

- 29 -

Use REST API actions  
in Copilot Studio



MONTHLY MASTERY

# FEATURE-A-DAY

with Copilot Studio



Sprinkled with holiday magic by  
Katerina Chernevskaya



# REST API actions

## Why it matters?

Using REST API actions in your app enables:

- **Dynamic functionality:** Interact with real-time data from external services.
- **Automation:** Trigger workflows or processes in third-party tools.
- **Enhanced user experience:** Provide up-to-date information directly in your app.
- **Integration across systems:** Connect siloed systems to create a unified experience.

REST API actions make it simple to **enrich** your app's capabilities without needing to develop everything in-house.



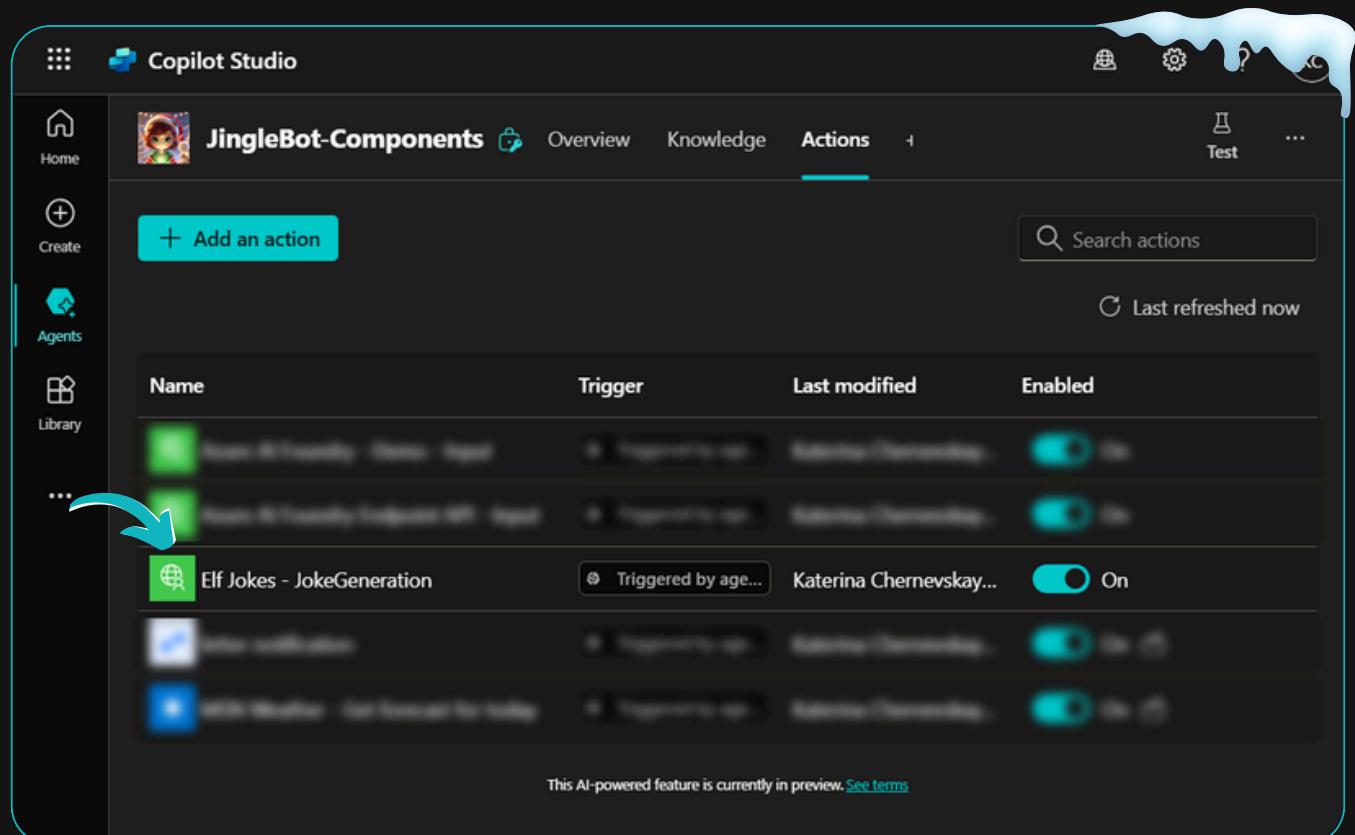
Sprinkled with holiday magic by  
Katerina Chernevskaya



# Use REST API actions

## Prepare the action

For this demo, we have prepared a simple REST API action that connects to the [Azure AI Foundry Prompt Flow Endpoint](#).



The screenshot shows the Copilot Studio interface with a dark theme. On the left, there's a sidebar with icons for Home, Create, Agents, Library, and an ellipsis. The main area is titled "JingleBot-Components" and shows a list of actions. A blue arrow points from the ellipsis icon in the sidebar to the "Add an action" button at the top of the list. The list has columns: Name, Trigger, Last modified, and Enabled. One action is highlighted: "Elf Jokes - JokeGeneration", which is triggered by age and was last modified by Katerina Chernevskaya. It is currently enabled. At the bottom of the list, it says "This AI-powered feature is currently in preview. [See terms](#)".



