Watsonx Orchestrate Lab 3: Decision Automation and Skill-based Actions

Use case

In this lab we show how to work with automations and specifically how to implement a decision automation that can accomplish a task based on certain criteria. For example, if you want to approve a loan application depending on the credit score and income of the applicant, a rule-based decision to accomplish this will be very helpful. This decision can be used as a skill in IBM watsonx Orchestrate and can be made available as a skill-based action in your assistant.

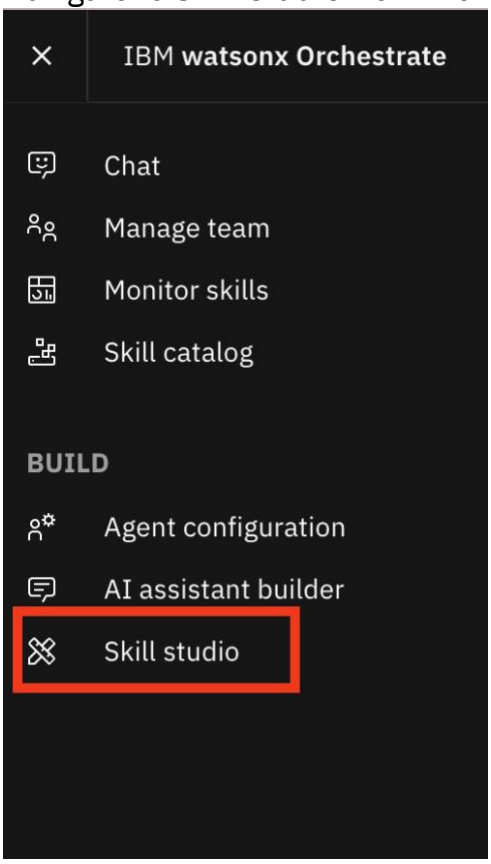
Assumptions

See lab **README** file for details

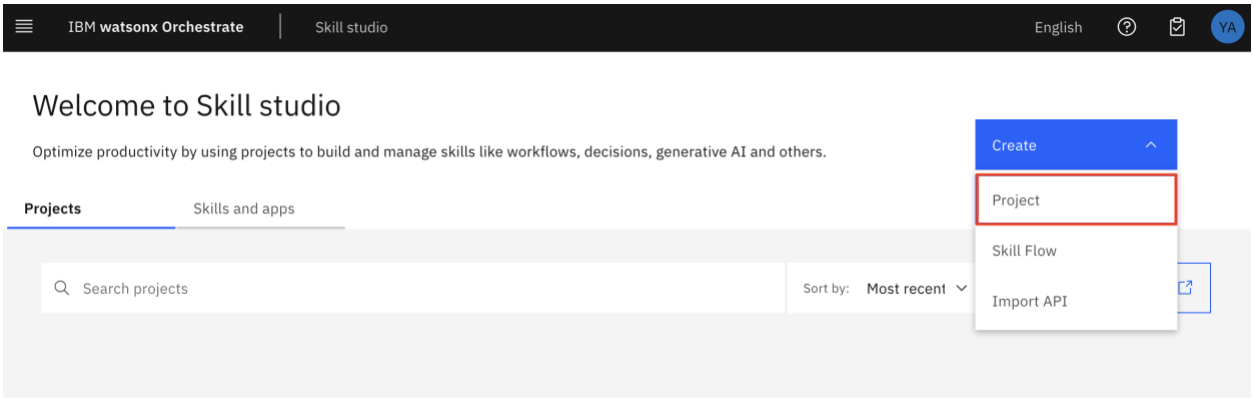
1. You have access to IBM watsonx Orchestrate (Standard Edition, which includes Automation/Projects in Skill Studio)
2. You are authorized to create automations in Skill Studio

Step 1: Create a decision automation

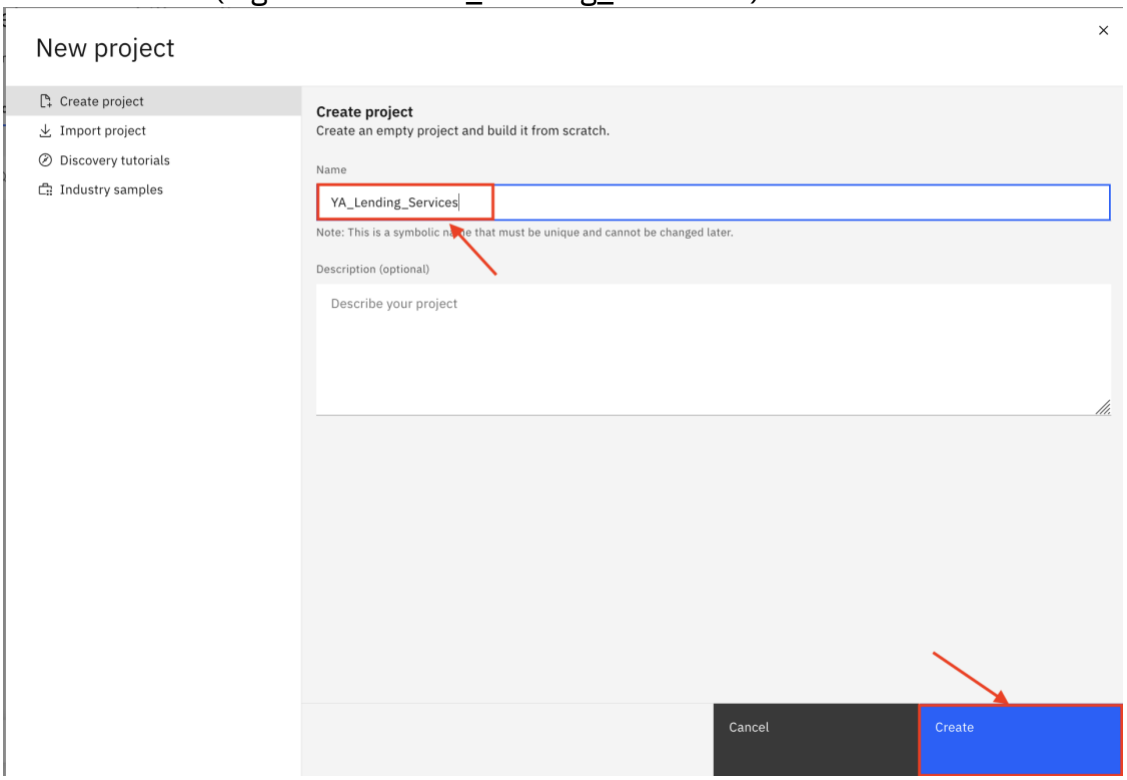
1. Navigate to **Skill studio** from the menu:



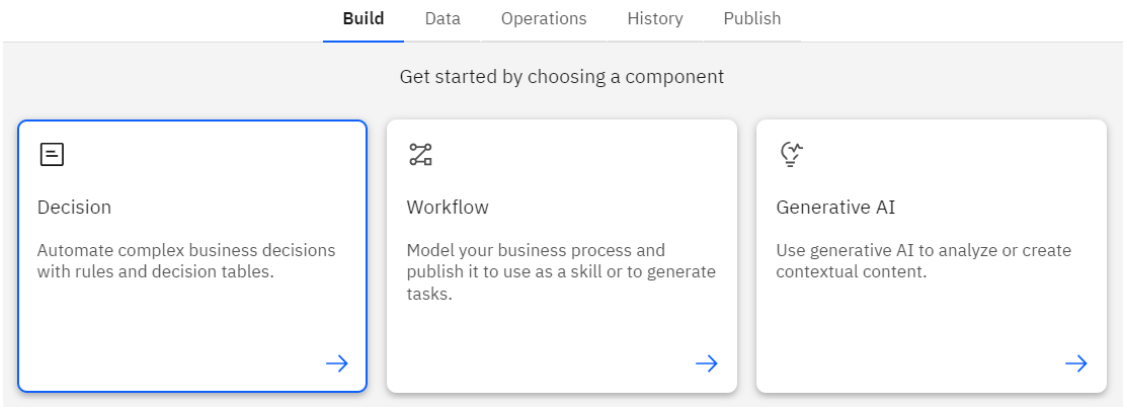
2. Click on the **Create** dropdown and select **Project** to create a new automation:



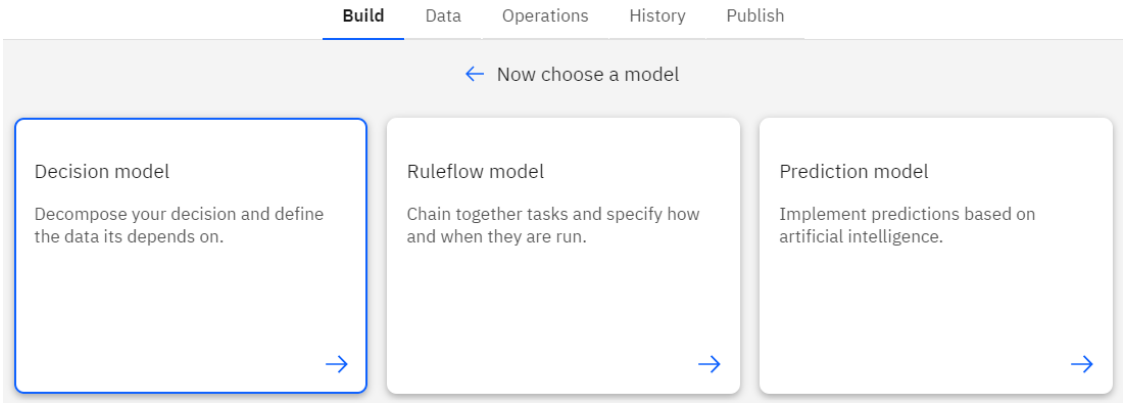
3. Give it a name (e.g. “YourInitials_Lending_Services”) and click **Create**:



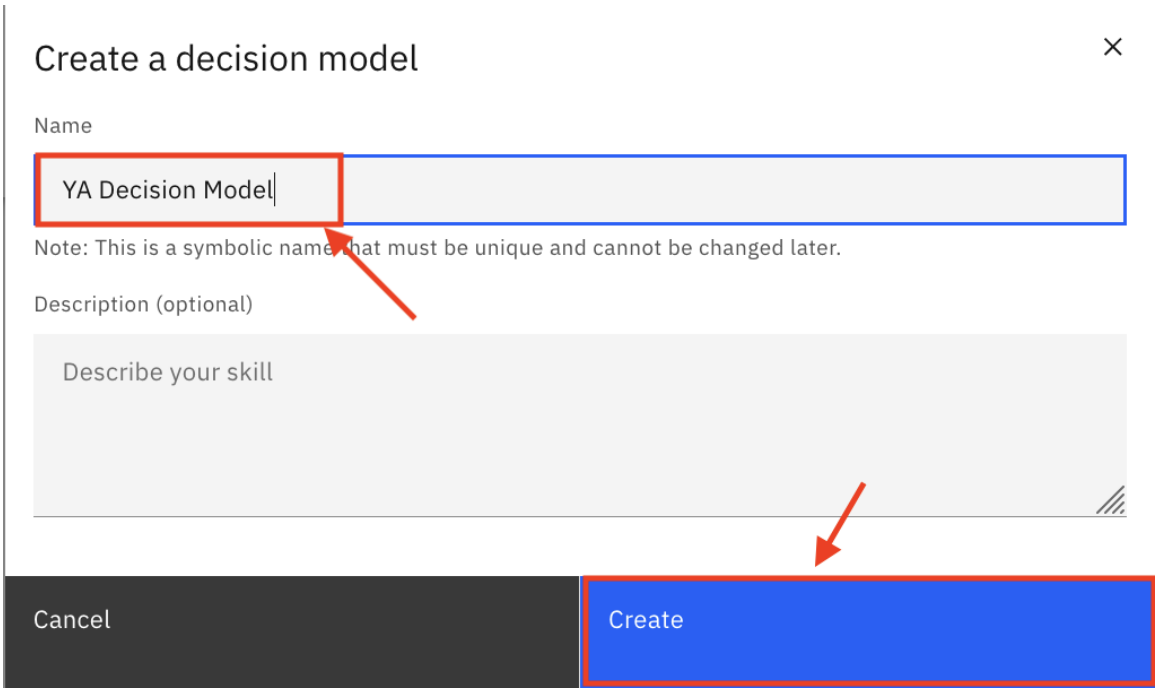
4. Next select **Decision** tile.



