For *Name,* enter **UsrXXXSalesCampaignApprovalWorkflow** and click **Create**.

3.2 Testing your Workflow as you develop

You should now see the Workflow Editor with a single User Task activity.
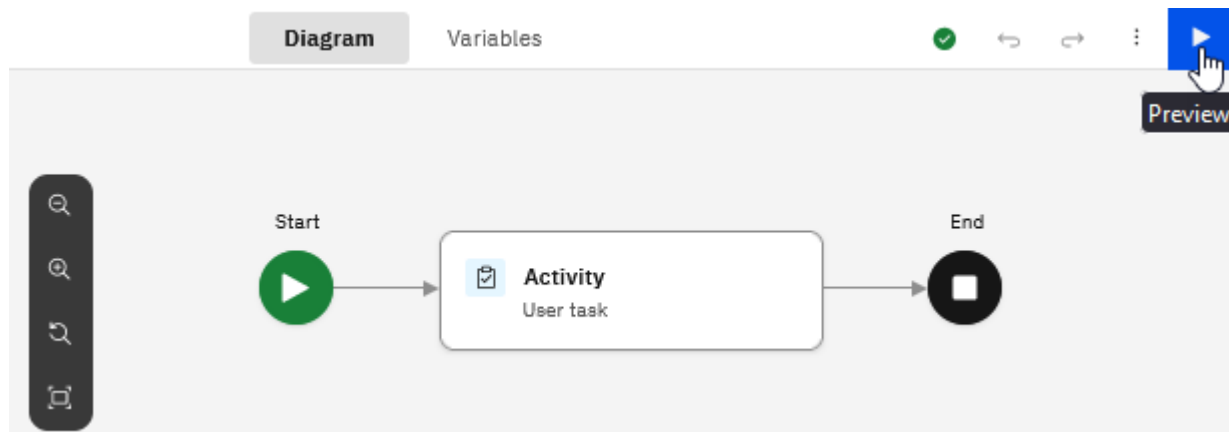
Note the Blue Preview button on the top right corner. You can use this button to test your Workflow anytime you make a significant change and want to test your work. When you preview your Workflow, a test instance is created, where you can ensure that the Workflow runs correctly.

In section 3.10 Test the Workflow Automation, you will find a detailed description of the steps required to test your Workflow.

3.3 Add and configure the wxO Scoreboard Skill

3.3.1 Add Scoreboard Skill

_1. Position the mouse cursor between *Start* and *Activity* and click the **+ icon.**
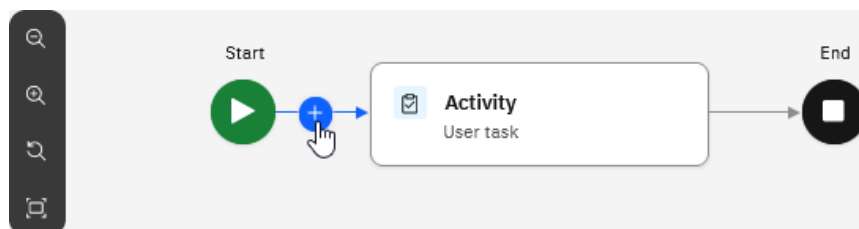
_2. Click **Skill.**
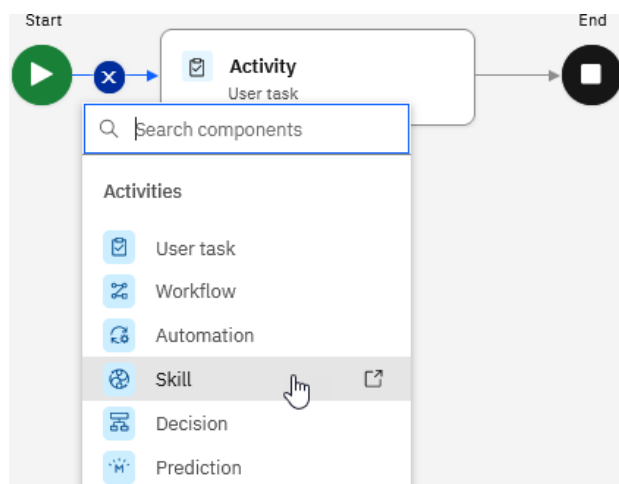
Note, If you see Choose a skill set to view the skill catalog window, select the **Personal skills** tile, and click **Go to Skill catalog.**

Choose a skill set to view the skill catalog

Team skills	Personal skills

Back | Go to skill catalog

_3. Click the **UsrXXX Client Onboarding App UAB** tile (remember that XXX is your CP4BA Credentials user id)

UsrXXX Client Onboarding ...
2 skills

_4. Select the **UsrXXX scoreboard** skill.

UsrXXX scoreboard
Perform scoreboard to evaluate the risk.

_5. Click **Save.**

You should now see the skill added to the Workflow.

_6. Click the **Data** tab.
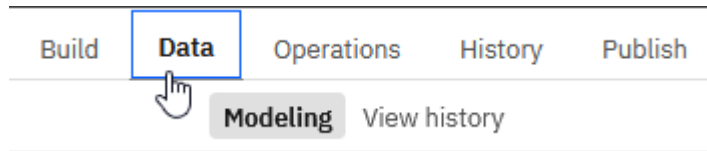
Note: When you added the scoreboard skill, wxO automatically created two data types based on the information about the skill for you: ClientInformationType and scoreboardType. These data types are part of the scoreboard's skill interface. The following steps will create new workflow variables using these data types.
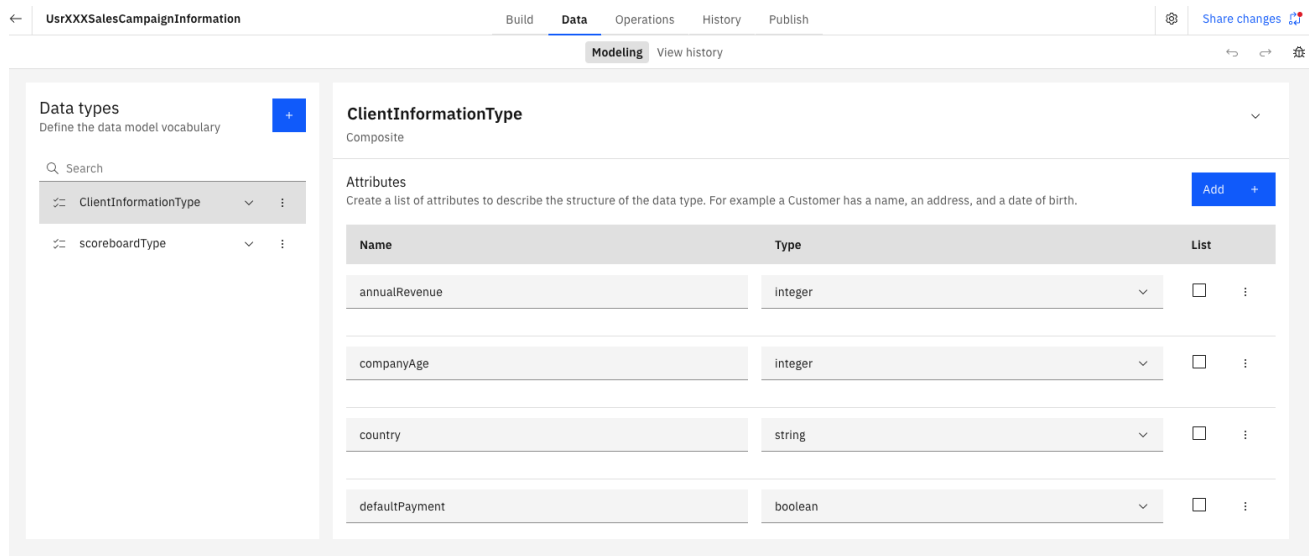
3.3.2 Create the Workflow Variables.

_1. Click the **Build** tab
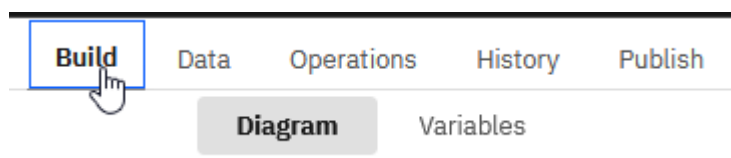
_2. Click the **Variables** tab.
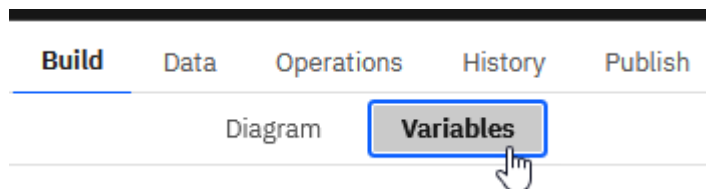
_3. Click the **Create variable +** button **five times** to create five new workflow variables.
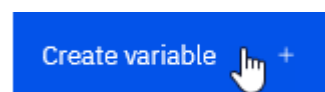
_4. Configure each variable according to the table below

Name	Data Type	List	Input	Output
clientName	String	Single value	Yes	Yes
scoreboard	scoreboardType	Single value	No	No
clientInformation	clientInformationType	Single value	No	No
upsellServices	String	Multiple values	Yes	Yes
sendEmailStatus	String	Single value	No	Yes

The Variables should look as shown below.

Name	Data type	List	Input	Output
clientName	String	Single value	Yes	Yes
scoreboard	scoreboardType	Single value	No	No
clientInformation	ClientInformationType	Single value	No	No
upsellServices	String	Multiple values	Yes	Yes
sendEmailStatus	String	Single value	No	Yes

3.3.3 Define Input and Output Data mapping for the Scoreboard skill

_1. Click the **Diagram** tab.

_2. Click **UsrXXX scoreboard** skill.

_3. Click **Define data mapping** button on the right side.

_4. In the input mapping section, click on **Select a variable** (Input mapping).

_5. Select the **clientName** variable

The Input mapping should look as shown below.

Note: You can also click on Insert suggestions in the top right corner and see which mapping wxO suggests to make your life easier!

_6. Click the **Output mapping** tab.

_7. For the *scoreboard* variable, click **Select a variable.**

_8. Select the **scoreboard** variable.
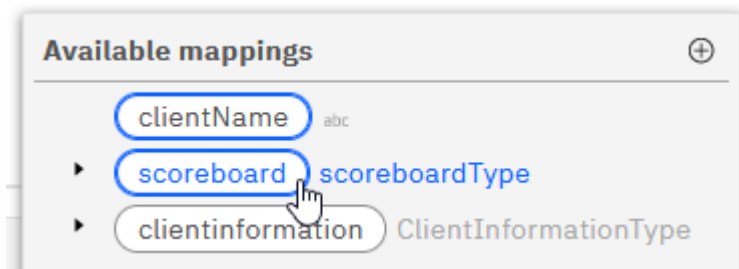
_9. For *ClientInformation* click the **Select a variable** button and then select **clientInformation.**

The Output mapping should look as shown below.

_10. Click the **OK** button to save your variable mapping.

3.4 Create and Author the Decision

3.4.1 Create the Decision

_1. In the bottom left corner, click **Create component +.**
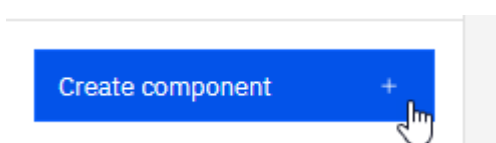
_2. Click the **Decision** tile.
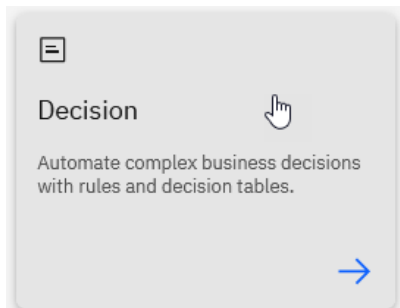
_3. Click the **Decision model** tile.
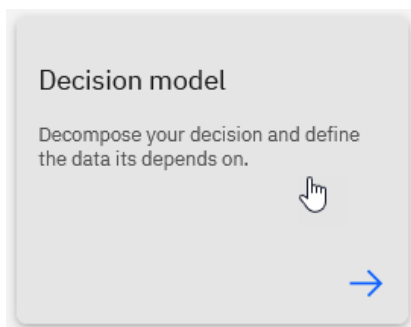
_4. For *Name,* enter **approvalDecision** and click **Create**.

