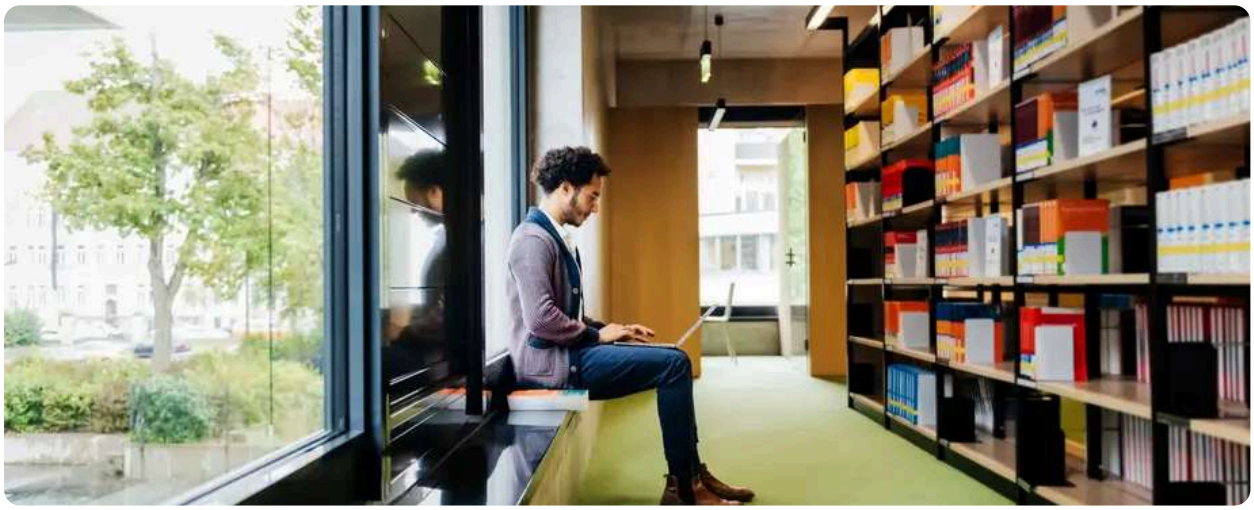
WORKBOOK

SAP Generative AI Hub with AI Launchpad, Orchestration and Document Grounding

7 Units 1 hr 45 mins Hands-on practice

Start Workbook



Overview

Get an introduction to the AI Launchpad in SAP Generative AI Hub. Learn how to create and manage LLM prompts, and experience a practical example of Retrieval Augmented Generative (RAG) search using document grounding.

SAP Generative AI Hub with AI Launchpad, Orchestration and Document Grounding

UNIT 1

Guided Tour

1 Lesson 5 mins

Content

Explore Demo Guided Tour

Go to learning

UNIT 2

Introduction to Generative AI at SAP

1 Lesson 5 mins

After completing this unit, you will be able to:
Get to grips with the AI Launchpad and its functionalities.

Content

Introduction to Gen AI at SAP

Go to learning

UNIT 3

SAP AI Core and SAP AI Launchpad setup

1 Lesson 10 mins

After completing this unit, you will be able to:
Setup and configure SAP Generative AI Hub using SAP AI Core and SAP AI Launchpad Boosters. SAP AI Core and SAP AI Launchpad are services which you can link to your BTP global account. SAP AI Core offers a powerful AI runtime which is natively integrated with SAP AI Launchpad. The launchpad offers an easy-to-use interface to manage AI workflow administration, processes, and tasks.

Content

- Quick start with SAP AI Core Setup

Go to learning

UNIT 4

Using SAP AI Launchpad

3 Lessons 35 mins

After completing this unit, you will be able to:
SAP AI Launchpad can be used by both AI scenario producers and AI scenario consumers. AI scenario producers, such as AI operations engineers or AI engineers, are responsible for developing and productizing AI scenarios. AI scenario consumers, such as business analysts, subscribe to a service that offers an AI scenario and consume it. The generative AI hub within SAP AI Launchpad can be used to interact with generative AI models.

Content

- Launchpad Introduction
- Chat
- Prompt Editor

Go to learning

UNIT 5

Prompt Management

1 Lesson 10 mins

After completing this unit, you will be able to:
Manage and maintain a collection of prompts

Content

Prompt Management

Go to learning

UNIT 6

Prompt Engineering

1 Lesson 20 mins

After completing this unit, you will be able to:
Design prompts in accordance with user requirements

Content

Prompt Engineering

Go to learning

UNIT 7

SAP Generative AI Hub Orchestration Service

2 Lessons 20 mins

After completing this unit, you will be able to:
Understand SAP Generative AI Hub Orchestration Service features and consumption methods.