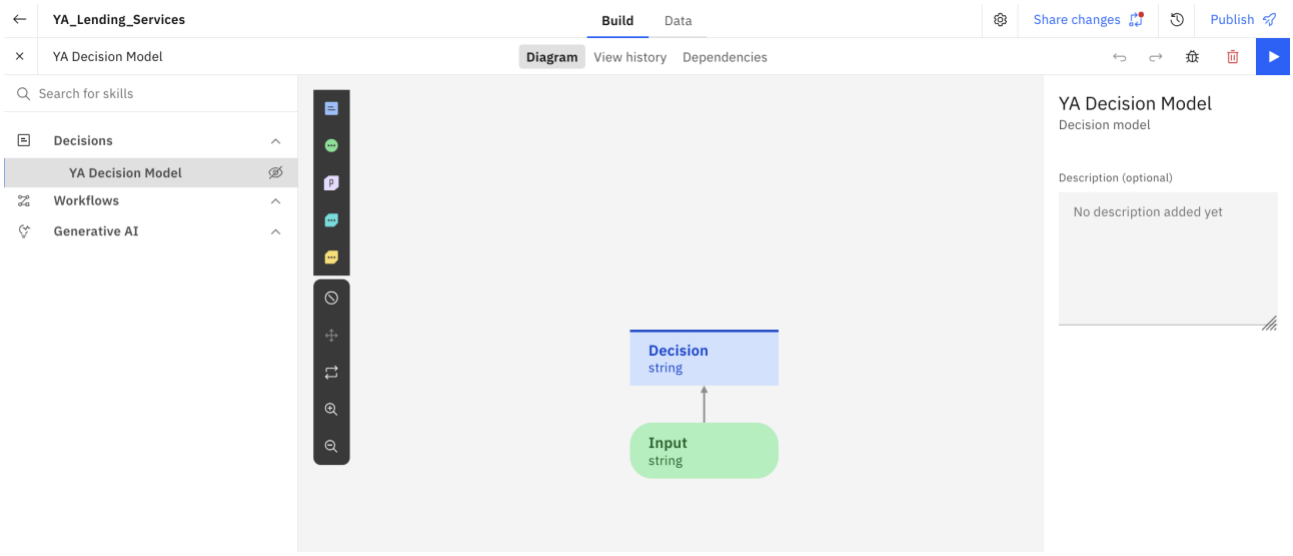
5. Select **Decision model** as the model for your decision.



6. Give a name to your decision (e.g., “YourInitials Personal Loan”) and click **Create:**



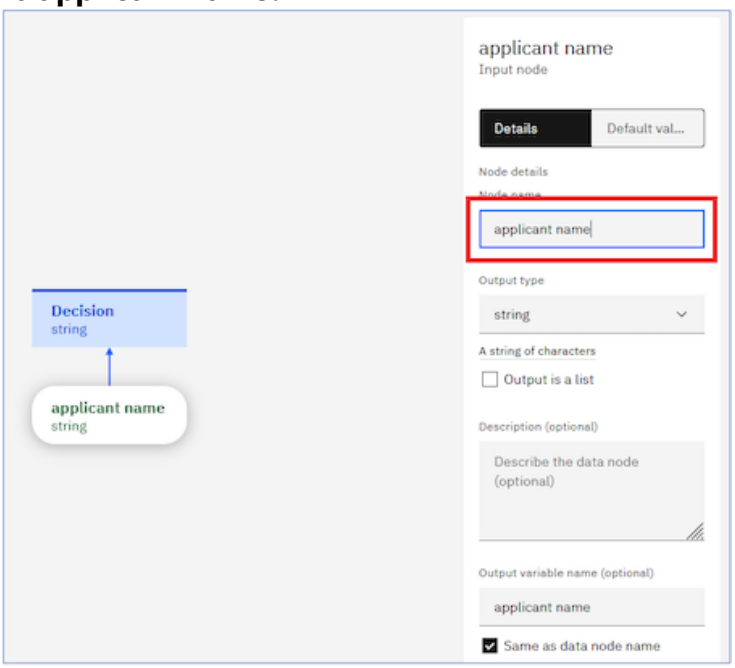
7. The **green nodes** (data nodes) represent data elements that are used by the decision’s rules. The **blue nodes** (decision nodes) represent a step in the decision. They contain rules that will execute to achieve that step. Each decision node outputs a partial decision. In complex decisions, there are many decision nodes, and the output of one decision node will flow as input into another decision node:



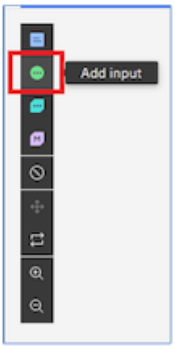
Step 1.1: Create the Data nodes

Now, let's create the data nodes.

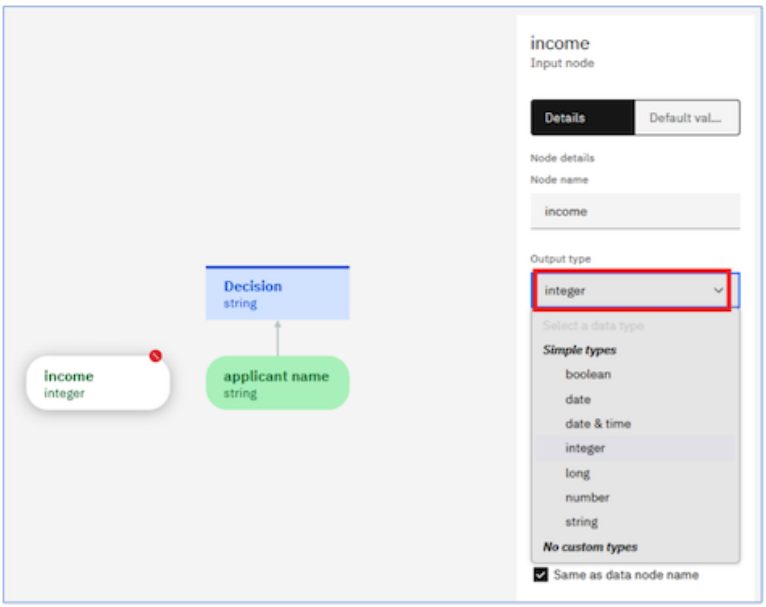
1. Click on the data node and on the right-hand side, change the name of the node to **applicant name**:



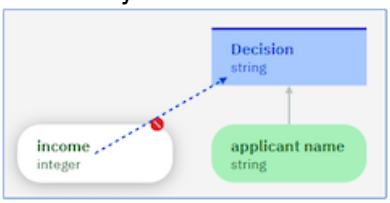
2. Create a new data node by clicking on the **Add input** button in the palette:



3. Select the newly created node, change the name of the node to **income**, and change the type of the node to **integer**:



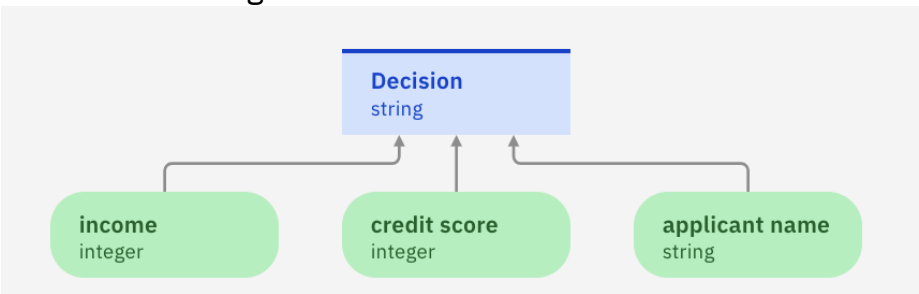
4. Hover on the node and click the **Connect to another node** button, drag the connector to the **decision node** and click on the **decision node**.
Note: If you **Add Node** from the decision node, it is automatically connected.



5. Reapeat the steps 1-4 in this section to add additional nodes. They are:

Node name	Node type
income	integer
credit score	integer

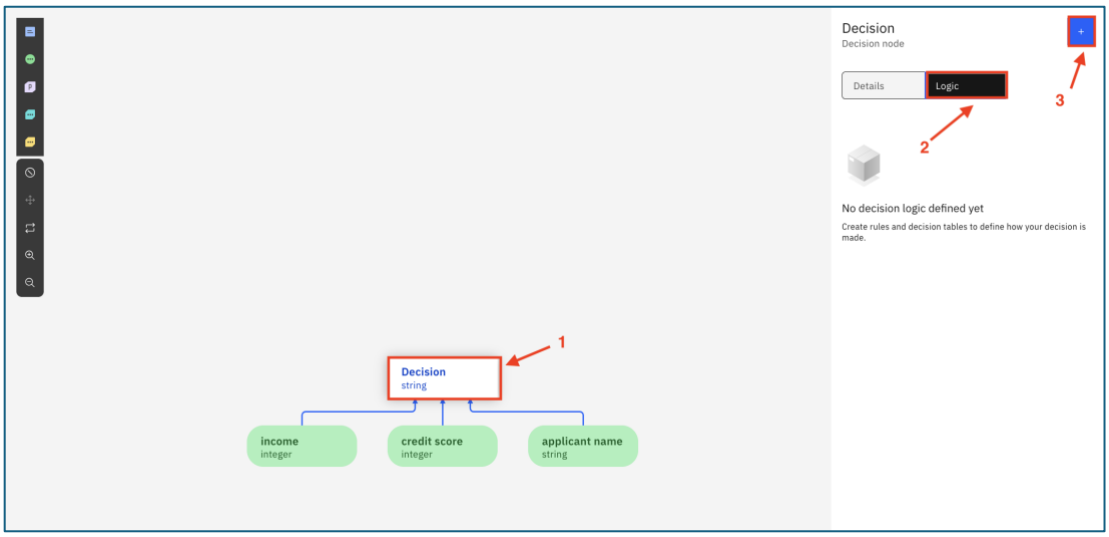
6. Your decision diagram should now look like this:



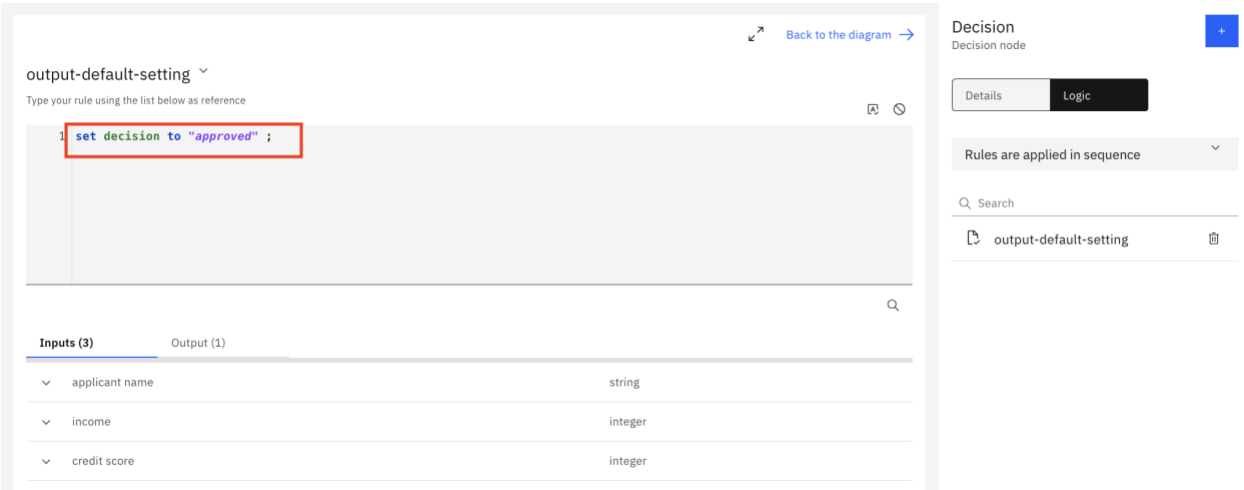
Note: *this is an example created for the purposes of our hands-on lab. In a real business scenario, you would use additional parameters e.g. SSN, loan purpose, whether an applicant is employed, etc. and you could create additional rules.*