3.4.2 Create Data Types for the Decision

_1. Click the **Data** tab.
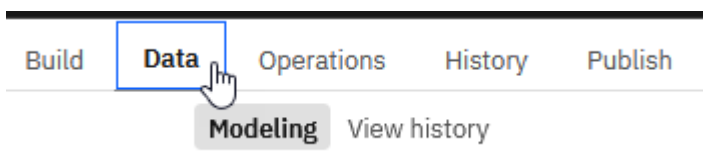
_2. In the upper left area click the **+ button** to create a new Data Type.
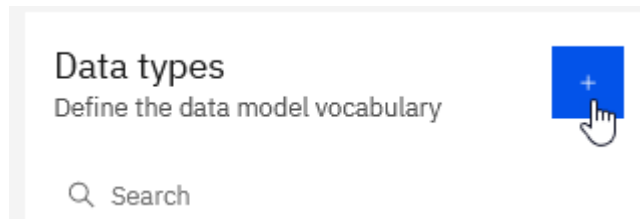
_3. Select **Composite.**
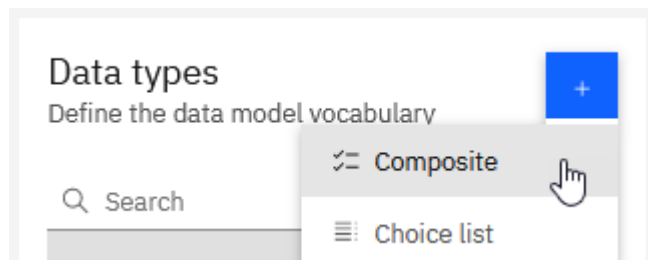
_4. Change the *Composite* from *new type* to **ApprovalDecisionType**.
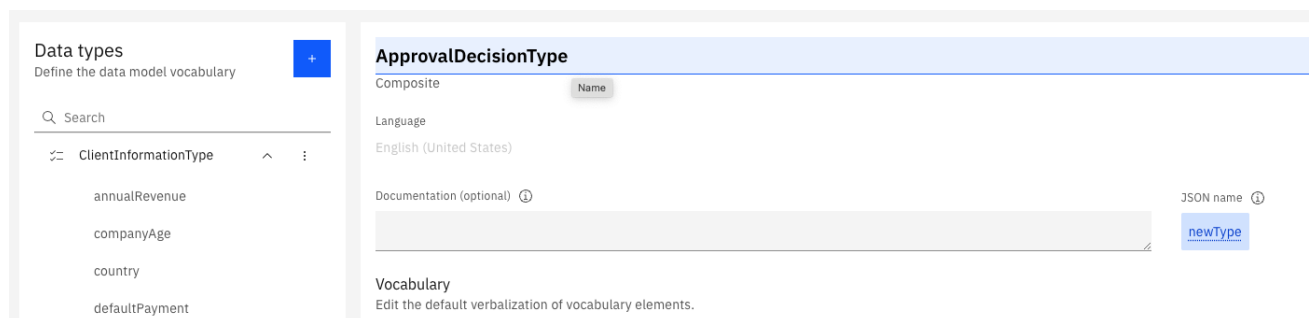
_5. In the lower right area, click the **Add +** button **three times** to add three Attributes.

_6. Configure each Attribute according to the table below:

Name	Type	List
approved	boolean	No
explanation	string	No
autoRejected	boolean	No

The Attributes should look as shown below:

Name	Type	List
approved	boolean	☐ ⋮

an ApprovalDecisionType is not approved, an ApprovalDecisionType is approved

explanation	string	☐ ⋮

the explanation, the explanations, an explanation, explanations

autoRejected	boolean	☐ ⋮

an ApprovalDecisionType is autoRejected, an ApprovalDecisionType is not autoRejected

3.4.3 Define the Output types

_1. Click the **Build** tab.

Build Data Operations History Publish
Diagram View history Dependencies

_2. Click the **Input node.**

Decision
string

↑
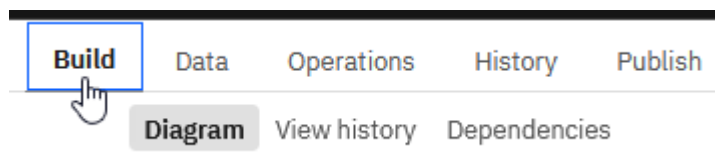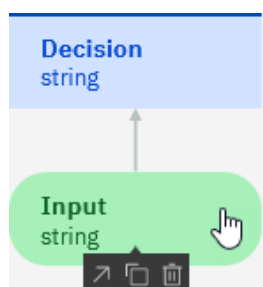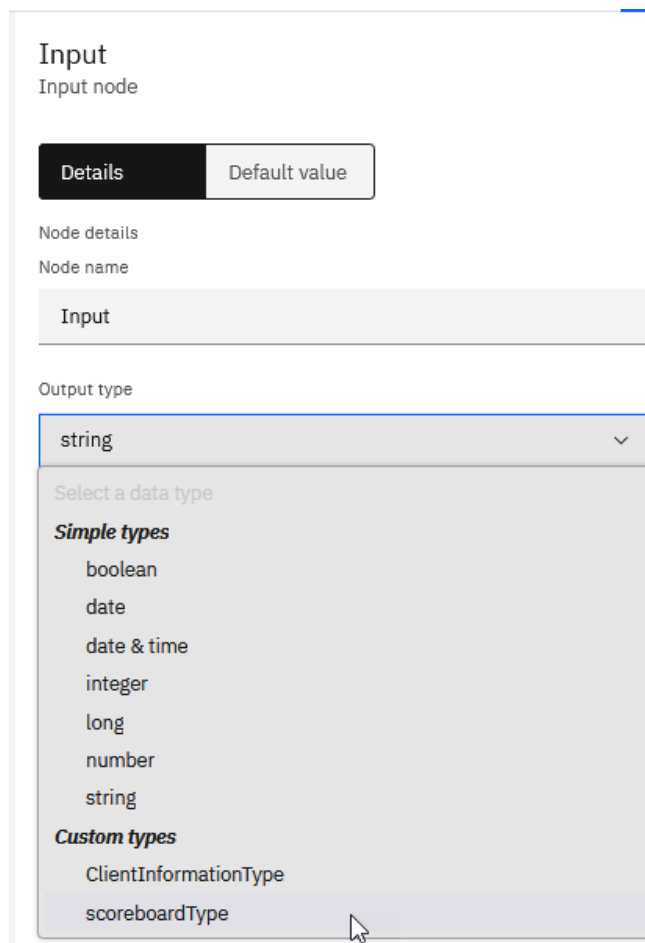
Input
string
↗ ▢ 🗑

_3. In the right section select **scoreboardType** for the *Output type*.

Input
Input node

| Details | Default value |

Node details
Node name

Input

Output type

string
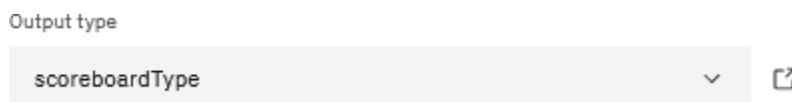
Select a data type
Simple types
boolean
date
date & time
integer
long
number
string
Custom types
ClientInformationType
scoreboardType

The Output type should look as shown below.

Output type

scoreboardType

_4. Click the **Decision** node in the diagram.

Decision
string

Input
scoreboardType
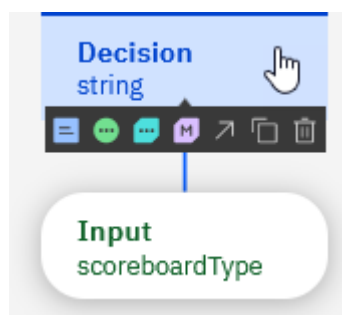
_5. For *Output type,* select **ApprovalDecisionType.**

3.4.4 Create the Decision Table

_1. In the right section, click the **Logic** button.

_2. Click the **+ button** and select **Decision table.**

_3. Enter the *Create decision table* parameters **exactly in the order specified** below and as shown in the screenshot:

> The exact order is essential because it determines how the decision table is built. We need the "High Risk" column before the "Confidence" column for grouping purposes.

1) Name: **approvalDecisionTable**
2) Input, select: **'Input' is highRisk**
3) Input, select: **the confidence of 'input'**
4) Click **Create**

_4. Expand the Decision column to make the data entry easier.

_5. Configure each variable according to the table below.

Row	high Risk	Confidence min	Confidence max	approved	autoRejected	explanation
1	true	0	95	false	false	Manager approval is required. Reason: There is a high risk of payment default, and the confidence level from the predictive score is below 95.
2	true	95	100	false	true	Auto rejected. Reason: There is a high risk of payment default, and the confidence level from the predictive score is above 95.
3	false	0	95	false	false	Manager approval is required. Reason: There is a low risk of payment default, but the confidence level from the predictive score is below 95.
4	false	95	100	true	false	Auto approved. Reason: There is a low risk of payment default, and the confidence level from the predictive score is above 95.

The Variables should look as shown below:

Note that there are some warnings that we need to fix!

_6. Use **Ctrl – Mouse Click** to select **cell 1 and 2** in the **highRisk** column.

	highRisk ↑↓	confidence	
		min	max
1	✓	0	95
2	✓	95	100
3	☐	0	95
4	☐	95	100

_7. With the two cells selected, right-mouse click and choose **Group.**

	highRisk ↑↓	confidence	
		min	max
1	✓		
2	✓		
3	☐		
4	☐		
5			
6			
7			
8			
9			
10			

✂ Cut
▢ Copy
↳ Paste

Insert copied cells
⊟ Insert row >
◌ Clear
Split
Group
Change operator >

_8. Use the above steps to group **highRisk** cells in rows **3 and 4**.

_9. The table should now look as shown below with no error markers in the editor.

	highRisk ↑↓	confidence	
		min	max
1	✓	0	95
2	✓	95	100
3	☐	0	95
4	☐	95	100

Note: The separator line between cells 1 and 2 and cells 3 and 4 in the highRisk row is lighter.

3.5 Add and configure the Decision

3.5.1 Add the Decision Node to the Workflow

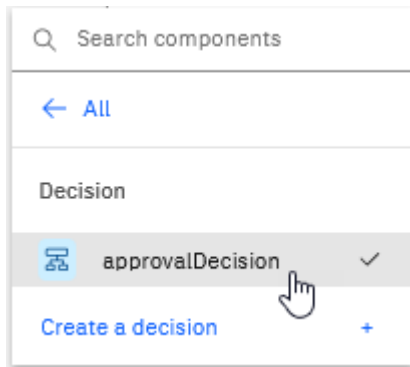_1. In the left pane, select **UsrXXXSalesCampaignApprovalWorkflow.**
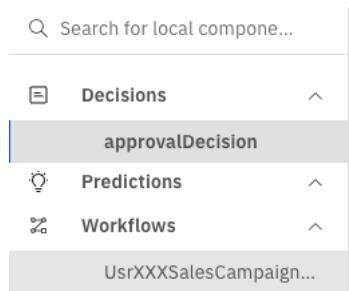
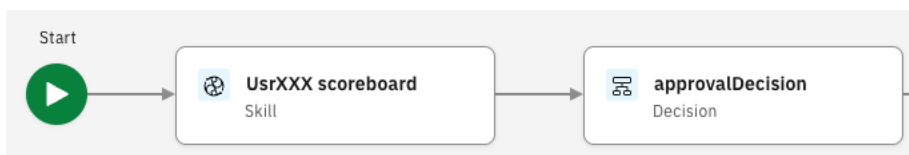_2. Click the **+ icon** between *UsrXXX scoreboard* and *Activity.*
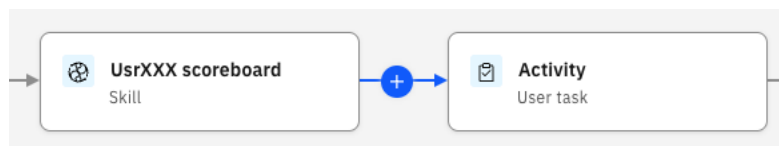
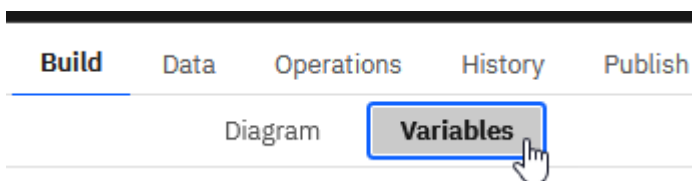_3. Select **Decision (1).**
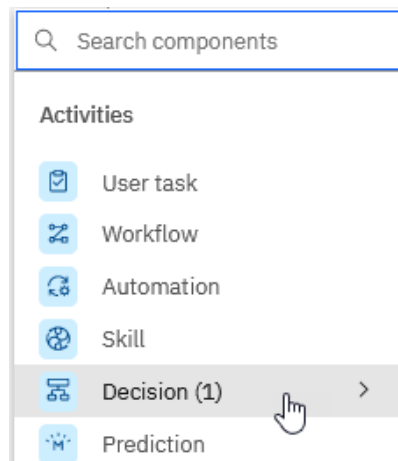
_4. Select **approvalDecision.**

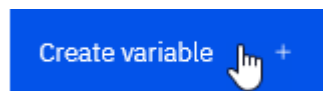You should now see the Decision approvalDecision added as part of the Workflow.

3.5.2 Create a new variable for the Decision Output

_1. Click the **Variables** tab.

_2. Click **Create variable +** to create the output variable.

_3. Configure the new variable according to the table below.

Name	Data Type	List	Input	Output
approvalDecision	ApprovalDecisionType	Single value	No	Yes

The variable should look as shown below.

