**watson-assistant-watsonx.ai-integration**

Table of Contents

[# Integrating Watson Assistant with watsonx.ai foundation models 1](#_Toc170137183)

[## Prerequisites 1](#_Toc170137184)

[## Getting Started 1](#_Toc170137185)

[### Obtain the OpenAPI definition of the watsonx.ai generation endpoint 1](#_Toc170137186)

[### Add an empty virtual assistant to the Watson Assistant instance 1](#_Toc170137187)

[### Creating a watsonx.ai API key 2](#_Toc170137188)

[### Getting your watsonx.ai project id 2](#_Toc170137189)

[### Adding the custom extension for watsonx.ai to the Integrations catalog 3](#_Toc170137190)

[### Create your action 5](#_Toc170137191)

[### Testing the action 13](#_Toc170137192)

[## Next Steps 14](#_Toc170137193)

## # Integrating Watson Assistant with watsonx.ai foundation models

The extensibility of the new Watson Assistant provides a simple way to perform an integration between a dialog flow and generative AI inferencing services in watsonx.ai. This integration brings generative AI features directly into Watson Assistant. This hands-on exercise will show how to create the integration and use it in a simple use case. It is outside the scope of this guide to provide an introduction to all Watson Assistant features. For that, check out the [Documentation](https://cloud.ibm.com/docs/watson-assistant?topic=watson-assistant-welcome-new-assistant)

## ## Prerequisites

- Watson Assistant instance on IBM Cloud - this can be any service plan, including lite

- Access to a watsonx.ai project

## ## Getting Started

### ### Obtain the OpenAPI definition of the watsonx.ai generation endpoint

Before you can add the integration for watsonx.ai to a Watson Assistant virtual assistant, you need to download the OpenAPI definition for the watsonx.ai foundation model inferencing service.

### ### Add an empty virtual assistant to the Watson Assistant instance

For initial experimentation with the integration, use a new virtual assistant. If you have a newly created Watson Assistant service instance created, it will not have any assistants defined. Create one using the Create Assistant wizard using a name of Watsonx integration and then selecting web as the deployment location. The values you provide in the other parts of the wizard will not matter very much.

If you have other assistants defined already in your instance, add one following the [documentation](https://cloud.ibm.com/docs/watson-assistant?topic=watson-assistant-assistant-add)

### ### Creating a watsonx.ai API key

If you have previously created an API key for calling watsonx.ai from a notebook or just the REST endpoint with curl you may skip these steps and reuse that API key. Otherwise, continue on.

* Log in to [watsonx.ai](https://dataplatform.cloud.ibm.com/wx/home?context=wx) and make sure that the same account as the watson machine learning instance for your watsonx.ai project is selected.
* To generate an API key from your IBM Cloud user account, go to [Manage access and users - API Keys](https://cloud.ibm.com/iam/apikeys) and click on Create. In the pop-up, provide a name like watsonx-apikey and a short description, then click Create

A screenshot of a computer

Description automatically generated

* Immediately make a copy of the API key and store it in a secure vault (like 1Password, etc.) as there will be no way to retrieve it again.

### ### Getting your watsonx.ai project id

* In the watsonx.ai page, select the left 4-bar pull down menu, expand Projects and then click on View all projects.
* From the list, select the project that your are using for the integration.
* Click on the Manage tab and check that General is selected. Copy the Project ID to a location where you can access it later. This is not sensitive like the API key so it can be in notepad or something equivalent.

A screenshot of a computer

Description automatically generated

### ### Adding the custom extension for watsonx.ai to the Integrations catalog

* Return to your browser tab with the assistant you created. In the assistant left navigation panel, go down to the lower section and click on Integrations
* Scroll down to the Extensions section and click on the Build custom extension button:

A screenshot of a computer

Description automatically generated

Review the getting started guide and click on Next

* On the basic information tab, name the extension watsonx and provide an optional description. Click Next
* Either drag and drop or use the file browser to select the watsonx-openapi.json file and then click Next

A screenshot of a computer

Description automatically generated

* Review the information displayed and click on Finish, now the integration catalogue shows the integration

A screenshot of a computer

Description automatically generated

* Click on Add in the lower right corner of the new integration tile, confirm Add when prompted
* Review the Get started information and click on Next
* In the Authentication panel, select Oauth 2.0, enter in the API key that you created earlier, keep the remaining default values and then click Next

A screenshot of a computer

Description automatically generated

* On the last page review the POST request parameters and response values. The assistant can set parameters in the request before calling the extension and will parse the response to show the user the output. Click on Finish, then click on Close if you are not returned to the Extensions section of the Integrations panel.