Step 1.2: Create the Rules

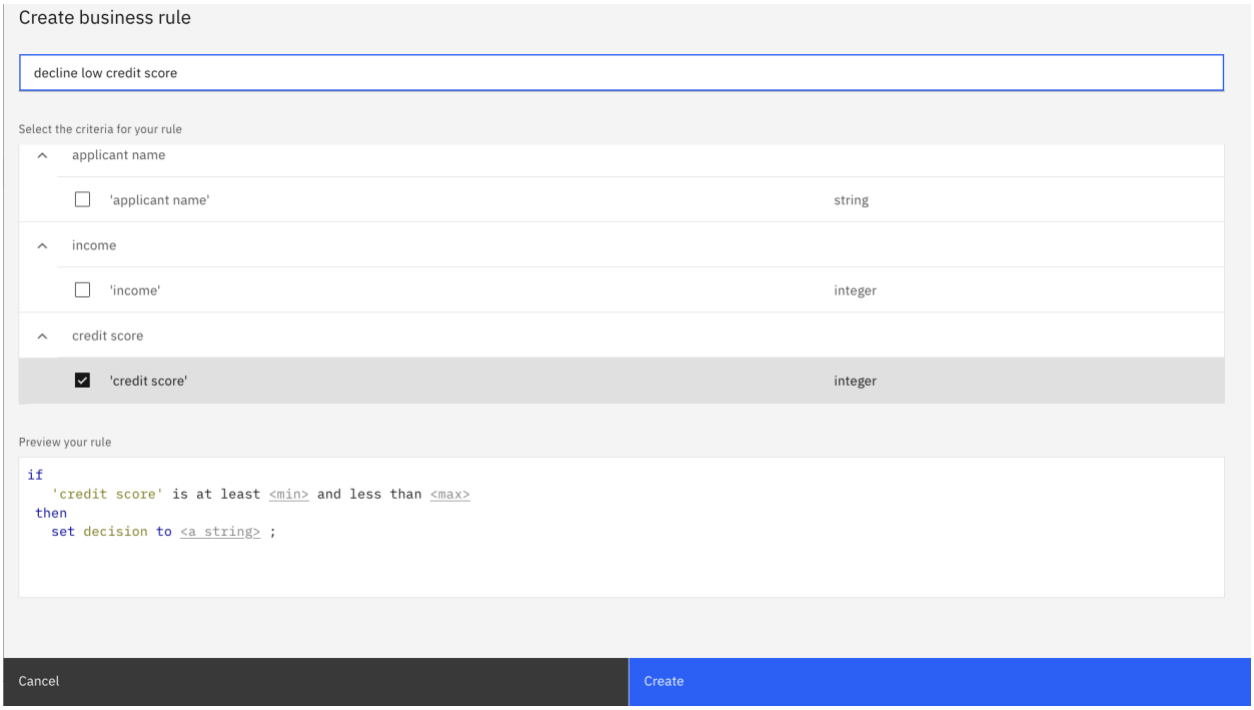
1. You will now add some rules to the decision node. The first rule that you will add is a **default rule**. It will initialize the outcome of the decision to approved. The other rules will then specify the conditions under which the loan is declined. To create the default rule, click on the **decision node**, click the **+** button and select **Default rule**.



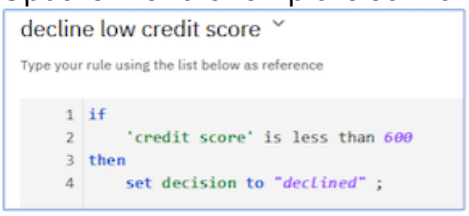
2. Click on the **string placeholder**, select **string** in the dropdown, and enter **approved**. You just created your first rule!



3. Now, let's add another one to state that any applicant with a credit score under 600 will be declined. Click again on the **+** sign and select **Business rule**. Give your rule a name, **decline low credit score,** select **credit score** in the criteria choices and click **create**.



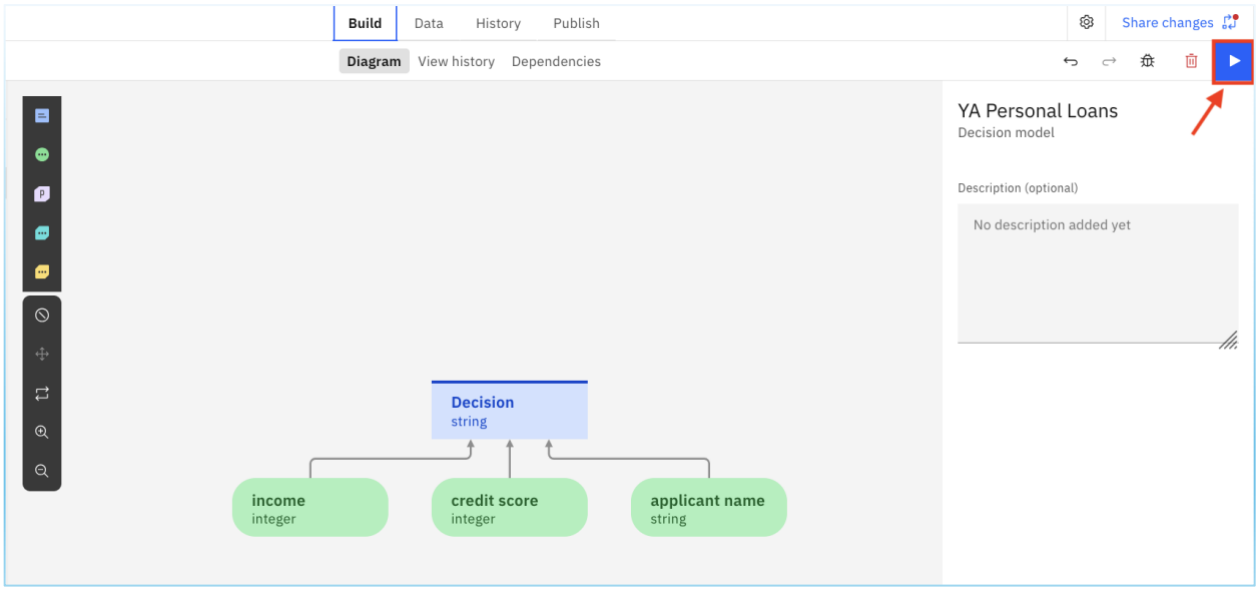
4. Update the rule template so that your rule looks like this:

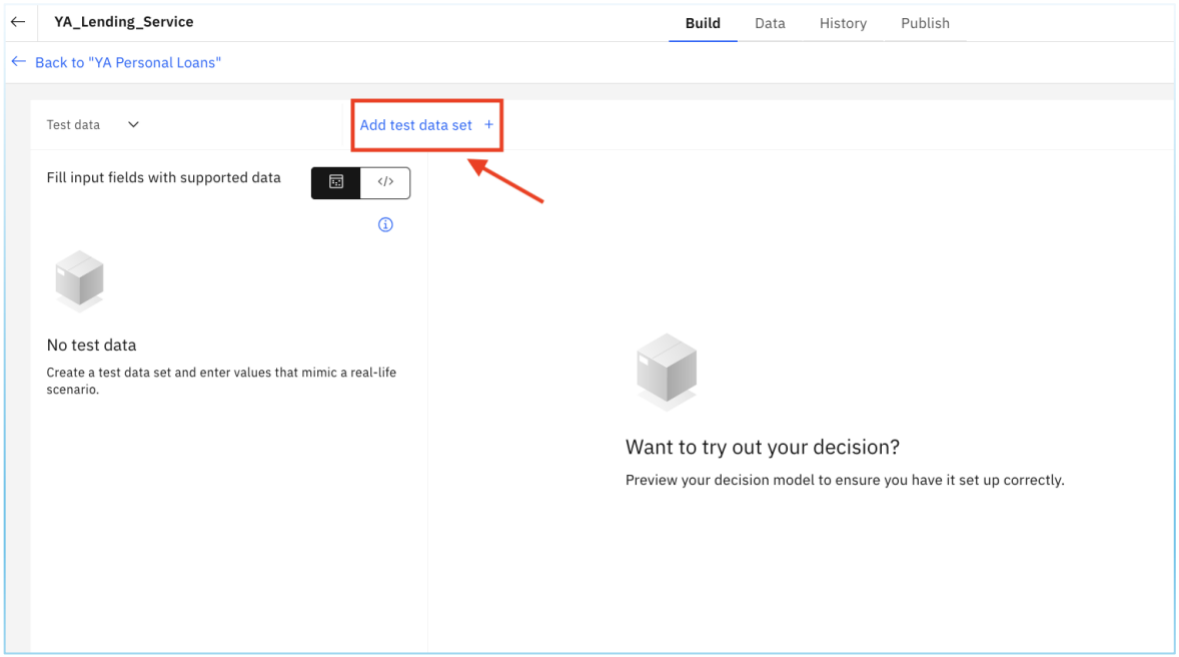


5. Add another rule to decline if income is below a certain threshold (e.g. 50,000):

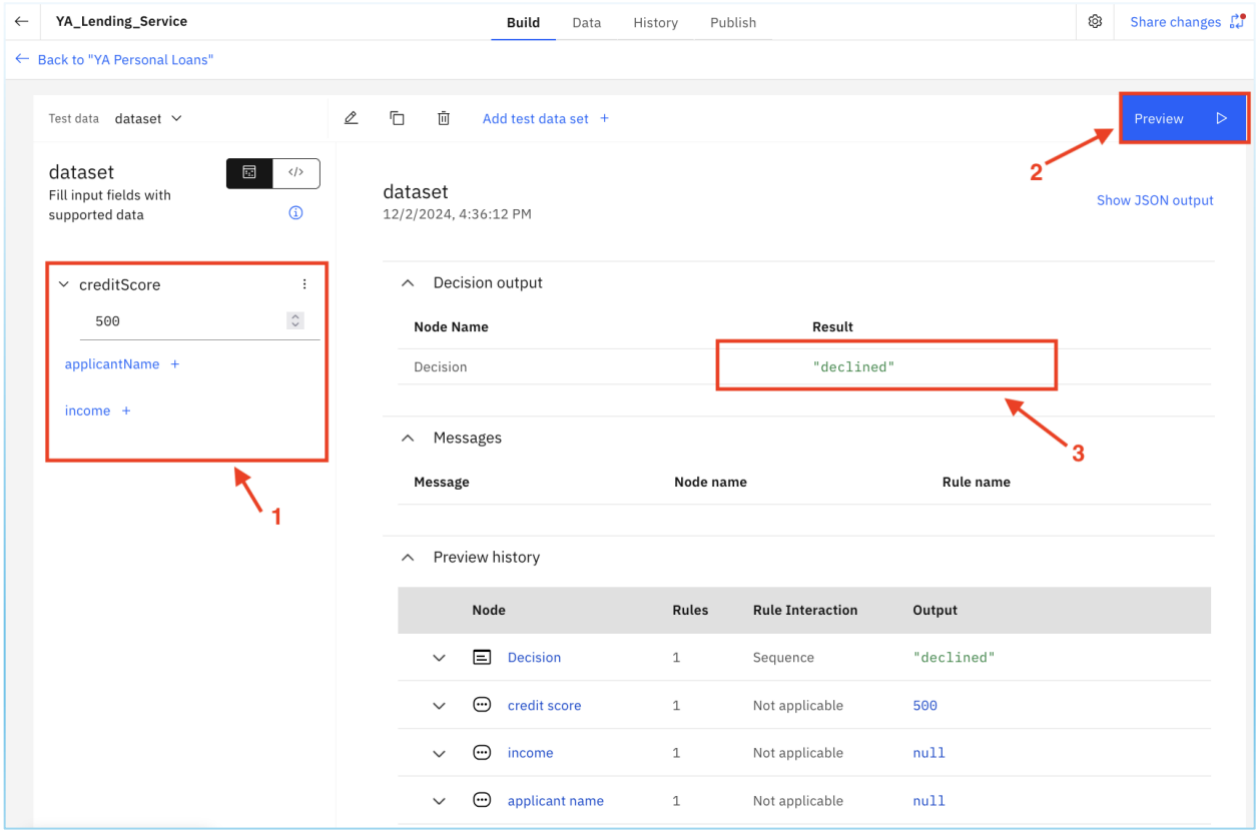
Step 1.3: Test the Decision

1. You can test your rule by clicking on the **Preview** button and then selecting **Add test data set:**



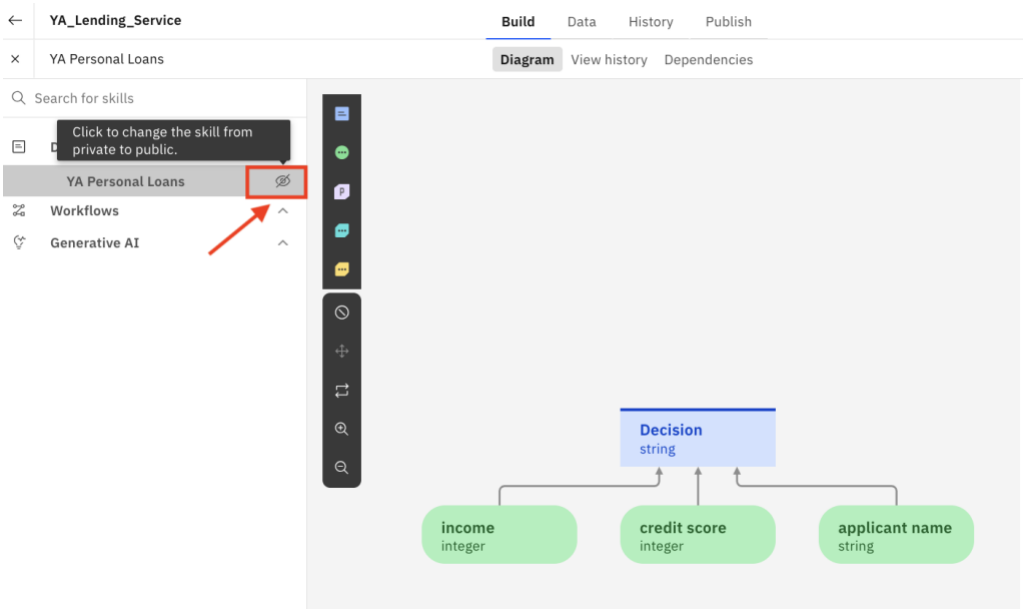


2. Enter data for the input fields and click the Preview button to see the outcome of the decision. For example, enter a value of **500** for the credit score, and you should get a declined output:

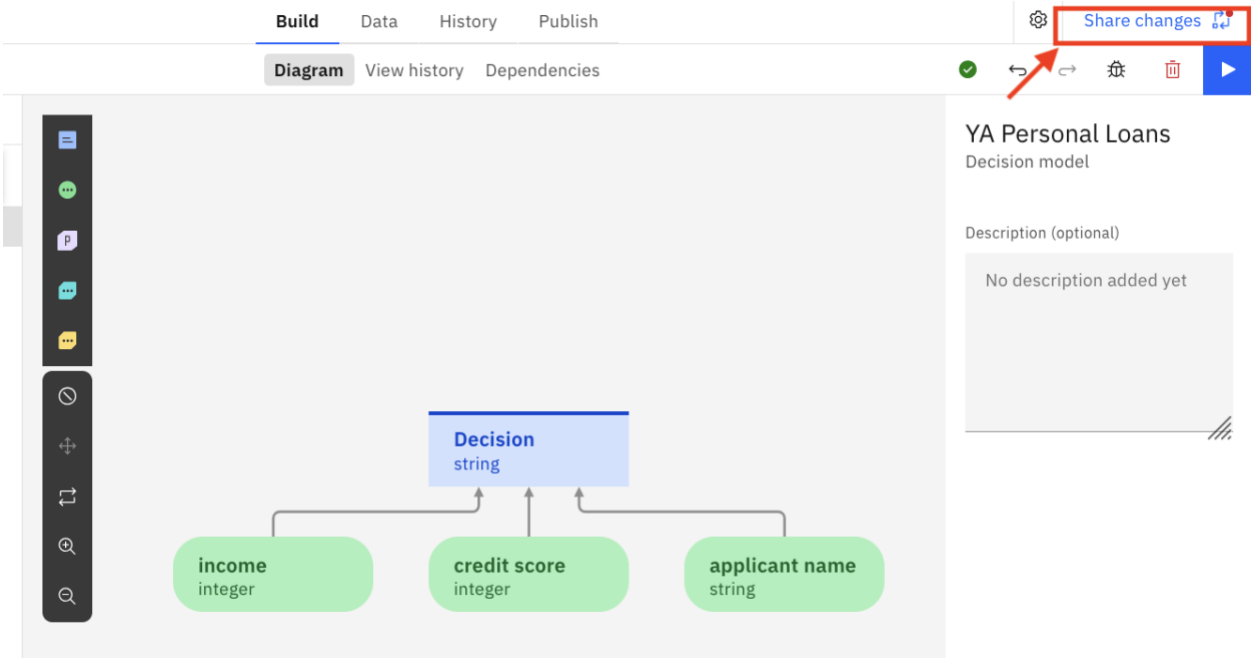


Step 1.4: Deploy the decision service

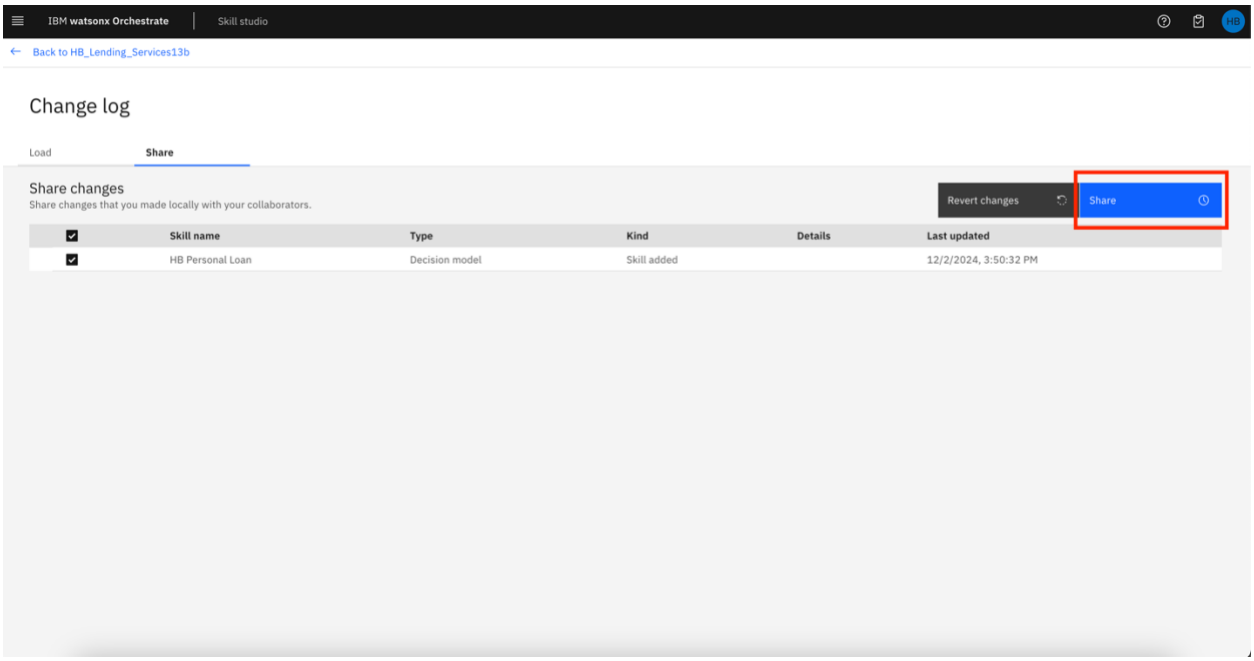
1. Go back into our decision model and click on the crossed out eye icon beside your decision model name, to make your skill public.

2. 

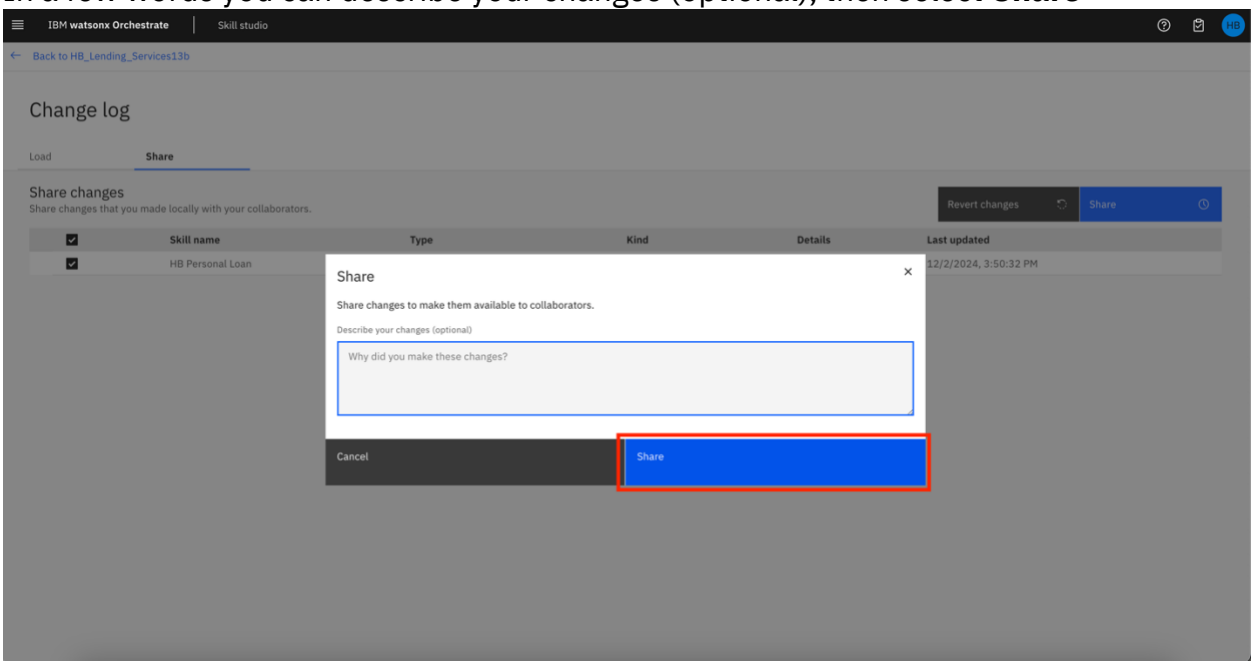
3. Share your changes, by clicking on the **Share changes** button.