3.5.3 Define the Input and Output Data Mapping

_1. Click the **Diagram** tab.
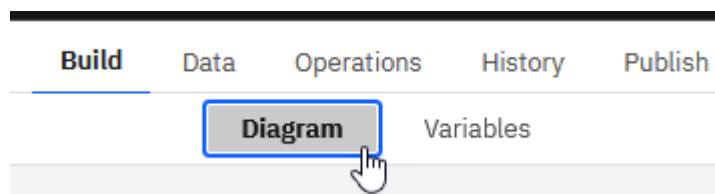
_2. Select the **approvalDecision** Decision.
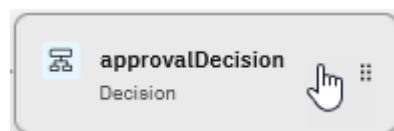
_3. Click **Define data mapping.**
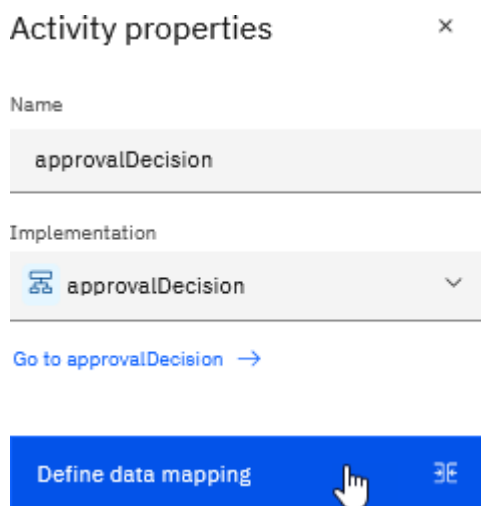
_4. In the input mapping section, click on **Select a variable** (Input mapping tab).
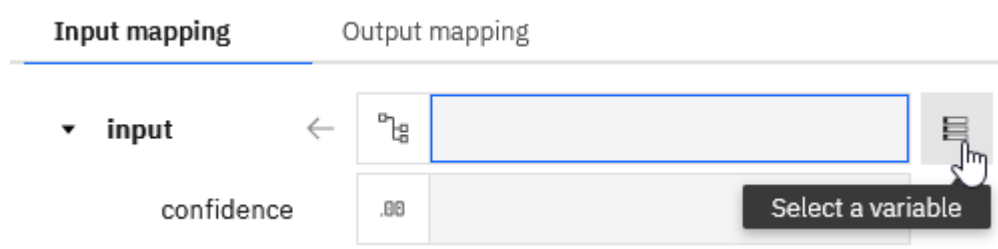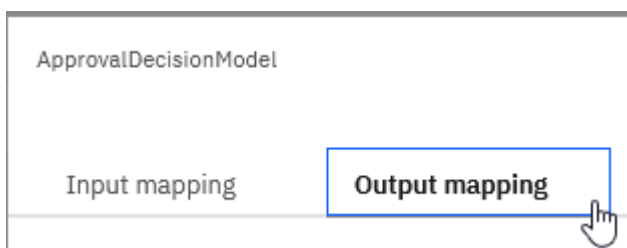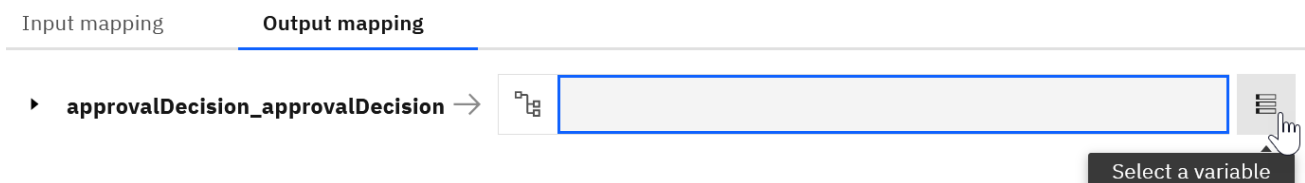
_5. Select the **scoreboard** variable.
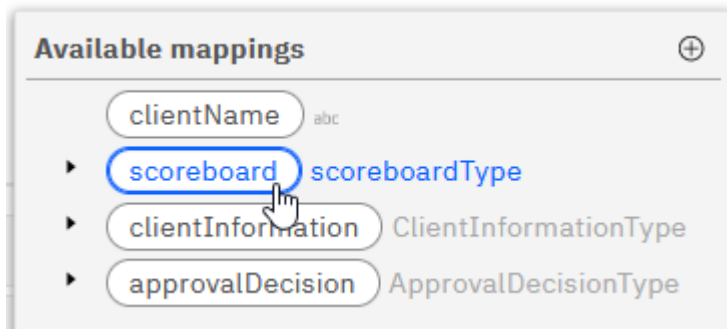
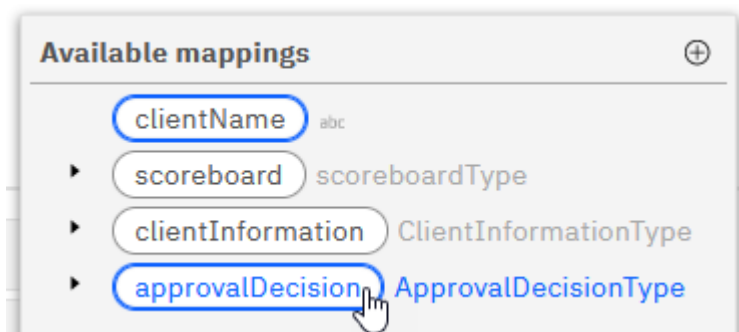The Input mapping should look exactly like this.
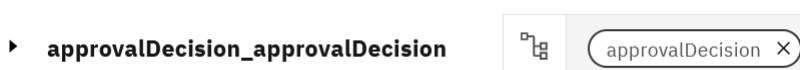

_6. Click the **Output mapping** tab.

_7. For the *ApprovalDecision_ApprovalDecision* variable, click **Select a variable.**

_8. Select the **approvalDecision** variable.
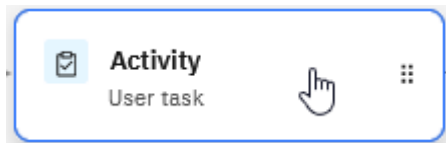

The Output mapping should look exactly like this.

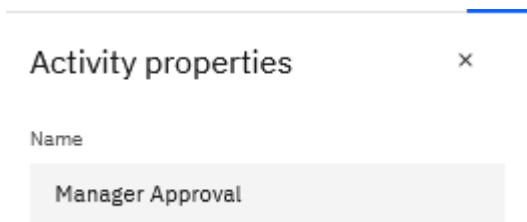_9. Click the **OK** button to save your variable data mapping.

Cancel	OK

3.6 Configure the Manager Approval User Task

3.6.1 Rename the User task

_1. Click the **Activity** User task.

Activity
User task

_2. For *Name*, enter **Manager Approval.**

Activity properties ×

Name

Manager Approval

3.6.2 Assign User

_1. Ensure the *User assignment* is set to **Workflow initiator**

Activity properties ×

Name

Manager Approval

Edit user interface
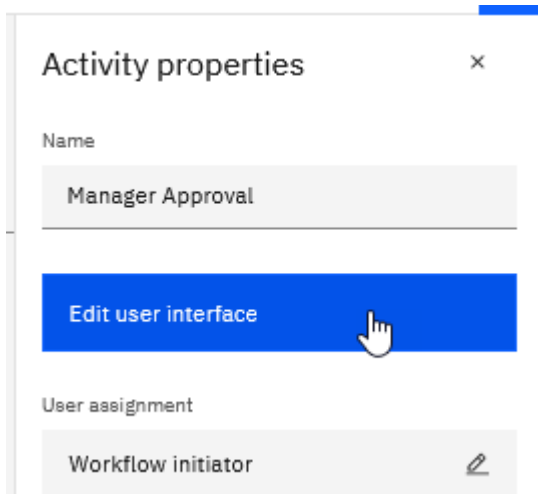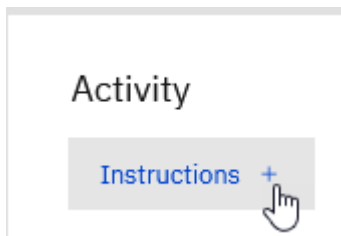
User assignment

Workflow initiator
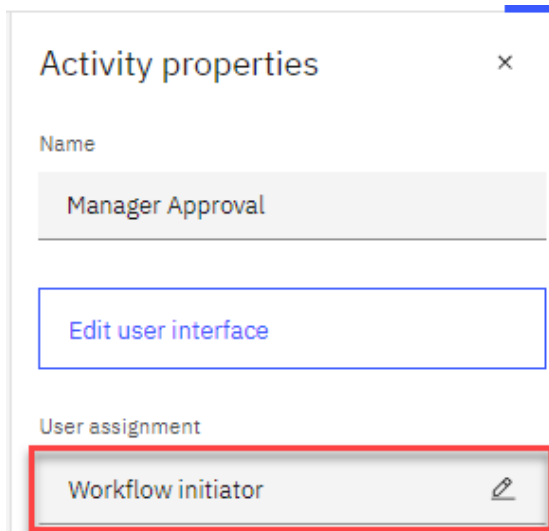
> Note that you would typically select specific users authorized to see and complete a task. In this case, you selected the Workflow Initiator since you will be the only user to run and test this Workflow and User task.

3.6.3 Author User Interface

_1. Click **Edit user interface**.
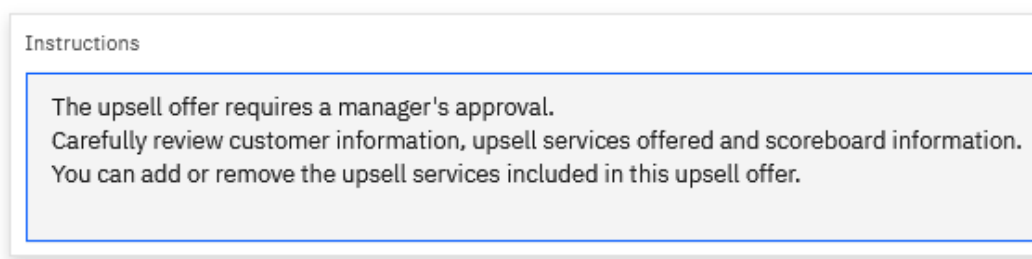

_2. Click **Instructions +.**

_3. In the *Instructions section*, replace *Placeholder text* with the following text:

The upsell offer requires a manager's approval.
Carefully review customer information, upsell services offered and scoreboard information.
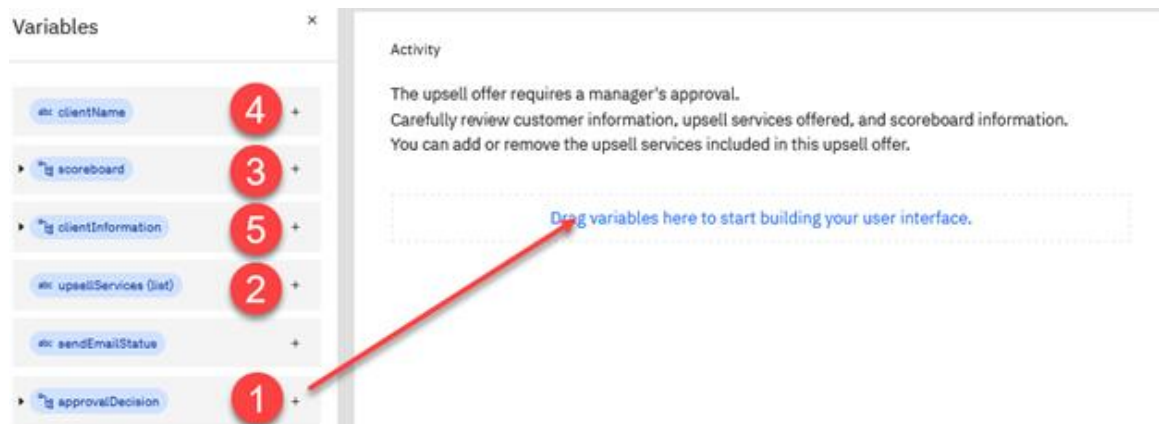**You can add or remove the upsell services included in this upsell offer.**

_4. **Drag and drop** all the variables from the Variables section to the **Drag variables here to start building your user interface** section.

Please follow the order shown in the screenshot below. Drop the next variable below the last one you added (the bottom of the already added variables).

The completed UI form should look as shown in the red rectangle in the figure below:

_5. Click the **autoRejected** checkbox element.

Activity

The upsell offer requires a manager's approval.
Carefully review customer information, upsell
services offered and scoreboard information.
You can add or remove the upsell services
included in this upsell offer.

☐ approved

☐ autoRejected

_6. Set the *Field properties* to **Read-only.**

Activity

The upsell offer requires a manager's approval.
Carefully review customer information, upsell services offered
and scoreboard information.
You can add or remove the upsell services included in this upsell
offer.

☐ approved

☐ autoRejected

Field properties   ×

Associated variable

approvalDecision ⓘ

Label name

autoRejected

Visibility

○ Editable

◉ Read-only

_7. Follow the steps above to set all fields to **Read-only** <u>except the ones listed in the table below</u>.

Filed Name	Visibility	Explanation
approved	Editable	The Approval Decision initially sets the approval. However, when a manager's approval is required, the manager must be able to edit the approved field after reviewing the upsell offer information.
upsellServices	Editable	The manager reviewing the upsell offer can modify the offer by adding or removing the upsell services.