Congratulations, you now have a way to make a functional integration from an Action in assistant with watsonx generative AI endpoint.

### ### Create your action

Now it is time to create your action and have it use the integration to watsonx. To get started, you will create an action that uses generative ai to draft a marketing message from your prompt.

1. In the browser tab with Watson Assistant and click on Actions in the upper left.
2. In the main panel, click on Create action
3. Select Start from Scratch
4. For what the customer says enter Marketing Message and click on Save
5. Click on the pencil icon to set a step title. Use Prompt for a prompt as the title.
6. For what the assistant says, you can either make it a short message or be more prescriptive. Note - humans, like generative AI models can often do better when requests are specific.

For example:

**Please provide a prompt for the model to use to create the marketing message. Type as much as you like and include lists of details for me to use with the model.**

A screenshot of a chat

Description automatically generated

Expand Define customer response and specify that the assistant will expect a response from the user as Free Text input. When you click on Free Text, the UI will show User enters free text below the Assistant says panel. When this is shown, click on New Step to configure the next part of the Action.

* Name the next step Call watsonx.ai endpoint by clicking on the pencil (edit) icon. In the And then section, select Use an extension

A screenshot of a computer

Description automatically generated

* For the extension to use, select watsonx. Then select the Generation operation to show the parameters that can be sent to watsonx.ai on every invocation. For version, choose Enter text and then provide 2023-05-29, For input, choose Action Step Variables and then choose 1. Prompt for a prompt, continuing in the same way, set the model\_id to google/flan-t5-xxl and the project\_id to your watsonx.ai project id saved before. Don't apply yet

A screenshot of a computer

Description automatically generated

* Expand the optional parameters.
* For this text generation scenario, use temperature 0.8, set max\_new\_tokens to 200, set min\_new\_tokens to 50 and set repetition penalty to 2, set decoding\_method to sample.
* Then click on Apply

A screenshot of a computer

Description automatically generated

* Click on New Step
* In this step, select with conditions . In the condition, click on the first item and select watsonx (step 2)

A screenshot of a computer

Description automatically generated

* Select Ran successfully
* Click on Set variable values , then click on Set new value, then click on New session variable

A screenshot of a computer

Description automatically generated

* Call the variable result and select free text and then click on Apply

A screenshot of a computer

Description automatically generated

* For the variable assignment To, select Expression and type $ after this is pasted a pulldown will appear. Select watsonx (step 2)

A screenshot of a computer

Description automatically generated

* Select body.results, the box will fill in with the assigment for the result variable. Now add the text [0].generated\_text to the end of the expression:

A screenshot of a computer

Description automatically generated

* click Apply
* The full expression for the variable should be shown as:

A screenshot of a chat

Description automatically generated

* In the Assistant says panel, select function and then type in ${result}, or you can select Session variables and then result

A screenshot of a computer

Description automatically generated

The assistant should be ready, time to test it out.

### ### Testing the action

1. Click on the Preview button.
2. After the greeting, type I need to create a marketing message, and send to the chatbot.
3. The chatbot will reply with the prompt for the prompt.
4. Enter the following text into the chatbot

**Generate a 5 sentence marketing message for a company with the given characteristics.**

**Details**

**Characteristics:**

**Company - Golden Bank**

**Offer includes - no fees, 2% interest rate, no minimum balance**

**Tone - informative**

**Response requested - click the link**

**End date - July 15**

**Email**

1. Send to the chatbot - if all worked, you will get a response back from the model with the message.

A screenshot of a chat

Description automatically generated

You will most likely see something different because we are using a sample (vs. greedy) decoding approach. Don't like the response, re-run the assistant to get another message. Note - with these parameters, the assistant may create information like years for the date (because only July 15th is specified) or phone numbers. In a real use case, a human would use this generated message as a starting point, making edits as necessary and then approving the final version.

## ## Next Steps

Experiment in prompt lab for other scenarios using different prompts and parameters - then create new actions and adjust parameters for the new use case. Since all of the prompt is required for input in the current design, zero or one-shot (like the one in this lab) will work the best. However, with a bit more effort, it should be possible to define an assistant variable with more examples and then use an expression to build the prompt for the integration. Try it out!