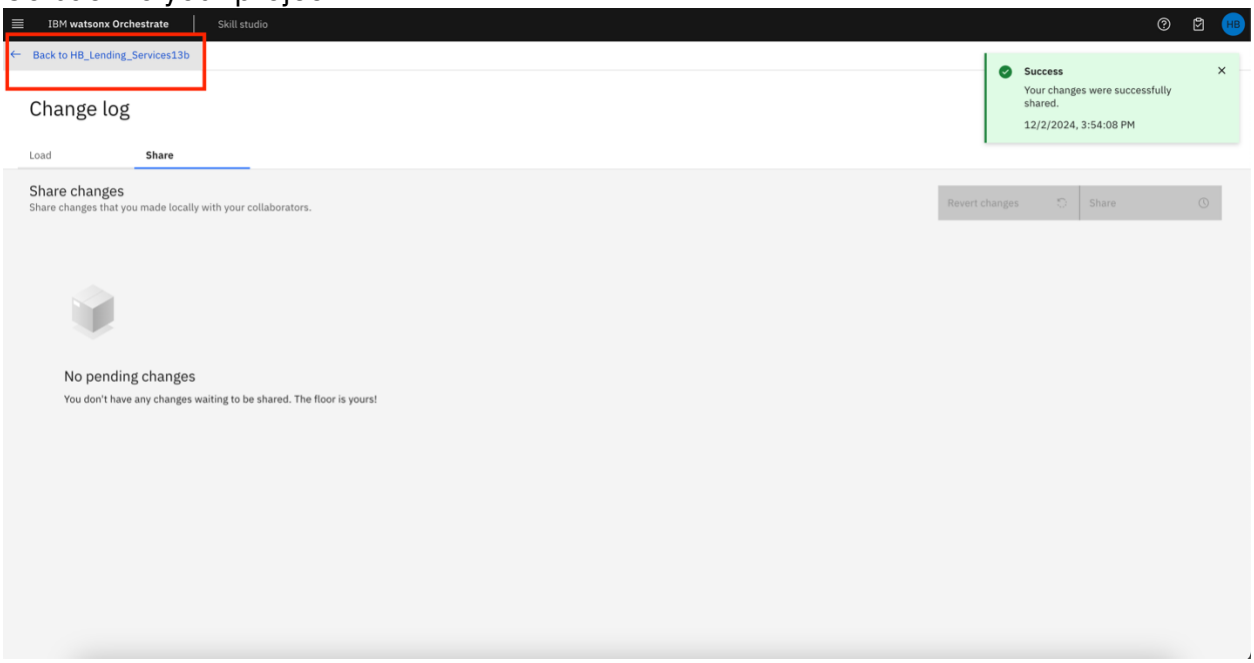
4. Click on **Share**



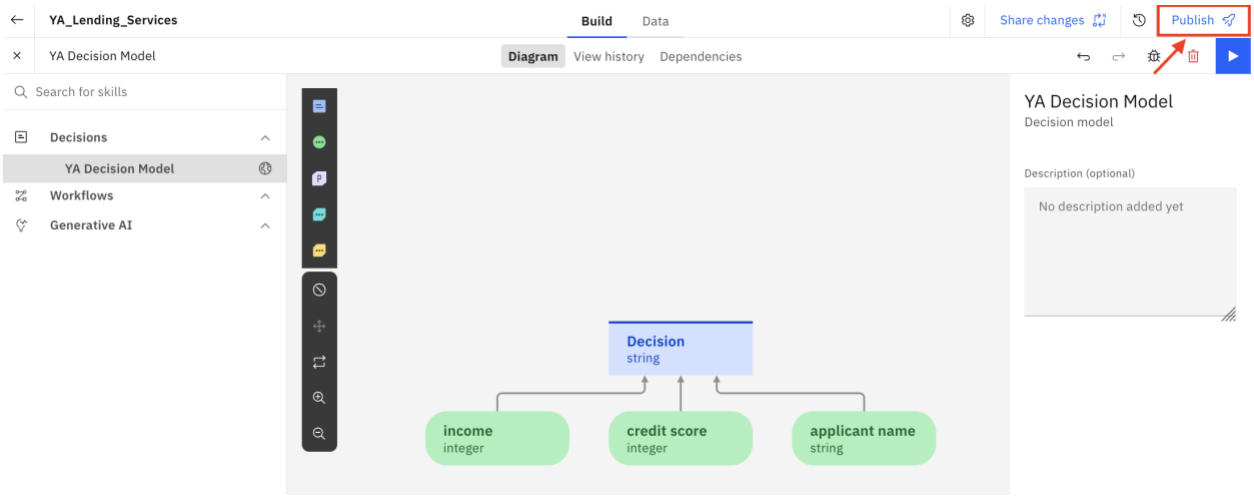
5. In a few words you can describe your changes (optional), then select **Share**



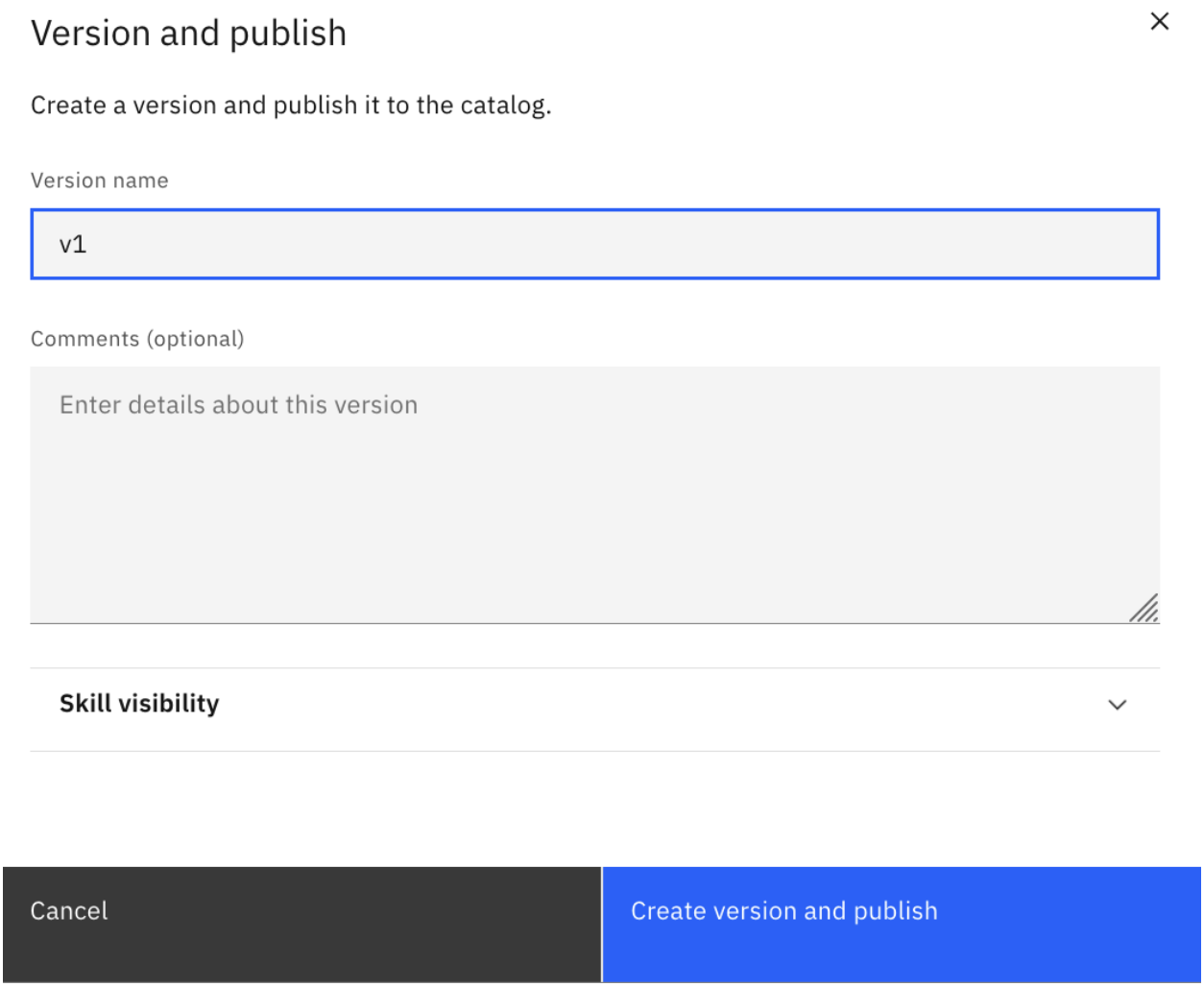
6. Go back to your project



7. Click on **Publish** in the top right corner:



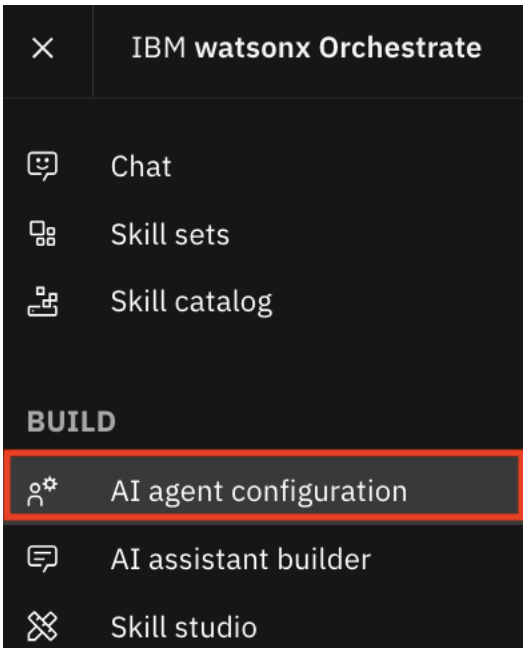
8. Provide a version name and click on **Create version and publish:**



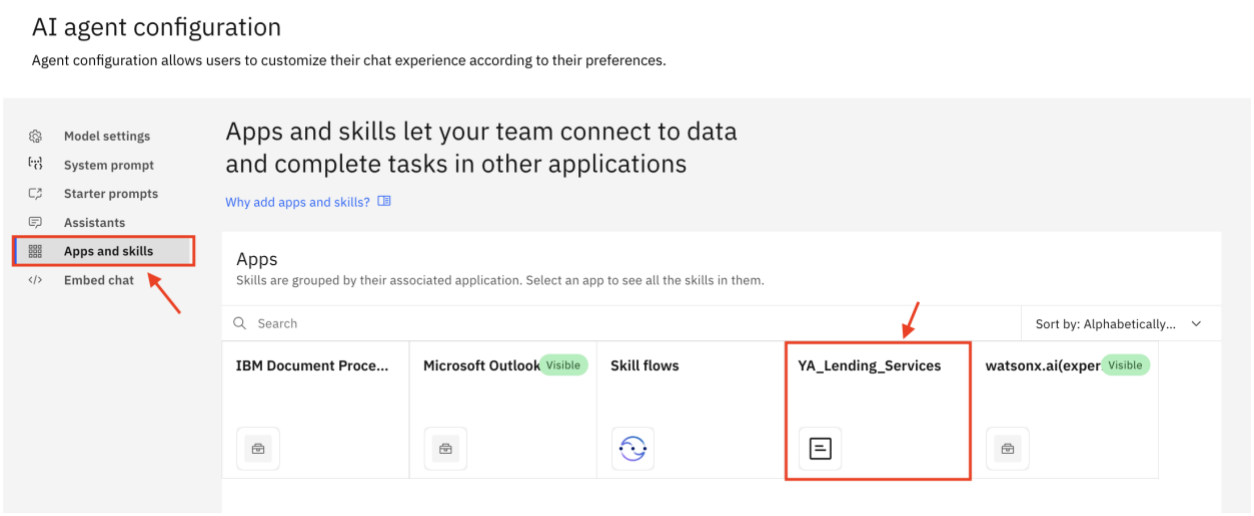
9. Your decision service is now published and operational