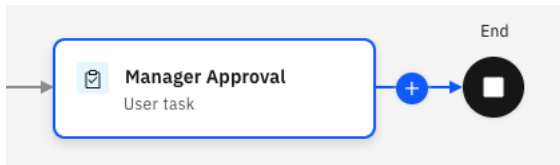_8. In the top left, click the **Back to UsrXXXSalesCampaignWorkflow** link.

☰   IBM watsonx Orchestrate | Automations
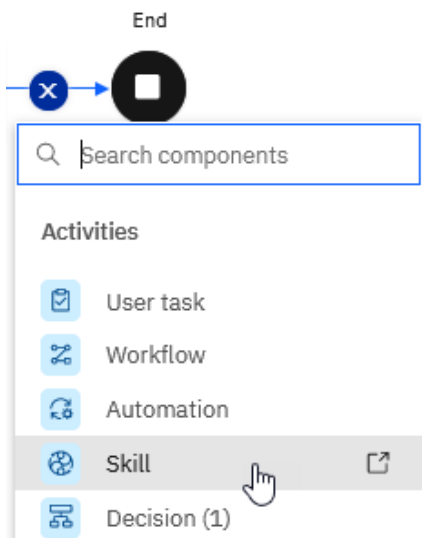
←   Back to UsrXXXSalesCampaignApprovalWorkflow

3.7 Add and configure the wxO Send Campaign Email Skill

3.7.1 Add the Send Campaign Email Skill to the Workflow
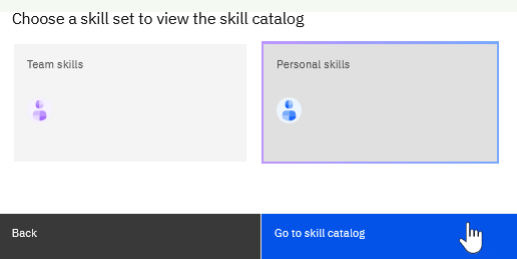
_1. Position the mouse cursor between *Manager Approval* and *End* and click the **+ icon.**
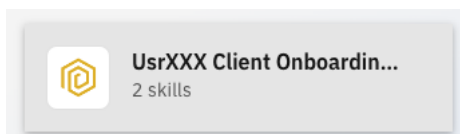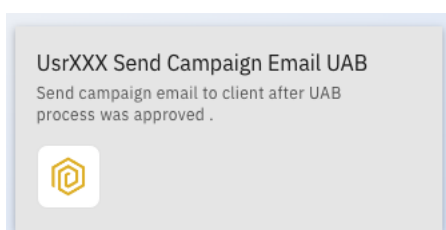

_2. Click **Skill.**

Note if you see this pop-up, select **Personal skills**.

_3. Click the **UsrXXX Client Onboarding App UAB** tile (remember that XXX is your CP4BA Credentials user id).

_4. Select the **UsrXXX Send Campaign Email UAB** skill.

_5. Click **Save.**

You should now see the skill added to the Workflow.
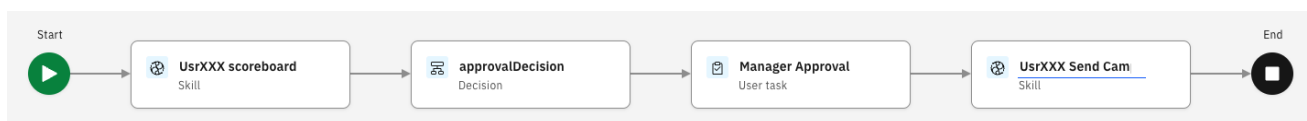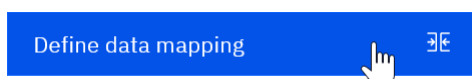

3.7.2 Define the Input and Output Data Mapping
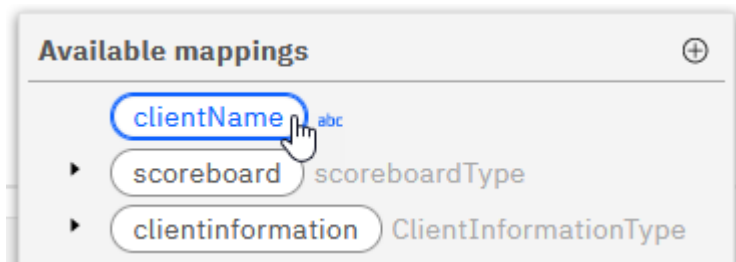
_1. Click the **UsrXXX Send Campai...** Skill.
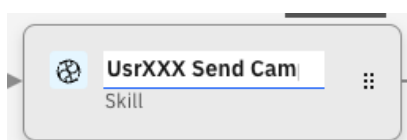
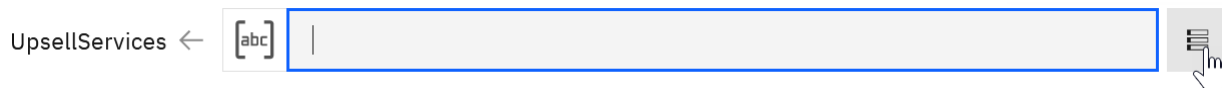
_2. Click the **Define data mapping** button.

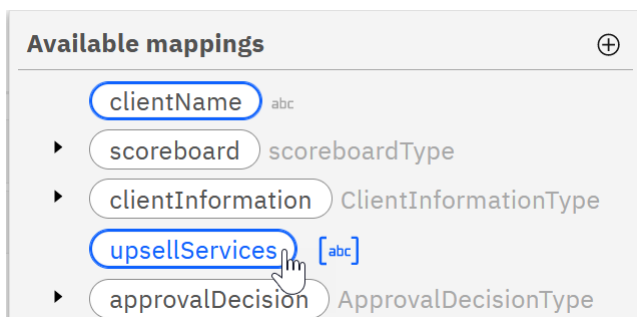_3. In the Input mapping section for ClientName, click on **Select a variable**.

_4. Select the **clientName** variable.

_5. In the input mapping section for UpsellServices, click on **Select a variable.**

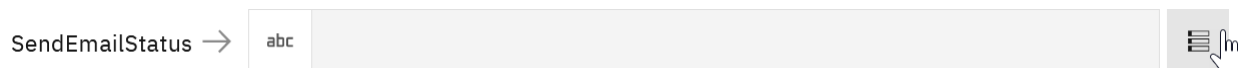
_6. Select **upsellServices.**

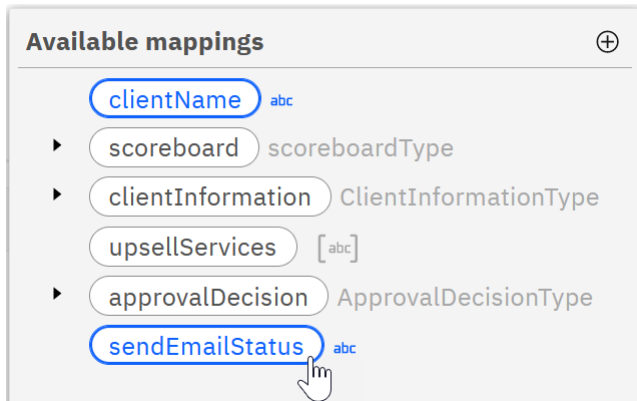The Input mapping should look exactly as shown below.

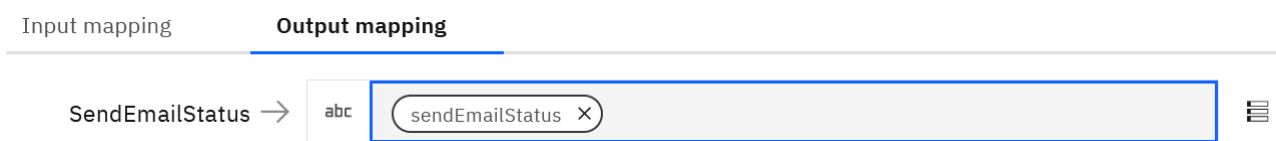_7. Click the **Output mapping** tab.

_8. For SendEmailStatus variable, click **Select a variable.**

_9. Select the **sendEmailStatus** variable.

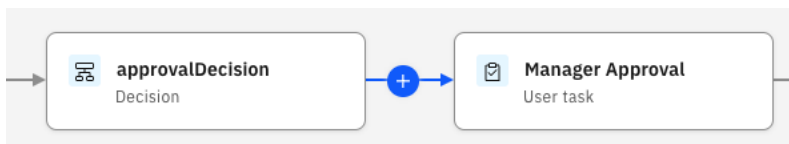The Output mapping should look exactly like this.

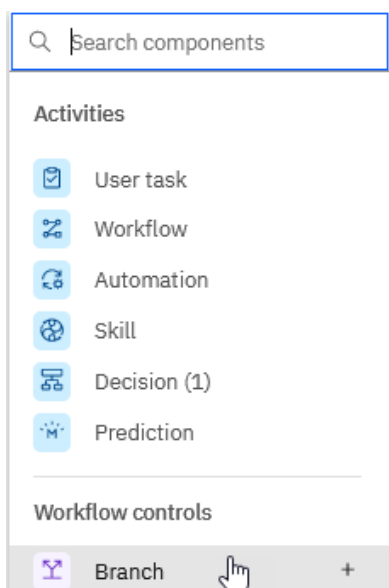_10. Click the **OK** button to save your variable mapping.

3.8 Add and configure the Approval Branch

3.8.1 Add the Approval Branch

_1. Click the **+ icon** between *approvalDecision* and *Manager Approval nodes.*

_2. Select **Branch.**

3.8.2 Configure the Approval Branch

_1. In *Branch Properties*, change the *Name* from *Branch* to **Is approval required?**

Branch properties   ×

Name

Is approval required?

_2. Change the first *Path name* from *Path 1* to **Approval required**.

Path name

Approval required   ⋮

Path name

Else

_3. Click the **Edit conditions** button.

Edit conditions

_4. Click **New condition +** to add a second row.

Route to Approval required if:

All of the following are true   ⌄   New condition +   New condition set +

Click to select a variable   ▤   is equal to   ⌄   Click to select a variable   ▤   ⋮

_5. In the *first condition row*, click **Select variable.**

All of the following are true   ⌄   New condition +   New condition set +

                    Select variable

Click to select a variable   ▤   is equal to   ⌄   Click to select a variable   ▤   ⋮

_6. Select **approvalDecision > approved**.

Available mappings

clientName abc

▸ scoreboard scoreboardType

▸ clientInformation ClientInformationType

upsellServices [abc]

▾ approvalDecision ApprovalDecisionType

approved ●○

autoRejected ●○

_7. For *Select a value* from the drop-down menu, select **False**.

All of the following are true ⌄			New condition +	New condition set +
(approved ✕) ☰	is equal to ⌄	Select a value ⌃ ☰	⋮
		True	
Click to select a variable ☰	is equal to ⌄	False 🖱 ☰	⋮

_8. In the second condition variable, click the **Select variable** button, and for the variable, select **approvalDecision > autoRejected**

(autoRejected ✕) ☰	is equal to ⌄	Select a value ⌄ ☰	⋮

_9. For *Select a value* from the drop-down menu, select **False**.

The condition set should look as shown below.

All of the following are true ⌄			New condition +	New condition set +
(approved ✕) ☰	is equal to ⌄	False ⌄ ☰	
(autoRejected ✕) ☰	is equal to ⌄	False ⌄ ☰	

> Note that "All of the following are true" is selected at the top, i.e., this path is only taken if the upsell offer is neither auto-approved nor auto-rejected based on the previously defined decision service.

_10. Click **Save** to save the conditions.

Cancel	Save 🖱

3.8.3 Move the Manager Approval task to the Approval required Path

_1. In *Manager Approval*, select the **User task handle** (the area with dots on the right side within the activity box)**.**

> 📋 **Manager Approval**
> User task   ⠿

_2. Click and hold the left mouse button and move **Manager Approval** User task to the **Approval required** branch.