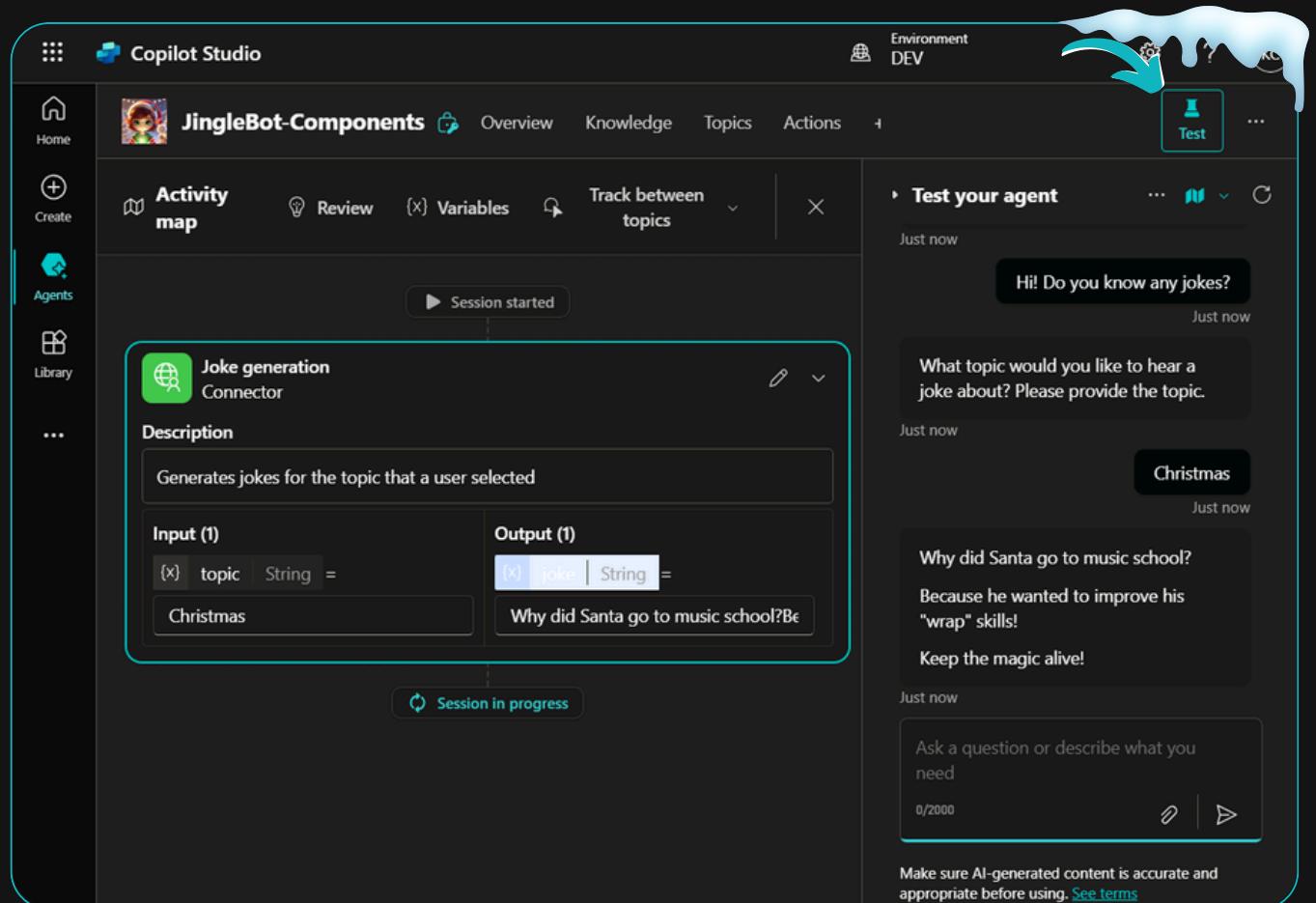
Sprinkled with holiday magic by  
Katerina Chernevskaya



# Use REST API actions

## Test the action

Open the **Test tool** and input a question or test that activates the action. The trigger phrase will vary based on the **description** you specified for the action.



Sprinkled with holiday magic by  
Katerina Chernevskaya



# Use REST API actions

## Update input description

By providing a clear **description for the action input**, you can effectively guide the action to extract the required information from the user's question.

The screenshot shows the JingleBot Components interface with the 'Elf Jokes - JokeGeneration' action selected. The 'Inputs' tab is active. The 'Topic' input is highlighted with a teal arrow pointing to its 'Description' field, which contains the text: 'The topic that a user want a joke about'. Other fields shown include 'How will the agent fill this input?' set to 'Dynamically fill with best option (default)', 'Display name' set to 'topic', and 'Identify as' set to 'User's entire response'.