Step 1.5: Test Decision Flow in AI Chat

1. Navigate to **AI agent configuration** from the hamburger menu:

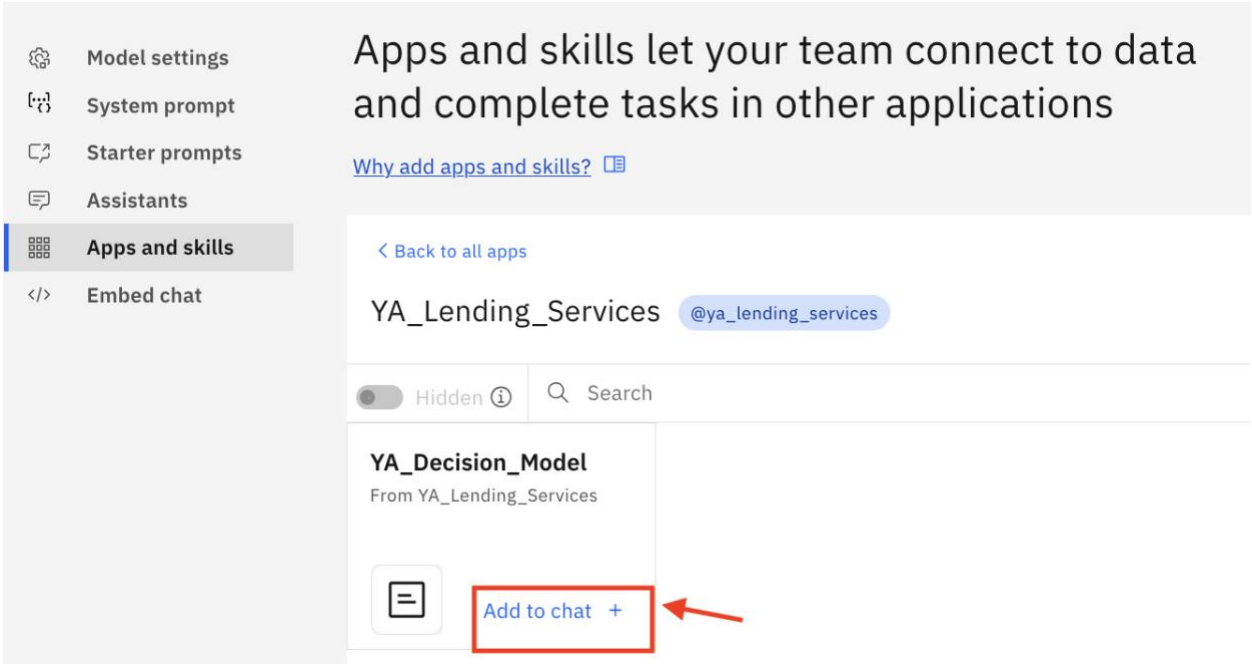


2. Select **Apps and skills,** then click on the tile with your decision application (**YourInitials_Lending_Services**):

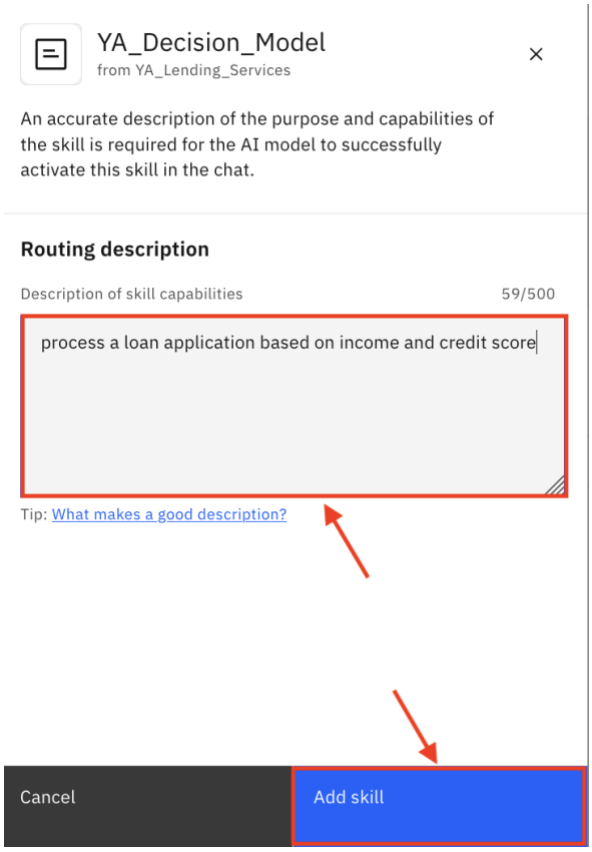


Note: Note that your application is automatically available under **Apps and skills** in the AI Agent configuration. This is true for any published automation-based skill. If you need to add a built-in skill or custom skill you built, you will need to connect to the application from the **Main Menu -> Skill sets** first (if you are working in wxO in IBM Cloud). If you are on an AWS tenant, go under Manage Team -> Skill sets.

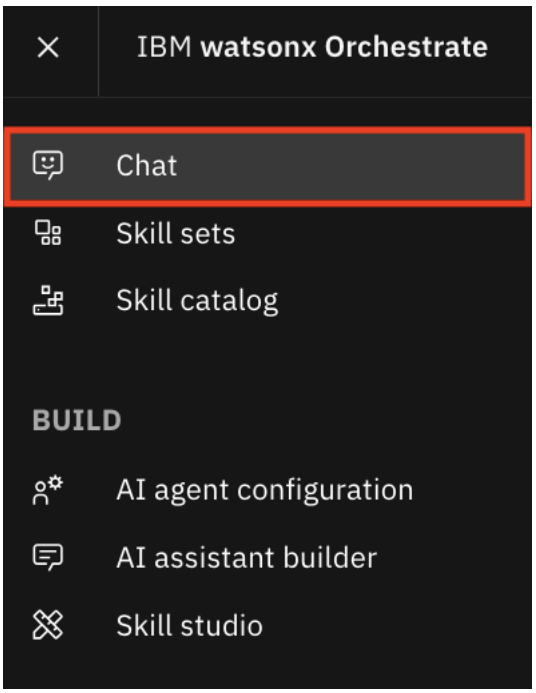
3. Click on **Add to chat+** to add the skill to the AI chat:



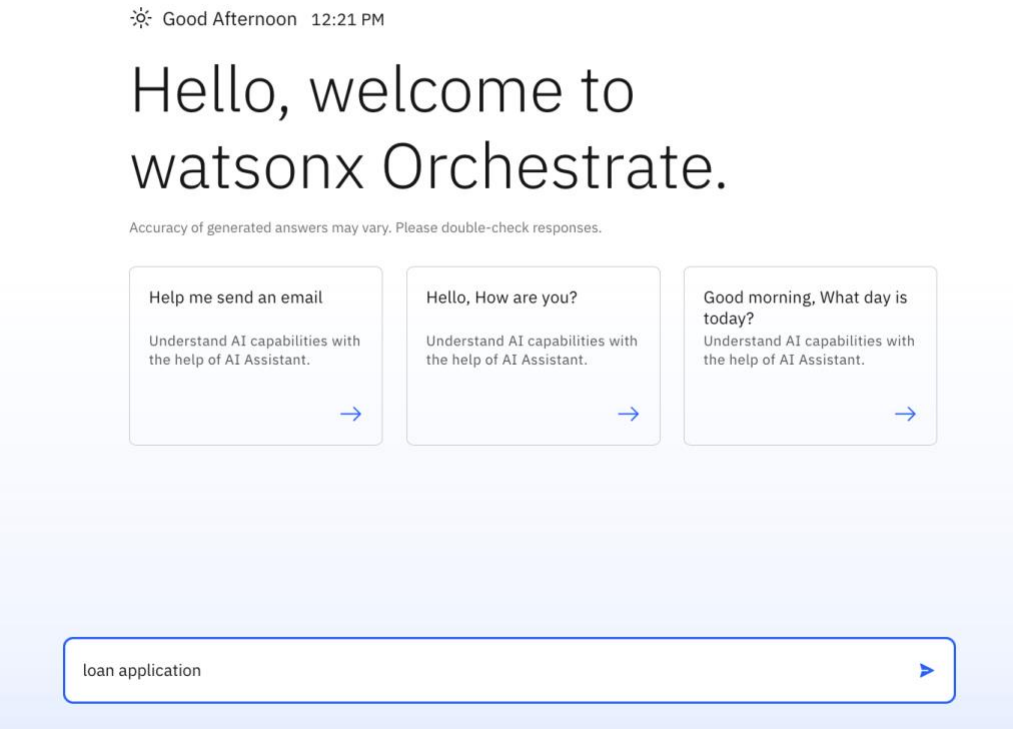
4. Enter a routing description and click on **Add skill:**



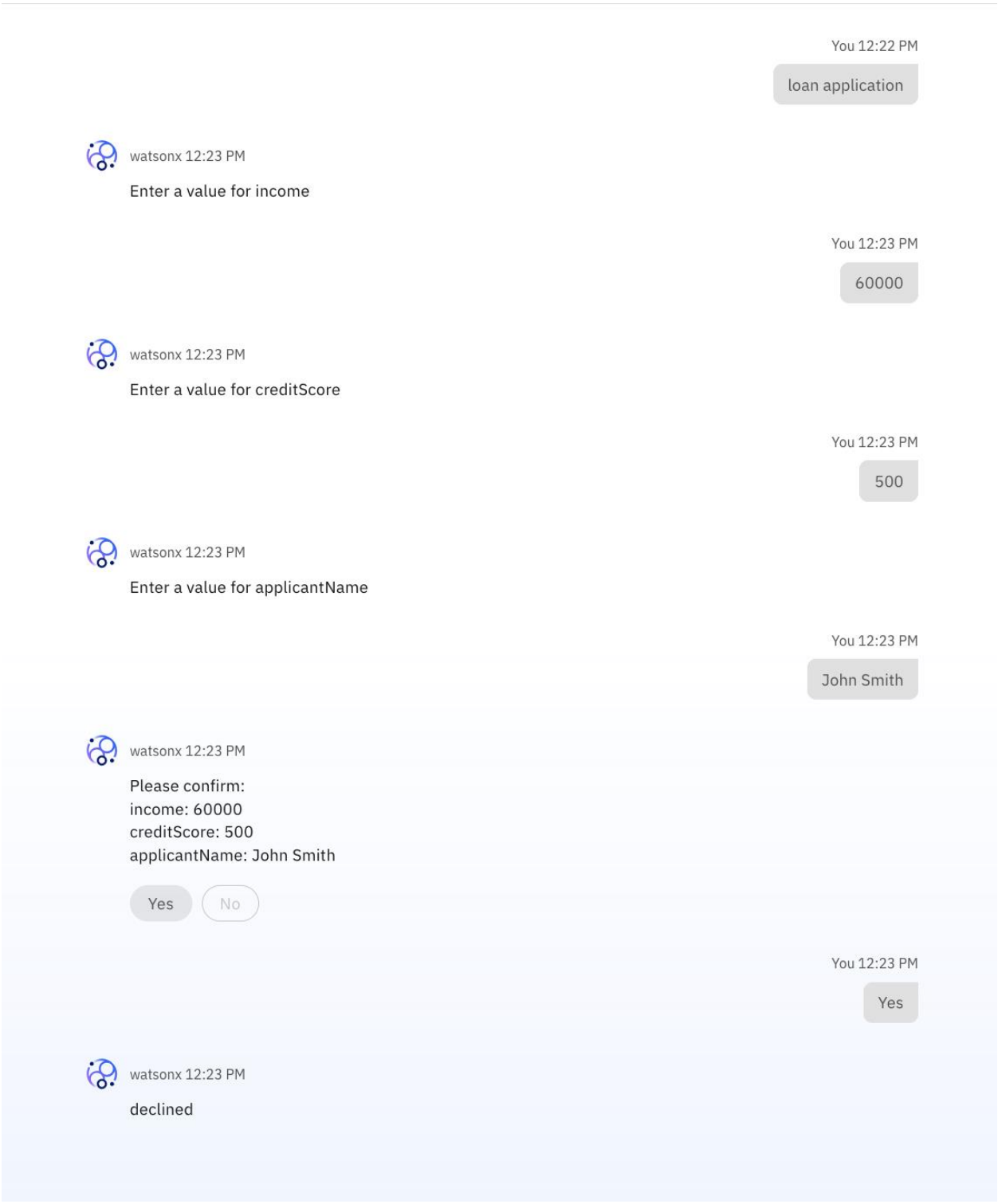
5. Navigate to **Chat** from the menu:



6. Type your query, e.g. ***loan application*** in the chat window:



7. The agent routes your query to the decision automation skill as expected:

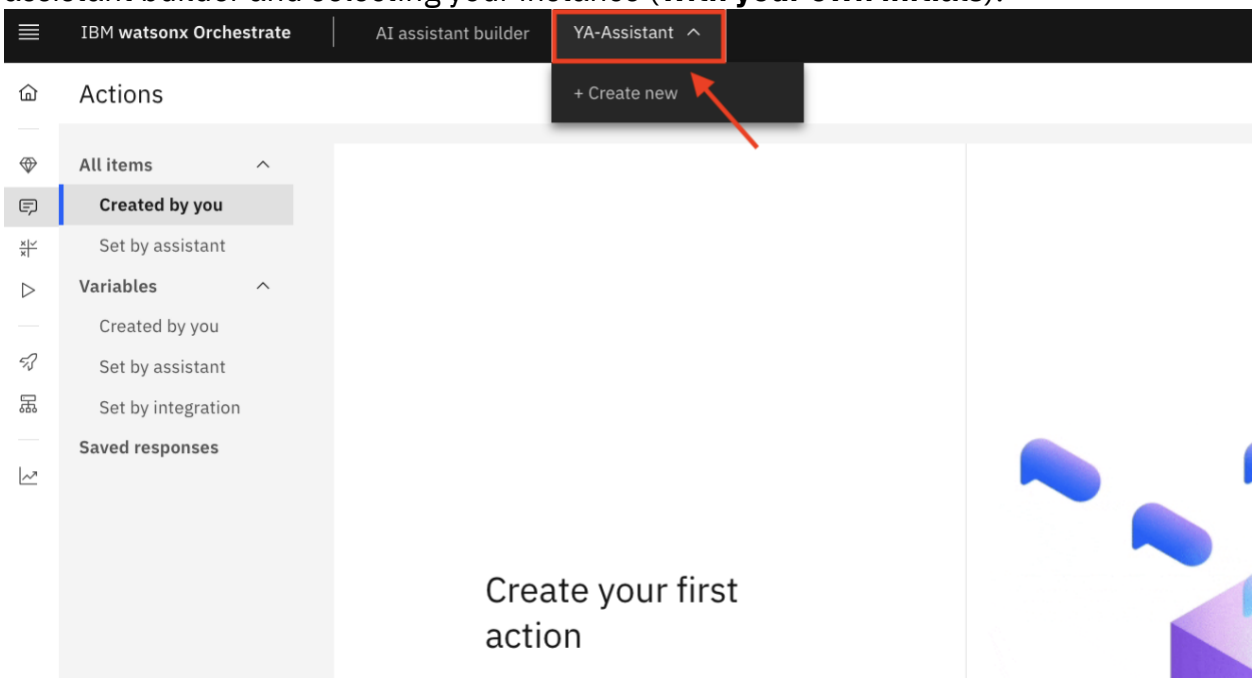


The application is declined, as expected, since the credit score is < 600

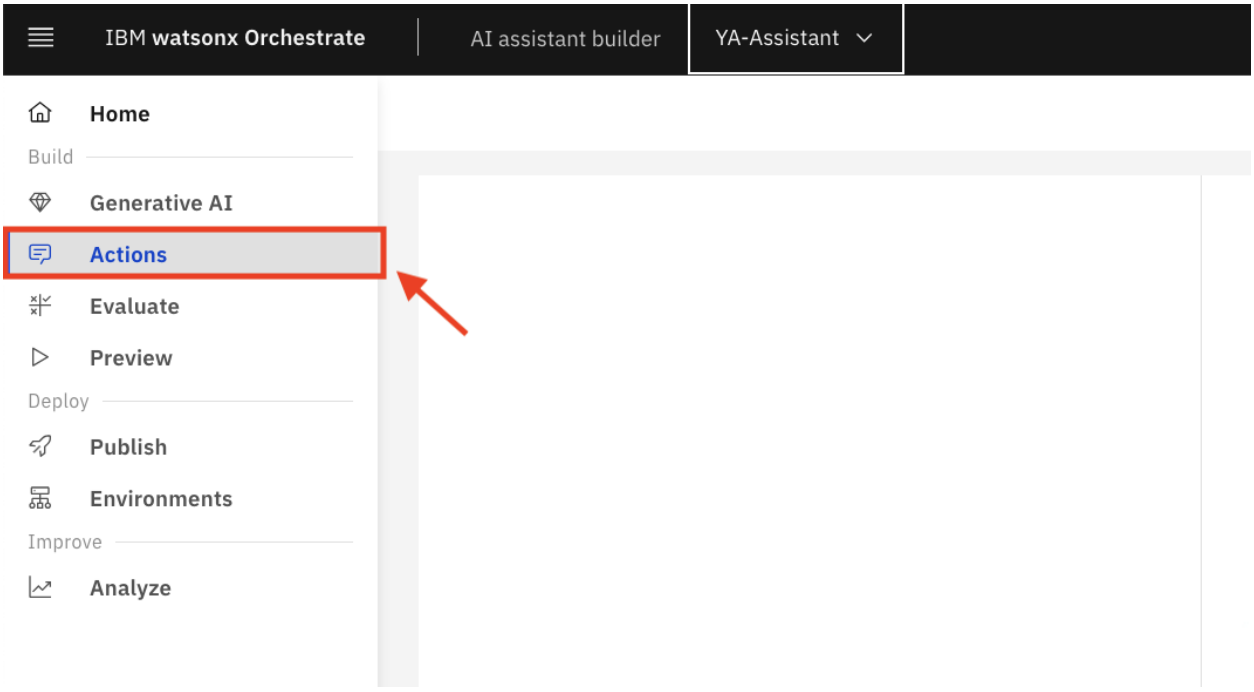
Step 1.6: Import the decision skill as an action in the assistant

As you saw in the previous step, you can bring an automation-based skill directly into the AI chat. Another option, if you want to make this skill a part of a conversation flow, is to make it available in an assistant. For example, we can use the assistant we created in lab 2 and create a skill-based action using the decision automation we just built. Once our assistant is implemented and published, we can add it to the AI chat instead of adding an individual skill (see Lab 4).

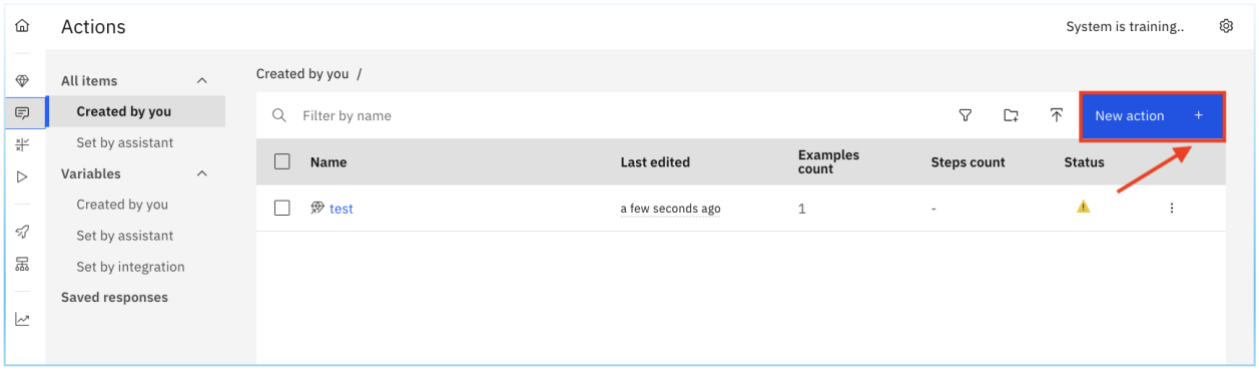
1. Open the assistant builder instance you created earlier by going into the AI assistant builder and selecting your instance (**with your own initials**):



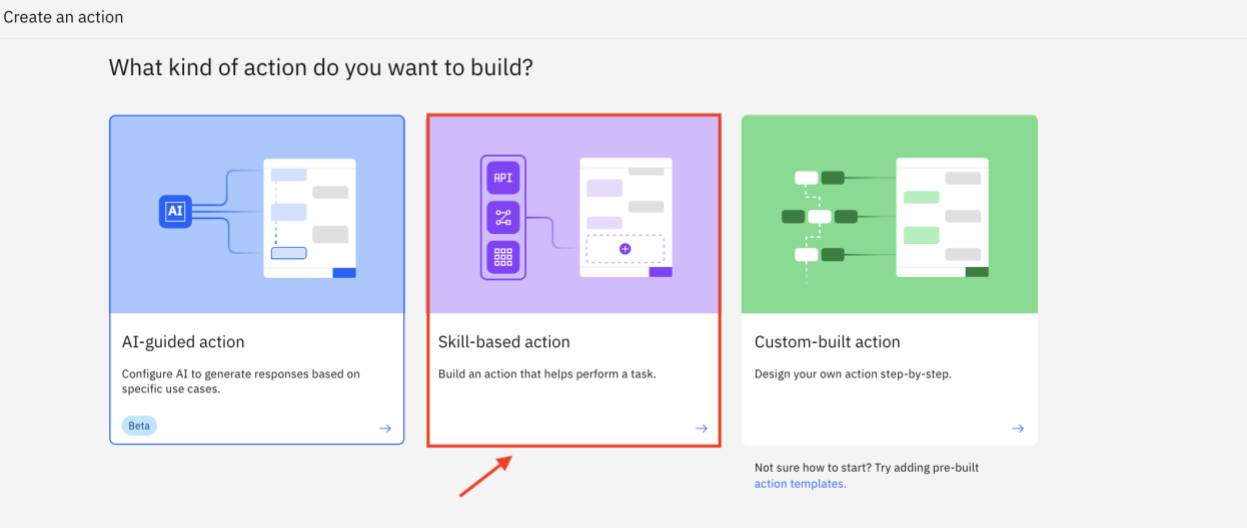
2. Open the actions tab:



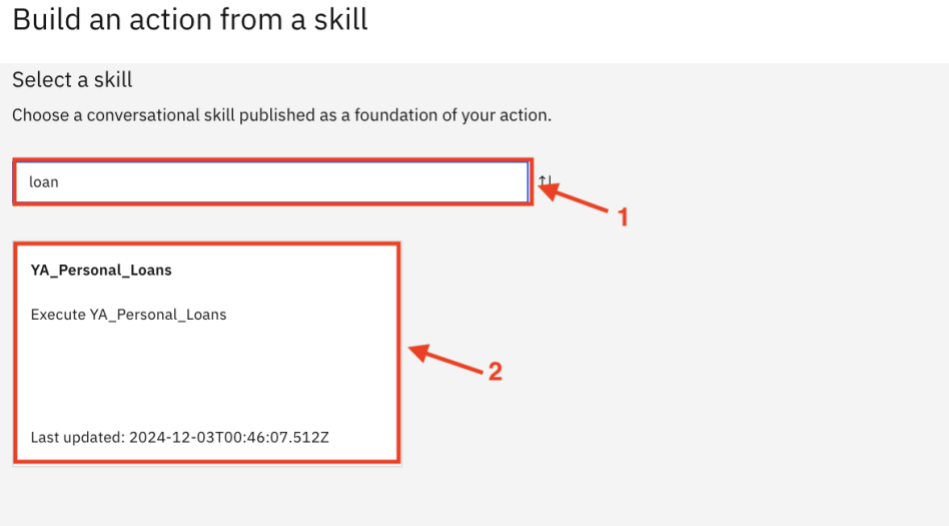
3. Create a new action:



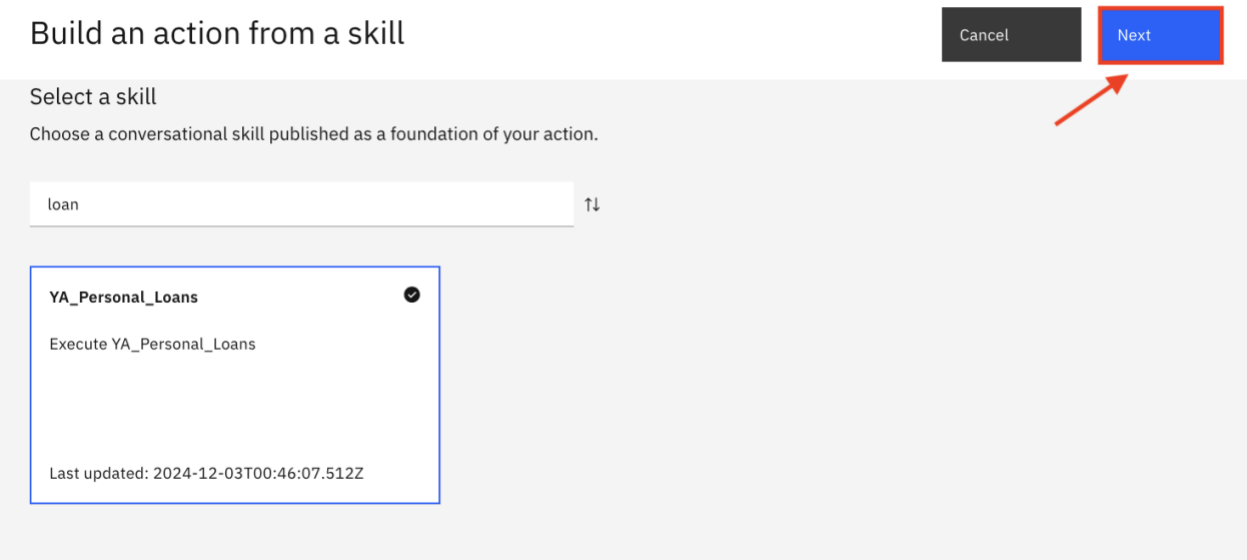
4. And make it a skill-based action:



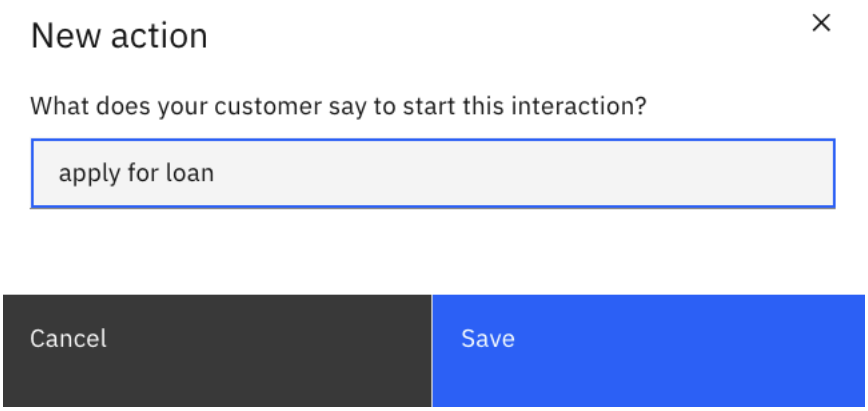
5. Search for your decision skill and select it:



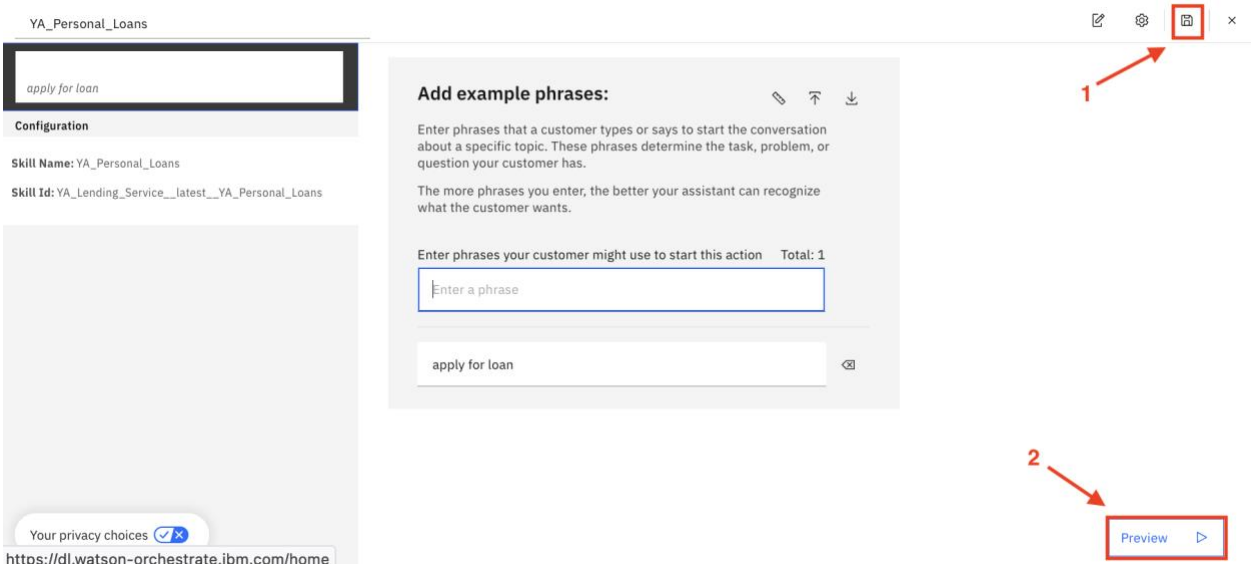
6. Click **Next:**



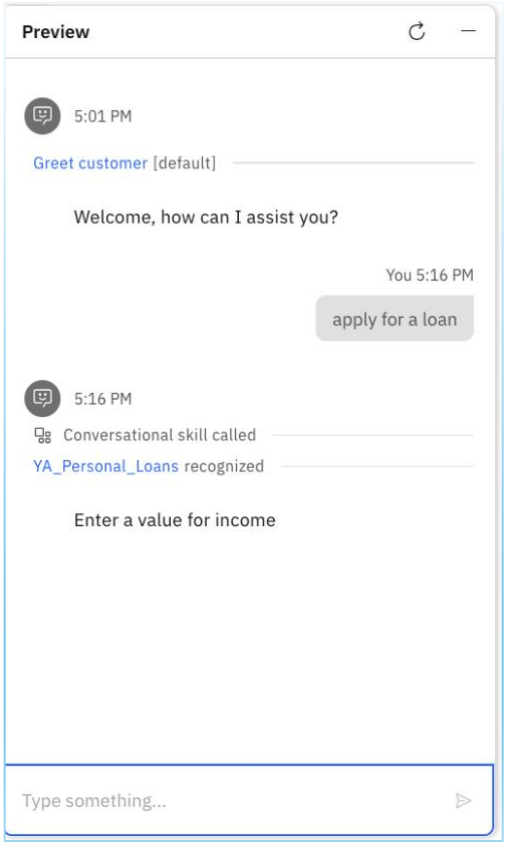
7. Provide an utterance (a phrase a user might type to invoke the action), e.g. **Apply for a loan** and click on **Save:**



8. Click on **Save** and then on **Preview:**

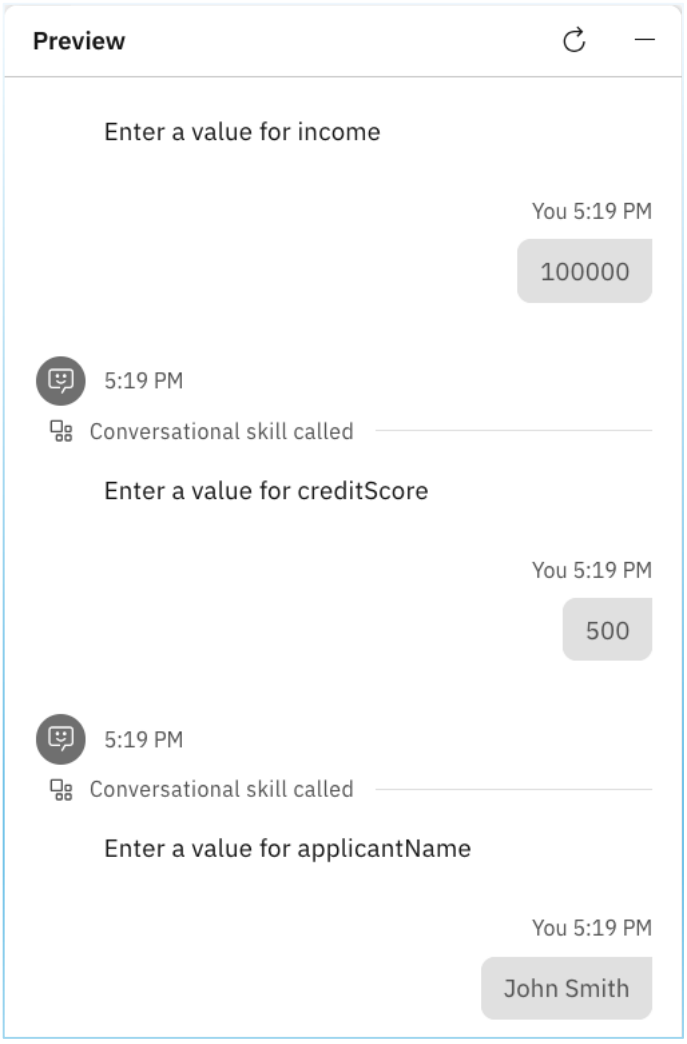


9. Once the changes have been added, type the suggested utterance in the assistant preview window and hit enter. You will now be asked to provide some of the information required as input for the decision automation:

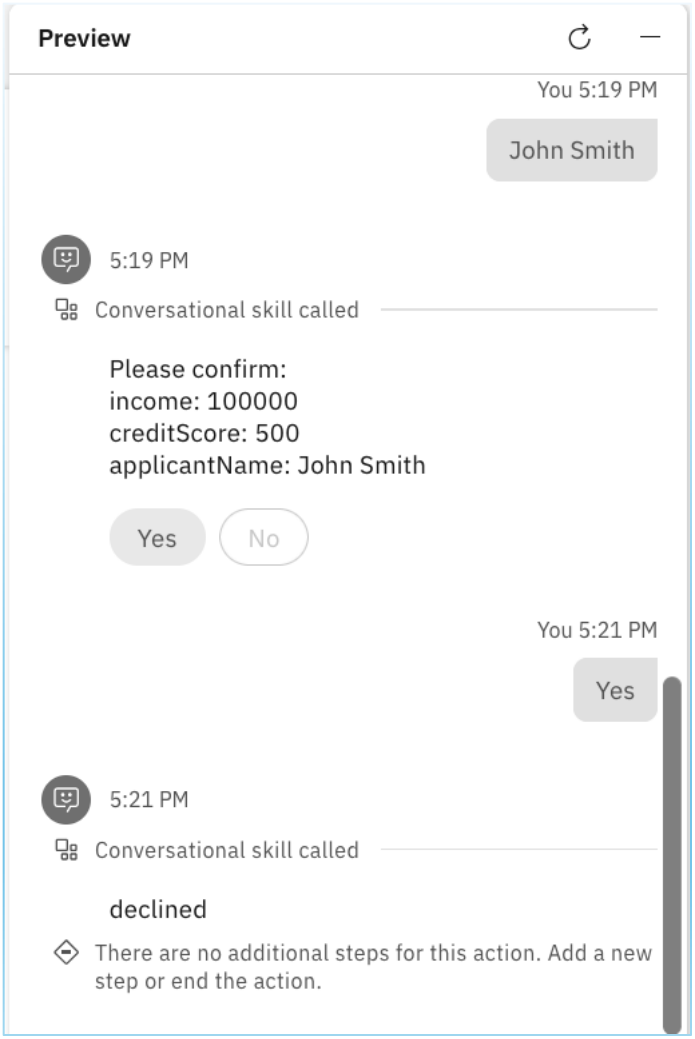


10. Provide answers as follows:
    Income = 100,000
    Credit score = 500
    Your name

    The conversation should look similar to:



11. At this point you will be asked to confirm the entered data, and you will be notified of the loan approval decision:



Now you have implemented a skill-based action in your assistant. It uses the decision automation skill you built earlier for the loan approval process. Of course, this is just a very simple version and much more can be done in watsonx Orchestrate! You can also bring other available skills into your assistant. Additionally, it is very easy to further configure how the questions and answers are displayed in the chat.

This concludes the lab