> You can release the mouse button when the Approval required line becomes blue. If needed, you can also use the zoom-in function of the diagram to position the Activity better.
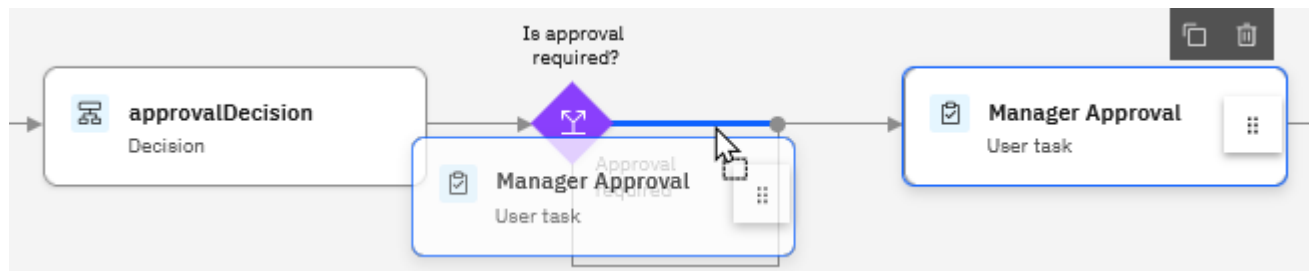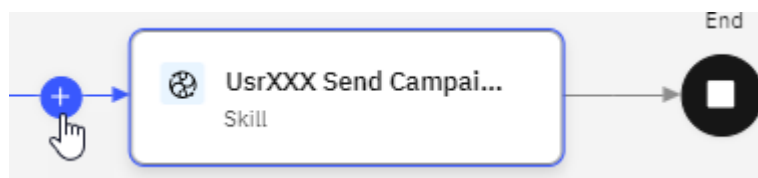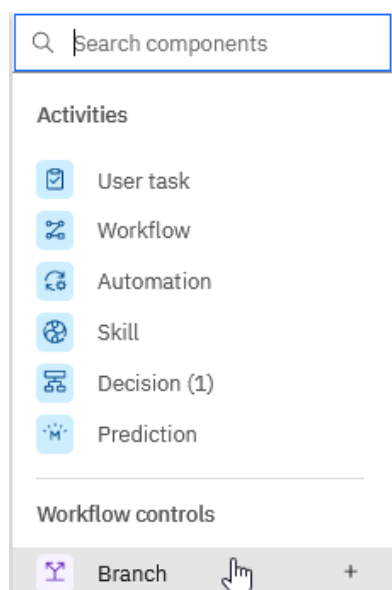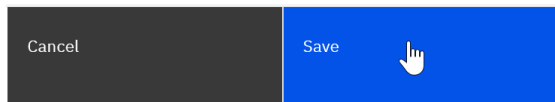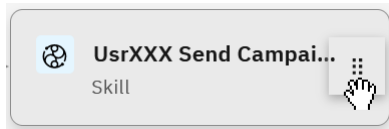

The flow should now look as shown below.
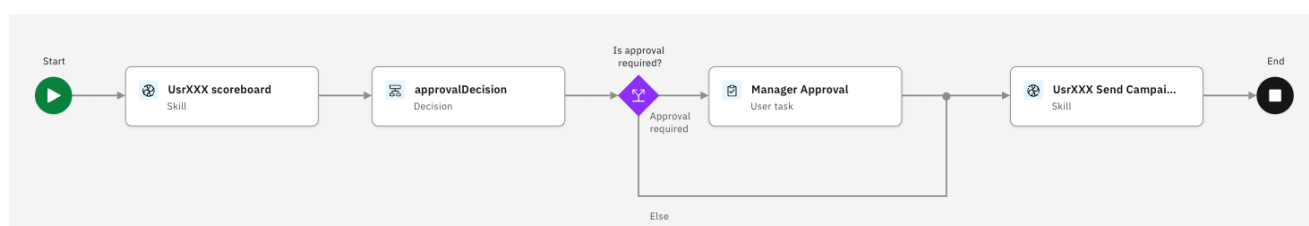
3.9 Add and configure the Send Campaign Email Branch

3.9.1 Add the Send Campaign Email Branch

_1. Click the **+ icon** to the left of **UsrXXX Send Campai...** *Activity.*

_2. Select **Branch.**

3.9.2 Configure the Send Campaign Email Branch

_1. In *Branch Properties*, set *Name* to **Send Campaign Email?**

_2. Change the first *Path name* to **Send Campaign Email**.

_3. Click the **Edit Conditions** button.

_4. In the condition variable, click **Select variable.**

_5. Select **approvalDecision > approved.**

_6. For *Select a value* from the drop-down menu, select **True**.

_7. Clock **Save** to save the conditions.

3.9.3 Move the Send Campaign Email task to Send Campaign Email Path

_1. Select Skill UsrXXX Send Campai... **handle.**

_2. Click and hold the left mouse button and move **Skill UsrXXX Send Campai...** to the **Send Campaign Email** branch.
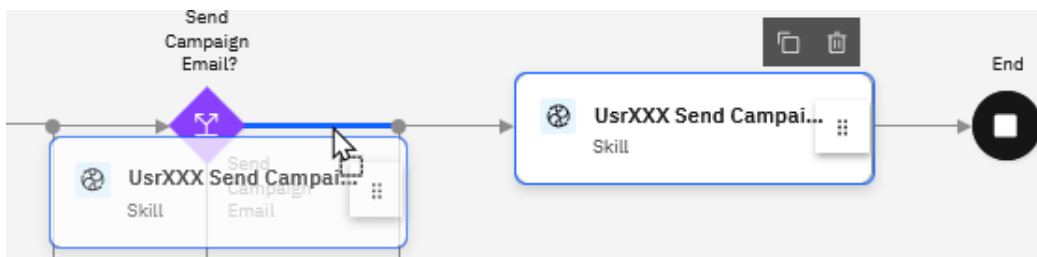
The flow should now look as shown below.
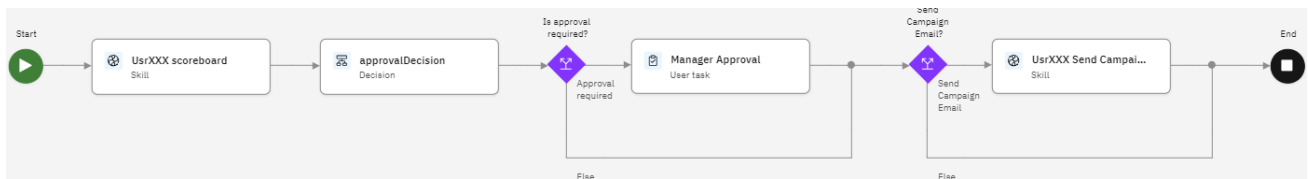
3.10 Test the Workflow Automation

3.10.1 Test 1 – Approval Required Case
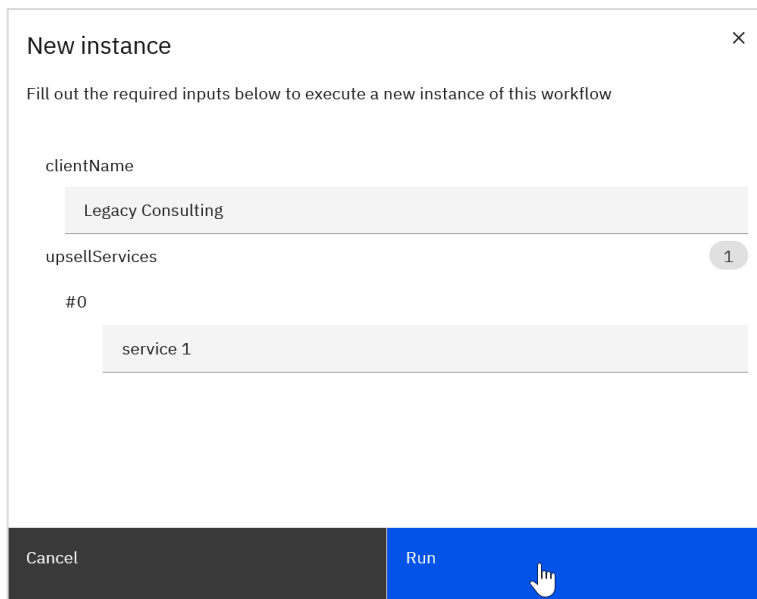
_1. Click the **Preview** button (top right corner).
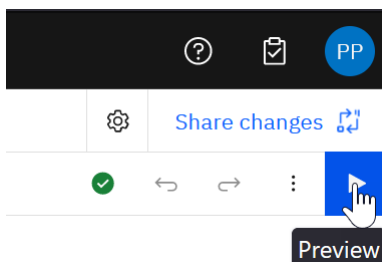
_2. Select **New instance.**
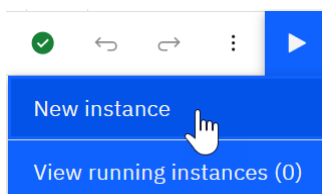
_3. In the *New Instance* window for *clientName* enter **Legacy Consulting,** and for *upsellServices* enter **service 1** and click the **Run** button.
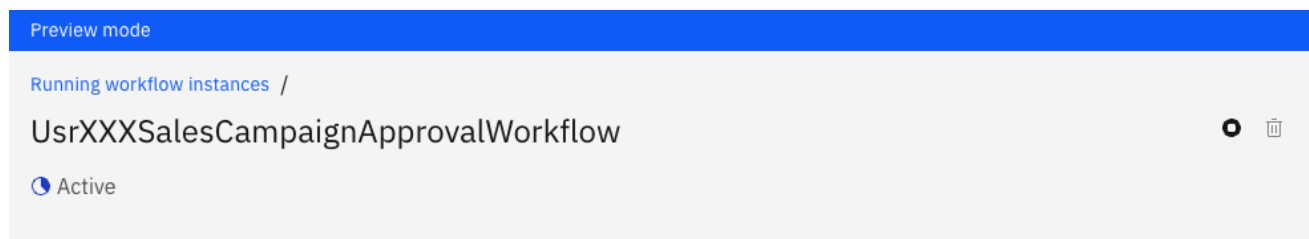
_4. You should shortly see a *Preview mode* Web Browser window with an **Active** status, but there are probably no tasks.

> If you cannot see this window, check if this pop-up window is blocked in your browser. If so, change your browser setting to allow pop-ups from wxO.
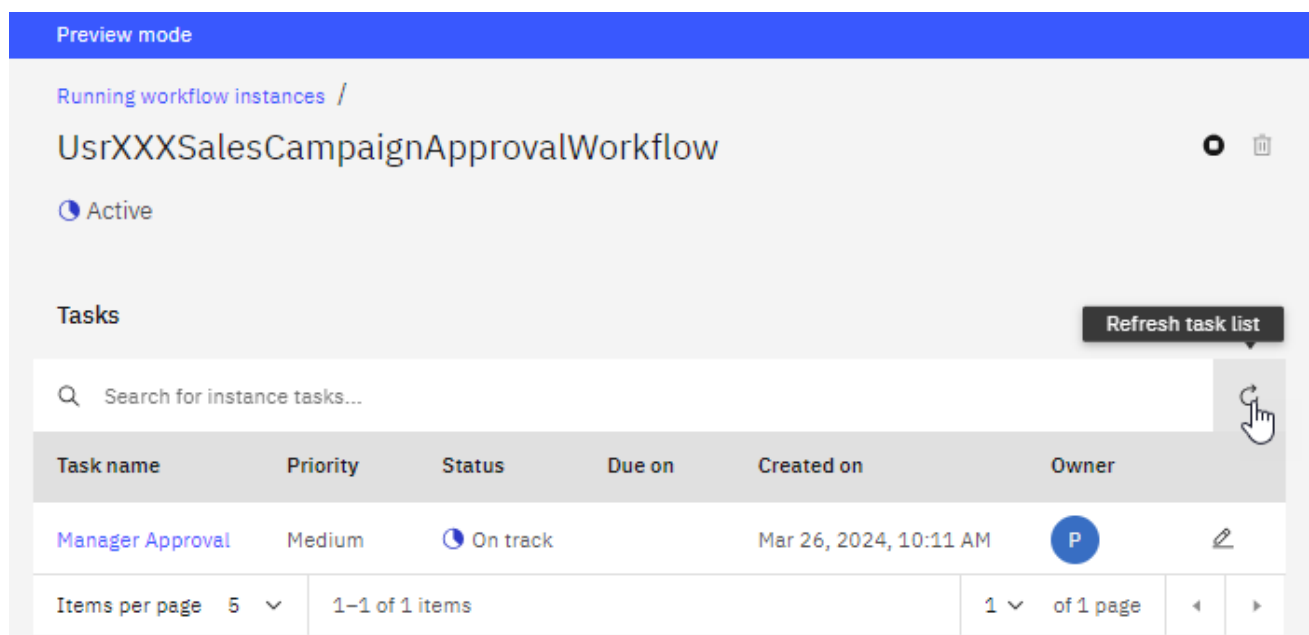
_5. Click **Refresh task list** or just wait until you see a task.

Recall that approval is required because Legacy Consulting is a low-risk customer, but it has a low confidence (of being low-risk) score.

_6. Click **Manager Approval** (the task name) to claim it and open the task UI

Task name	Priority	Status	Due on	Created on	Owner	
Manager Approval	Medium	On track		Mar 26, 2024, 10:11 AM	P	✎
Items per page 5 ∨	1–1 of 1 items				1 ∨ of 1 page	◄ ►

_7. Check **approved** check box and click the **+** in *upsellServices*, then add **service 2.**

_8. Scroll to the bottom of the form and click **Submit.**
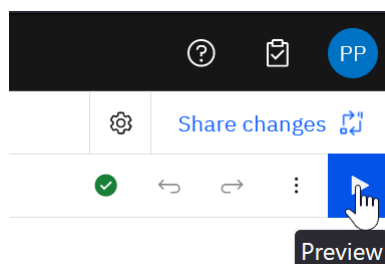
_9. **Close** the *Preview mode* Web Browser window.
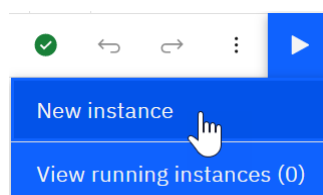

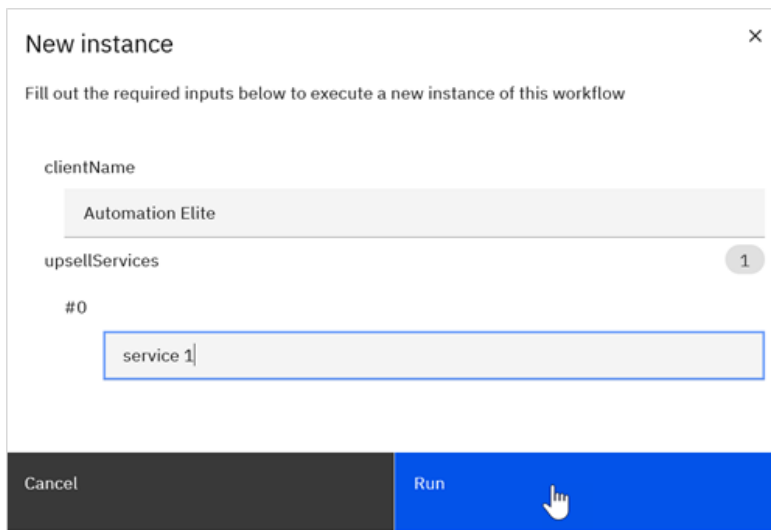## 3.10.2 Test 2 – Auto Approved Case

_1. Click the **Preview** button (top right corner).