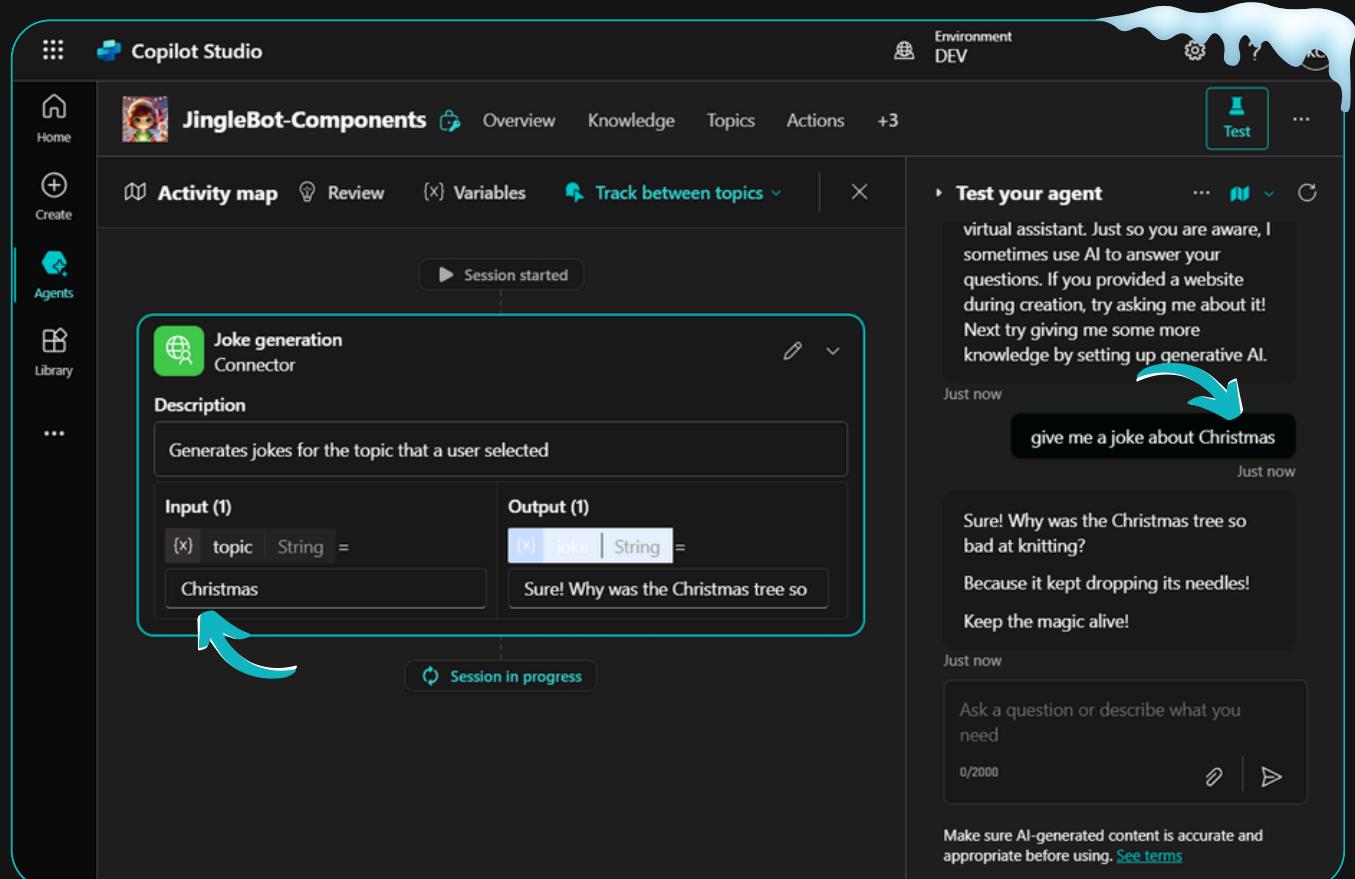
Sprinkled with holiday magic by  
Katerina Chernevskaya



# Use REST API actions

## Test again

A clear description of the action input helps the AI understand the user's intent and **avoid asking unnecessary questions**.



Sprinkled with holiday magic by  
Katerina Chernevskaya



# Use REST API actions

## Benefits

### Ease of integration

Leverage AI to guide you through connecting your app with external services effortlessly.

### Dynamic updates

Fetch or send live data automatically, powered by AI-driven insights.

### Smarter workflows

Trigger automated processes or make data-driven decisions directly in your app.



Sprinkled with holiday magic by  
Katerina Chernevskaya



# Today's Task: Use your REST API action



## 1. Pick a REST API action

Choose an action you've already created.

## 2. Check the action description

Ensure the description is clear. Test in the Test tool by asking something that can trigger your action.

## 3. Update action input description

Open the action, navigate to the Input tab and update the description for the Input. Test again to check that action can fetch required information from the user's question avoiding unnecessary questions



Sprinkled with holiday magic by  
Katerina Chernevskaya





MONTHLY MASTERY

# FEATURE-A-DAY

with Copilot Studio



Sprinkled with holiday magic by  
Katerina Chernevskaya

*Follow for more!*