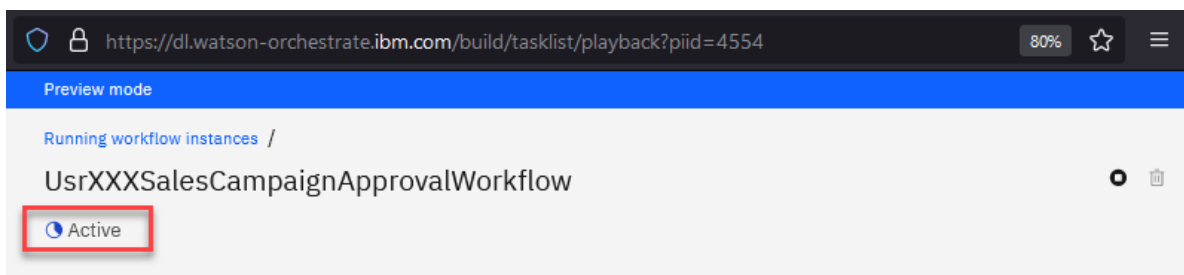

_2. Select **New instance.**

_3. In the *New Instance* window, change the *clientName* to **Automation Elite,** and for *upsellServices* keep **service 1** and then click **Run.**
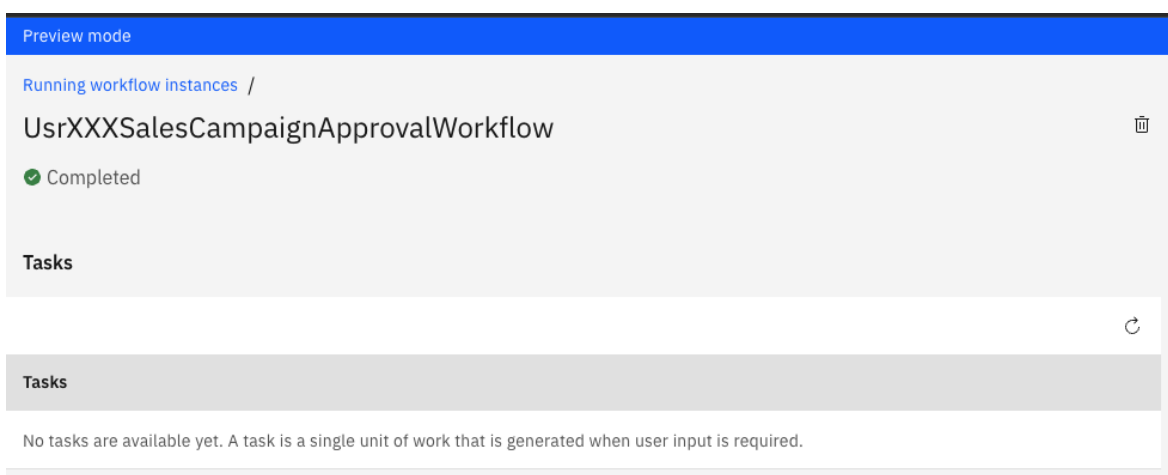

When the Run button stops spinning, you should see the Preview mode window with the instance's state set to *Active*.
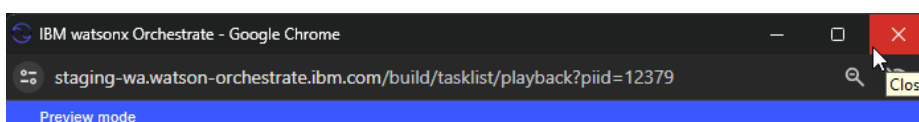

_4. Wait. You should shortly see a *Preview mode* Web Browser window with a **Completed** status for the Workflow.

Since Automation Elite is a low-risk customer with a high confidence (of being low-risk) score, the upsell offer is auto-approved. Hence, we do not see a user task.

_5. **Close** the *Preview mode* Web Browser window

### 3.10.3 Examine Completed Instances in the Process Admin Console

> **If the wxO user id you are using has the Builder role, skip this section and go directly to [3.9.4 Verify the Upsell Offer Email is in the Client's Mailbox](#).** Alternatively, look at the flow of actions to learn how to use Workflow Inspector to examine workflow instances.
>
> The following section requires the wxO Admin role and cannot be performed with the Builder role only. You will see no instances in the Process Admin Console or the Inspector with the Builder role.
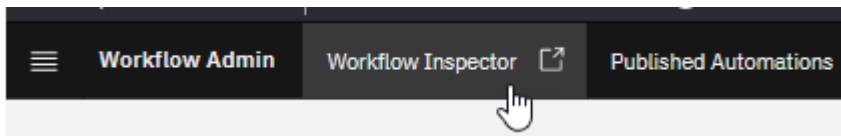
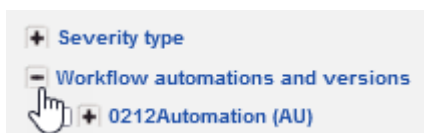_1. Click the **left arrow** in the top left corner.
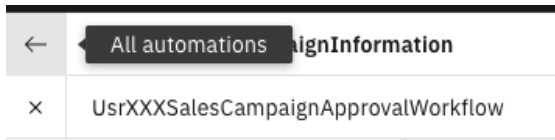
_2. Click **Workflow Admin.**
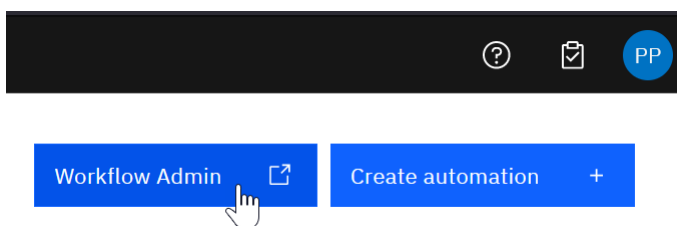
_3. Click **Workflow Inspector.**

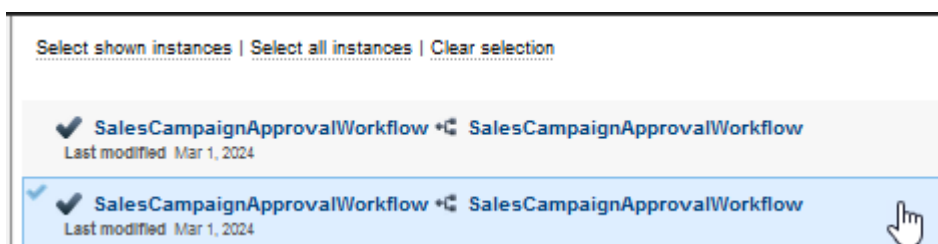_4. **Expand** *Workflow automations and versions*

_5. Find and select your Workflow automation - **UsrXXXSalesCampaignAutomation (UXXXSCA)**

Note that UsrXXX is your ID, and the acronym in the brackets may differ.

_6. Click the **second instance** (that corresponds to the first run, with Legacy Consulting as input).

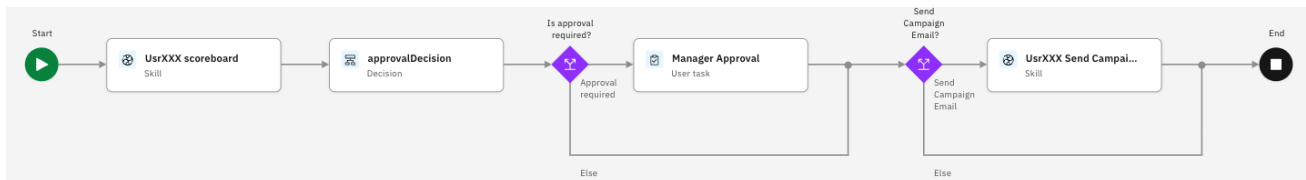_7. Expand **Tasks (4).**

Note the execution path you need to read bottom up and compare it with the Workflow. It includes the manual approval step.

_8. Select the **first instance** (that corresponds to the first run, with Automation Elite as input).

_9. Expand **Tasks (3).**

Note the execution path and compare it with the Workflow. It does not include a manual approval activity.
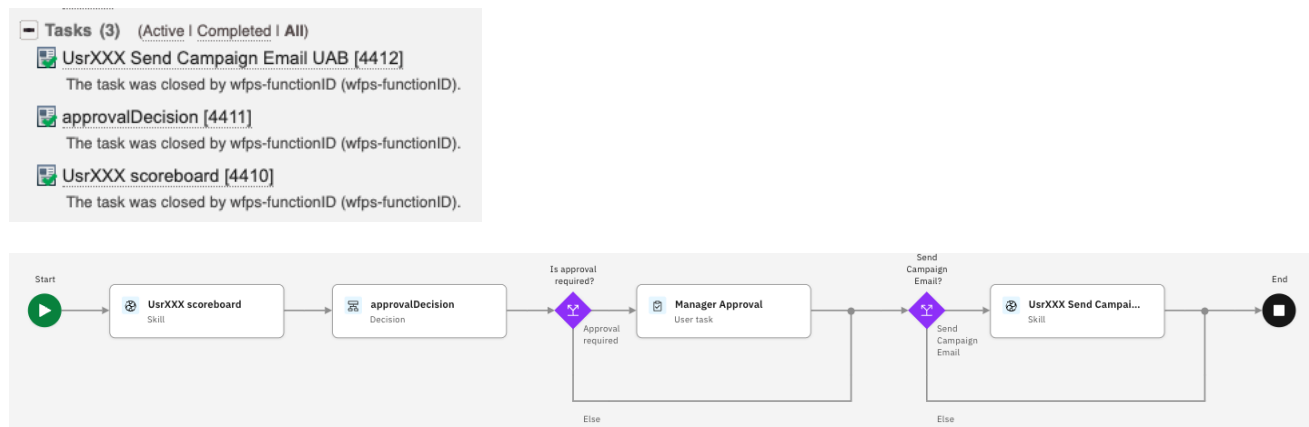
_10. Expand **Data > scoreboard(scoreboardType).**

> Note that the upsell offer was auto approved because the Automation Elite customer is not at high risk of default, and the confidence that the risk is low is high (99.9557).
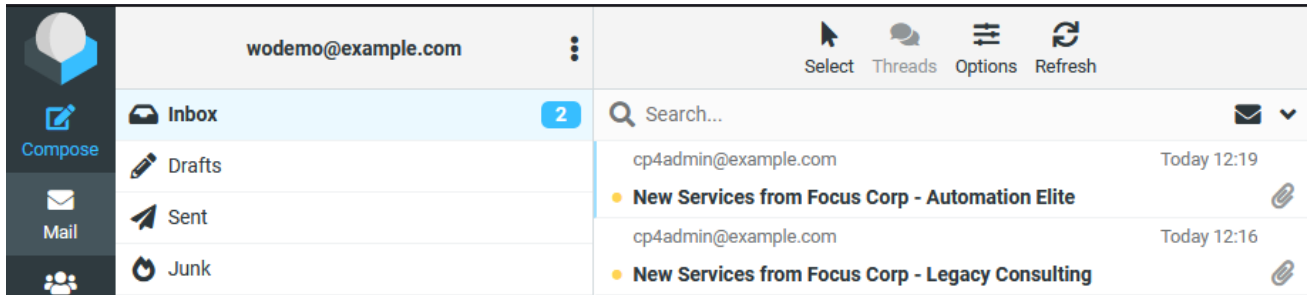
3.10.4 Verify the Upsell Offer Email

_1. Open the **Local Mail Client** in your browser (see 1.3.1 Systems, Lab Files, and Credentials)

_2. Enter your CP4BA Credentials and click the **LOGIN** button.

_3. You should now see two emails.

Note that the emails are customized for each client:
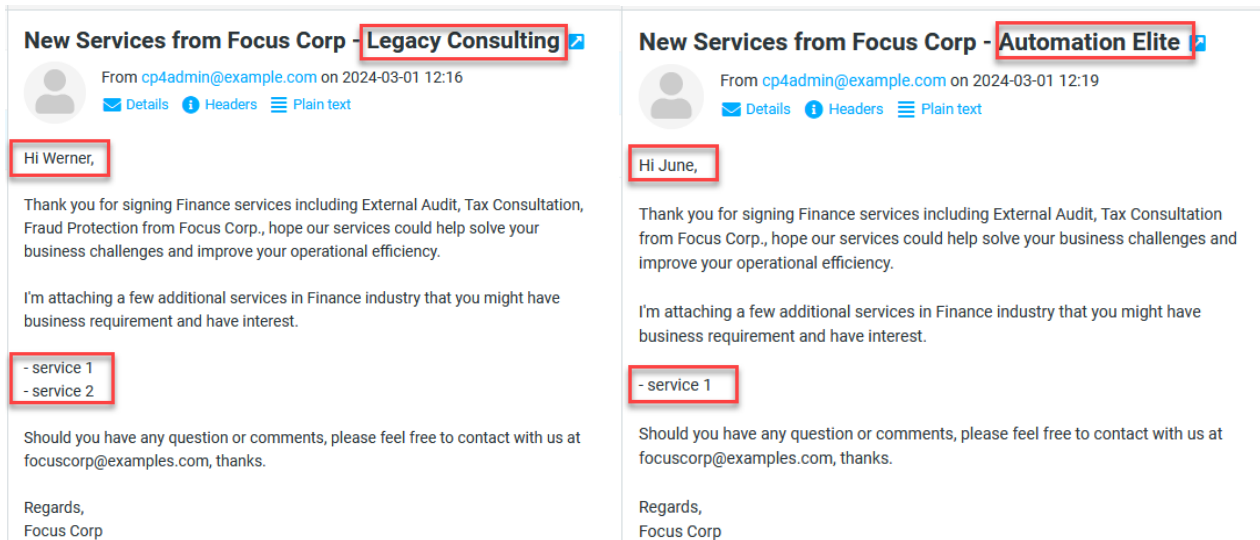
4 Exercise: Expose Workflow as a wxO Skill

4.1 Create an Operation

In this part of the lab, you will create an operation to expose the workflow component in your automation as a wxO skill so others can use it directly in their automations, chats, skill flows, or other Automations.

_1. Click the **Operations** tab.

_2. Click **Create operation +.**

_3. For the *Operation name* enter **UsrXXXSalesCampaignApproval** (where XXX are the digits of your CP4BA Credentials user id). For the *Component*, select **UsrXXXSalesCampaignApprovalWorkflow** and then click **Save**.
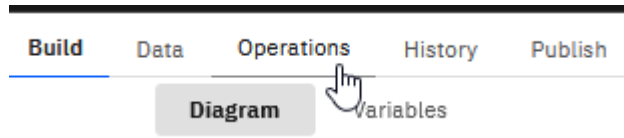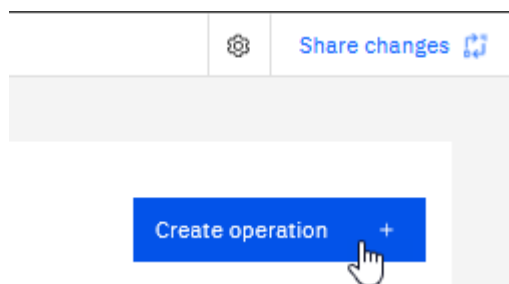

> **Note:** You will encounter an error if the operation name exceeds 31 characters.
>
>
>
> In this case, reduce the prefix (UsrXXXXXXXXXX) length to eliminate the error.

You should now see a new operation.

Name	Component	Description	Created on
UsrXXXSalesCampaignApproval	⤴ UsrXXXSalesCampaignApprovalWorkflow		3/18/2024, 9:57:16 PM

4.2 Create a New Version of the Automation

_1. Click the **History** tab.
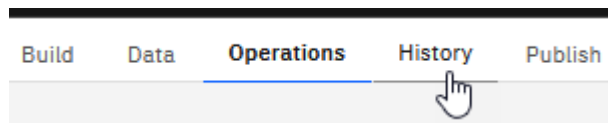
Note that you will see a message that you have local changes in your automation.
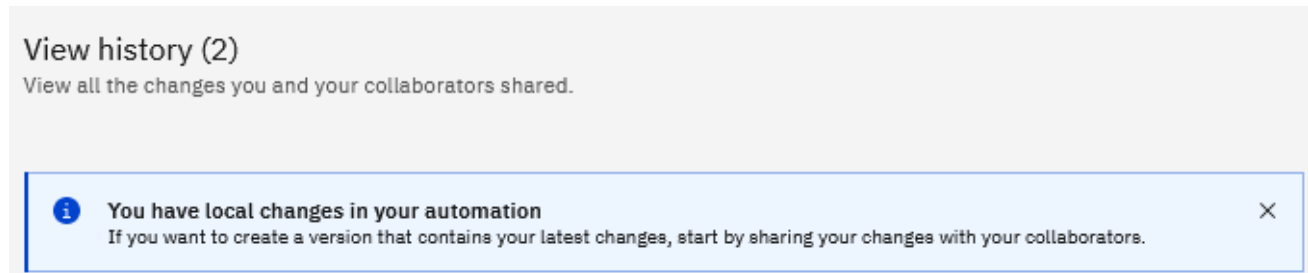
You will need to share your changes to create the latest version that contains your latest changes.

_2. Click **Share changes.**
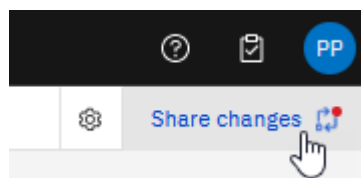
_3. Click the **Share** button.

_4. In the *Share pop-up* Window, click **Share** again.
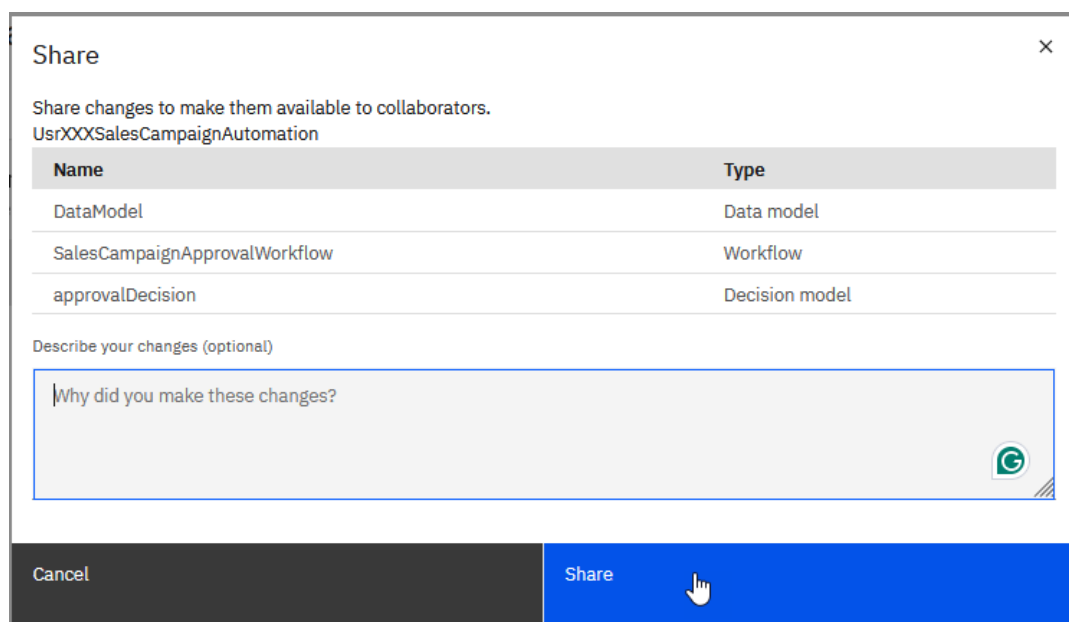
_5. Click the **Back to UsrXXXSalesCampaignAutomation** link.
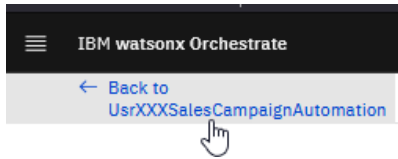
_6. Click the **History** tab.
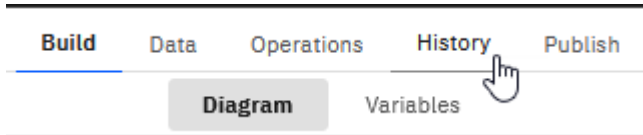
_7. Click **Version +** on the topmost entry to ensure you will use the version with the latest changes!

_8. For *Name,* enter **1.0.0** (if you wonder why this form (see [semantic versioning format](#)) and click **Create**.

_9. You should now see that your latest set of shared changes has been versioned.

4.3 Publish the Automation as a wxO Skill

_1. Click the **Publish** tab

Build	Data	Operations	**History**	Publish
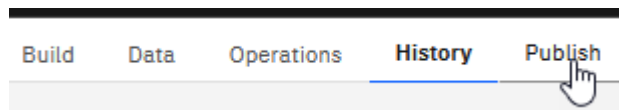
_2. Click the Version **Twisty.**

Version	
⌄	1.0.0

Items per page: 20   ⌄ | 1–1 of 1 items

_3. Click **Publish.**

Version	Shared on	Shared by
^ 1.0.0	3/1/2024, 3:33:38 PM	PAUL PACHOLSKI

Automation	Publish status	
UsrXXXSalesCampaignAutomation	Not published	Publish

Items per page: 20   ⌄ | 1–1 of 1 items
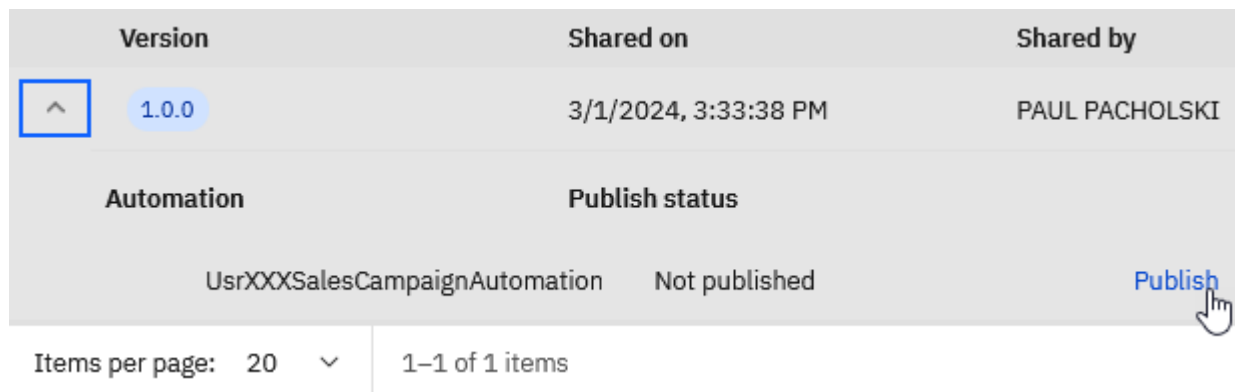
_4. Click the **Publish** button to start publishing the Automation.
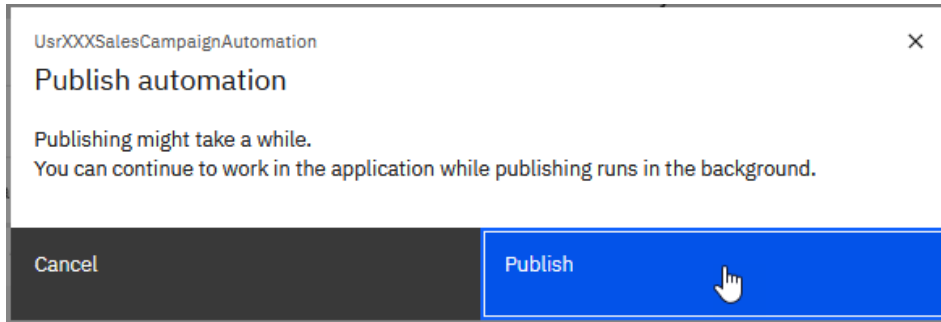
Note:

If you encounter an error message like this:
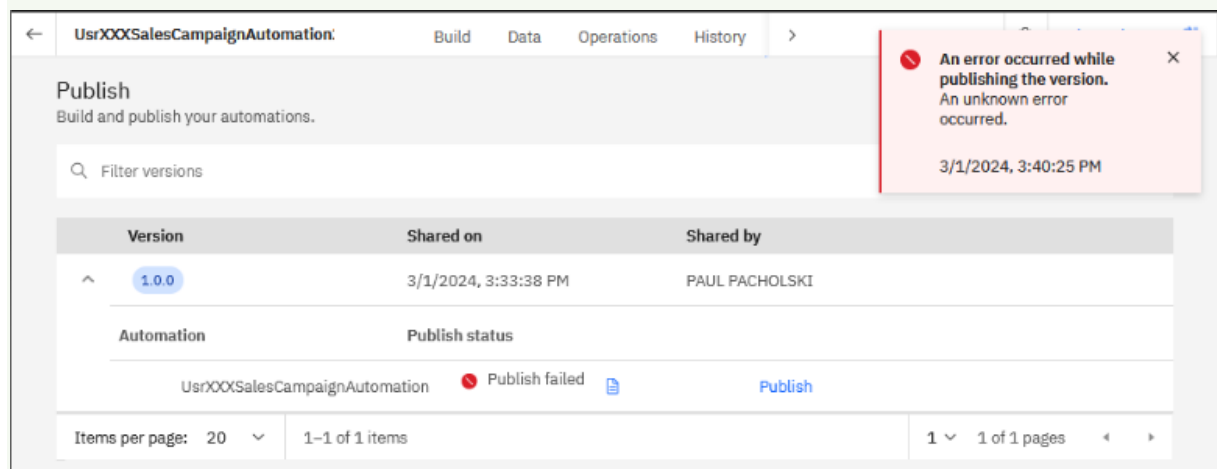
Close the message box and try to publish again.

_5. Wait until the *Publish status* changes from "**Publish in progress**" to "**Published on...**".

Note: the **"document"** link shows the publishing logs, and the "**{...}**" link points to the Swagger UI for the OpenAPI file

4.4 Enhance and Publish the Automation wxO Skill

_1. Click the **Hamburger** menu in the top-left corner.
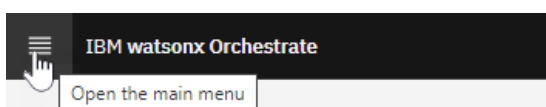
_2. Click **Skills and apps.**
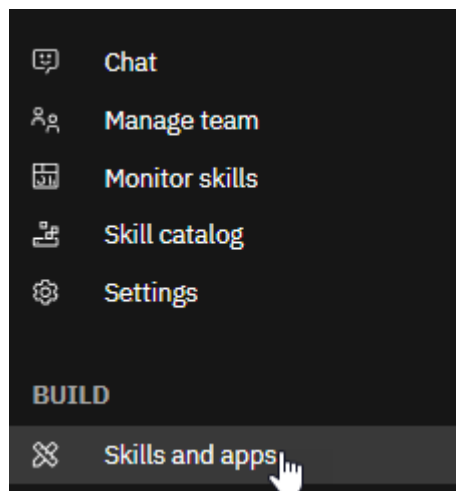
_3. Click the **Skills** tab.
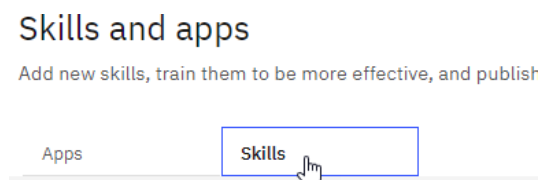
_4. In the Search, enter **UsrXXXSalesCampaignApproval** (remember XXX is your user id) and press **Enter**.
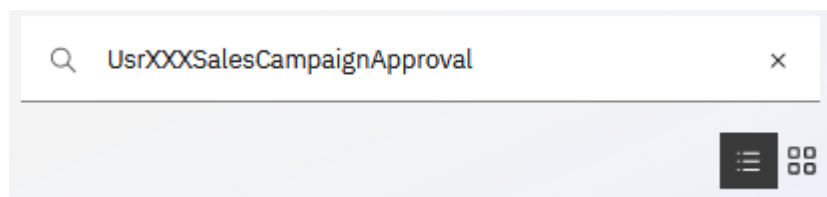
_5. Click the **vertical ellipses (3 dots) ...** and then select **Enhance this Skill.**

_6. Click the **Phrases** tab.

You might want to add a phrase in your local language to make access to this Automation skill easier within the chat.

_7. Enter **start sales campaign approval workflow** phrase.

Phrases are the text your user types in the chat bar to find and use a skill.

| UsrXXXSalesCampaignApproval | ⌫ |

| start sales campaign approval workflow |

Auto-generate phrases (Experimental)

_8. Click **Publish.**

| Cancel | Publish 👆 | Save as draft |
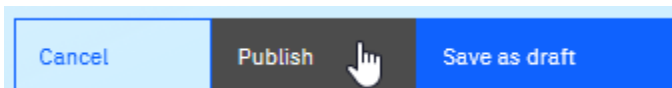
You should see a Published successful message.

✅ Published successful   ✕
Published skill
UsrXXXSalesCampaignApproval.

10:55:10
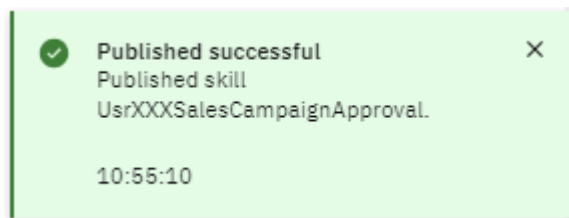
4.5 Add the Automation Skill to Personal Skills

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

☰ IBM watsonx Orchestrate
👆 Open the main menu

_2. Click **Chat.**

🗨 Chat 👆
👥 Manage team
🗂 Monitor skills

_3. Click **Add skills from the catalog** tile.

Add skills from the
catalog

👆

→
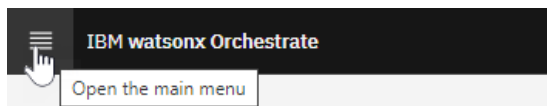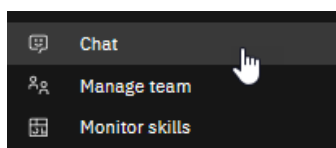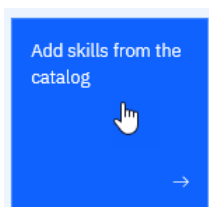
_4. To find the skill in the Skill Calaog type **UsrXXXSalesCampaignAutomation** (remember XXX is your user id) in the *search bar* and hit **enter key**.
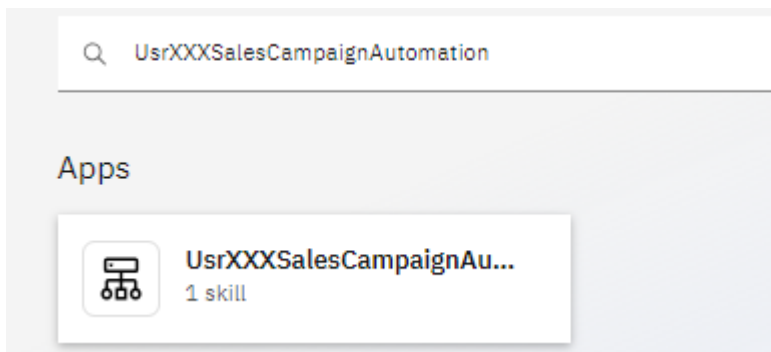
_5. Click the **UsrXXXSalesCampaignAutomation** App (remember XXX is your user id).
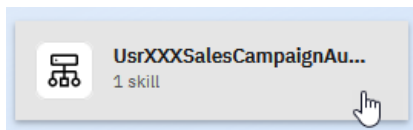
_6. Click **Add skill +**
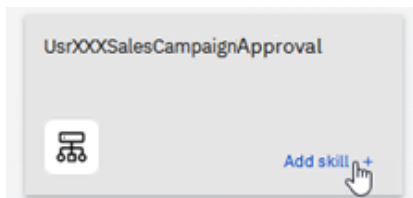
You should now see the **"Added"** status.

5 Exercise: Test the Workflow Skill in the Chat

5.1 Test the Auto Approval Case

_1. Click the **Hamburger** menu in the top-left corner.
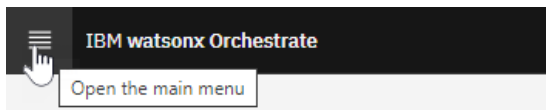
_2. Click **Chat**
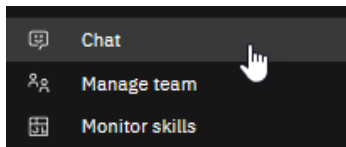
_3. In the *Chat*, type "**start**" and select the suggestion "**start sales campaign approval workflow".**

_4. Click the **Send** button to start the Workflow.

> Note that we are using a natural language interface to start the Workflow. Alternatively, we could click the Workflow.
>
>

_5. Enter input data and start the Workflow skill:

1) For *clientName* enter **Automation Elite**
2) Click the *upsellServices* **Twistie**
3) Click **Add +**
4) For *upsellService enter* **service 1,**
5) Click **Apply**

You just need to complete this form first.

UsrXXXSalesCampaignWorflow

clientName

Automation Elite ⬤1

upsellServices ⬤2 ⌃

service 1 ⬤4 🗑

Add + ⬤3

Cancel   Apply

Note: if you see this retry message, click the **Retry** button.

The UsrXXXSalesCampaignApproval skill can't be used right now. Retry your request later.

Show more

Retry   Cancel

You should see a confirmation message.

Working on getting the results might take a while. You can continue waiting or you can ask for something else to be done.

_6. Wait until the Workflow is completed.

_7. When the Workflow is complete, you will see the workflow skill output message.

Note:

1) The sales campaign upsell offer was approved by the decision service.
2) It was not auto-rejected by the decision service.
3) Note the explanation provided by the decision service telling us why the upsell offer was auto-approved
4) The customer name
5) Shows the confirmation from the email service in the Workflow that an email upsell email was sent to the customer.

5.2 Test the Approval Required Case

_1.   In the *Chat*, type "**start"** and select the suggestion "**start sales campaign approval workflow".**

_2. Click the **Send** button to start the Workflow.
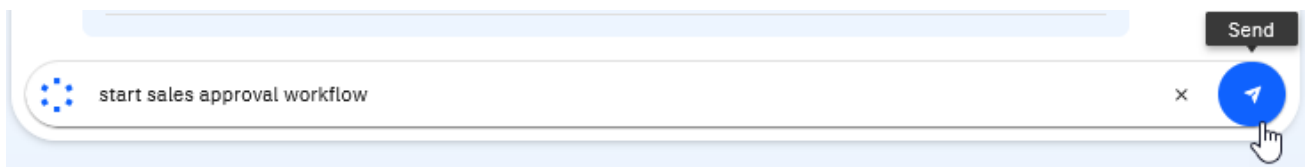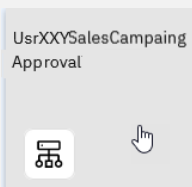
_3. For *clientName* enter **Legacy Consulting**, add *upsellService* **service 1,** and then click **Apply.**

You just need to complete this form first.

UsrXXXSalesCampaignWorflow

clientName

Legacy Consulting

upsellServices ^

service 1 🗑

Add +

Cancel | Apply

You should see a confirmation message.

Working on getting the results might take a while. You can continue waiting or you can ask for something else to be done.

_4. Click the **Messages Icon** in the top right corner of the chat window and expand **Task in progress (1).**

Task in progress (1) ^

start sales campaign approval workflow
3/26/2024, 11:04:18 AM

Results (2) ˅

Note that the message indicates that the skill is in progress. This status will persist until we complete the Workflow that implements this skill. Specifically, we need to complete the Manager Approval task created for this customer.

_5. Click the **Tasks icon** in the top-right corner.

Tasks

_6. To complete the Task, click **Manager Approval** in the Task name column.

Q Search for tasks...

Filter by: (🕐 On track (159)) (✅ Completed (13)) | (Claimed) (Unclaimed)   Email notification ⬤ ▽

Task name	Priority	Status	Due on	Created on	Owner	
Manager Approval	Medium	🕐 On track		Mar 26, 2024, 11:05 AM	P	✏

_7. Select the **approved checkbox** and add to *upsellServices* **service 2.**

_8. Scroll to the bottom of the form and click **Submit.**

_9. Click the **Hamburger** menu in the top-left corner.

_10. Click **Chat.**

_11. Click the **Messages Icon** in the top right corner of the chat window and expand **Results (1),** then click the latest run at the top of **start sales campaign approval workflow** task.
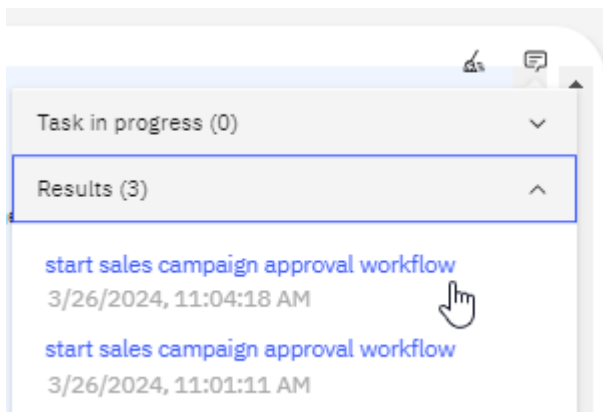
In the Chat, note the output of the approval process, including **service 2** added by the approver.

Note:

1) The sales manager approved the upsell offer.
2) It was not auto-rejected by the decision service.
3) Note the explanation provided by the decision service telling us why the upsell offer was not auto-approved or auto-rejected but required a manager's approval.
4) The customer name
5) The sales manager added the upsell service "service 2".
6) Shows the confirmation from the email service in the Workflow that an email upsell email was sent to the customer.

Congratulations on completing the lab!

NOTICES

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or Service is not intended to state or imply that only that IBM product, program, or Service may be used. Any functionally equivalent product, program, or Service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or Service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

> IBM Director of Licensing
> IBM Corporation
> North Castle Drive, MD-NC119
> Armonk, NY 10504-1785
> United States of America

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

TRADEMARKS

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., 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.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

ITIL is a Registered Trade Mark of AXELOS Limited.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the US and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

© Copyright International Business Machines Corporation 2020.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.