Content

- Introduction to Orchestration
- Orchestration with Grounding

Go to learning

Orchestration with Grounding

Objectives

After completing this lesson, you will be able to:

See the effects of grounding your model

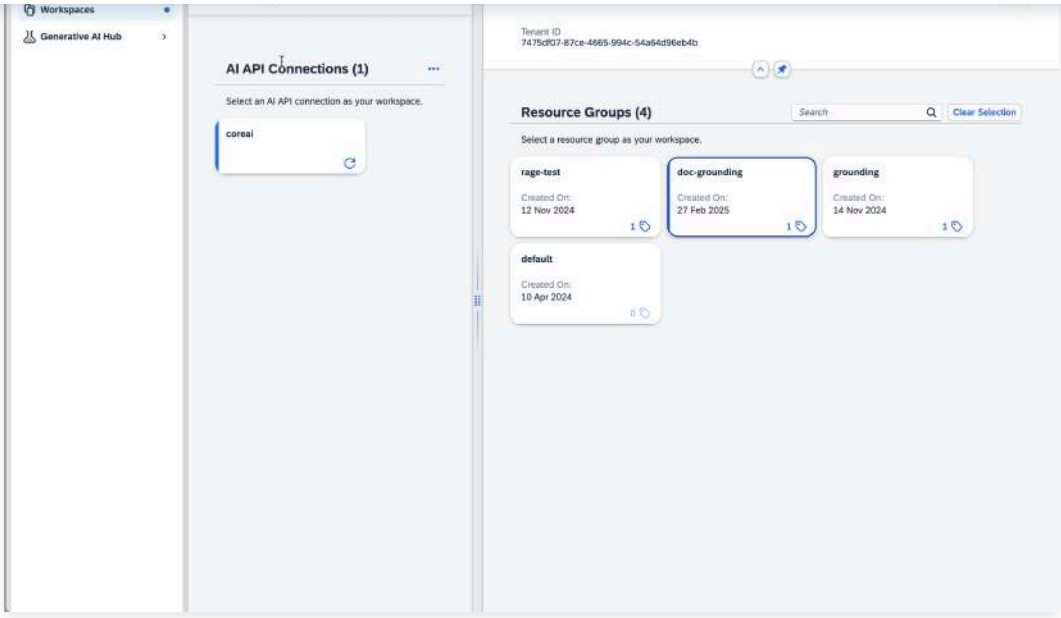
In this lesson, we are going to look at the concept of *Grounding* in terms of Generative AI. When a question is posed to a large language model, it will try to answer to the best of its knowledge, based on information it can find across the world wide web. However, it is usually much more beneficial to instruct the LLM to first consult a very specific domain before returning the results. This is known as Retrieval Augmented Generation , or RAG, whereby the results are 'augmented' by first grounding the LLM.

Grounding provides specialized data retrieval through vector databases, augmenting the retrieval process using specific external and context relevant data. Grounding combines generative AI capabilities with the capacity to use real-time, precise data to improve decision-making and business operations, for targeted business AI-driven solutions.

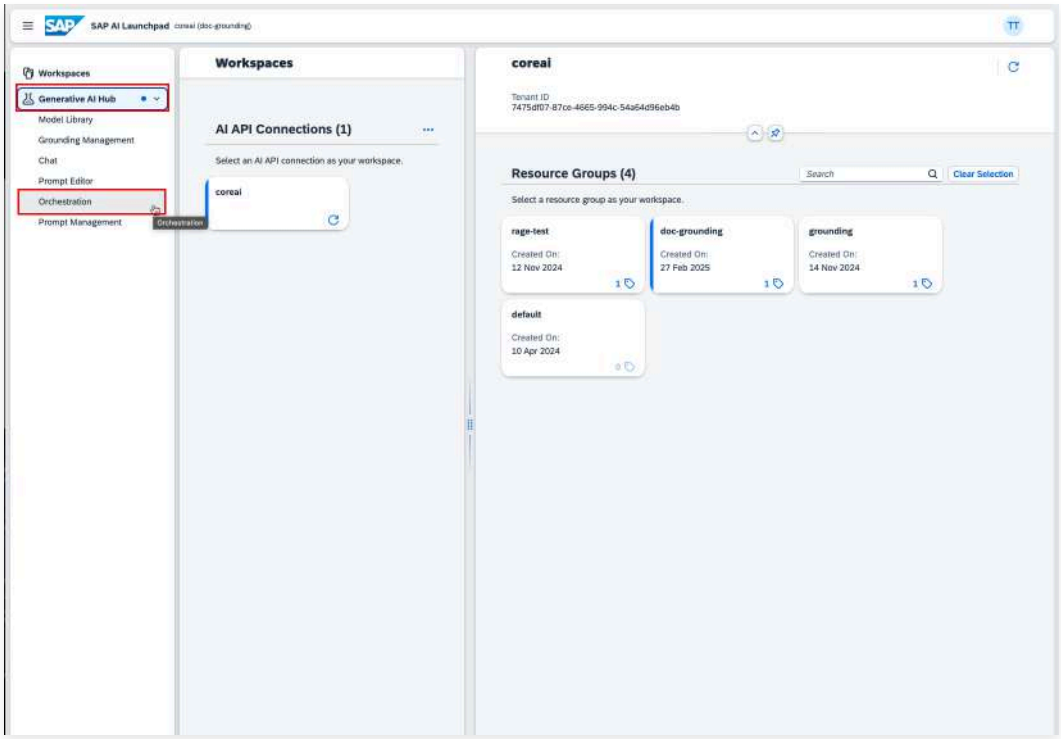
Try it out!

Note: Make sure to select the **doc-grounding** resource group as this is where the orchestration deployment with grounding is running.

Basic Trials



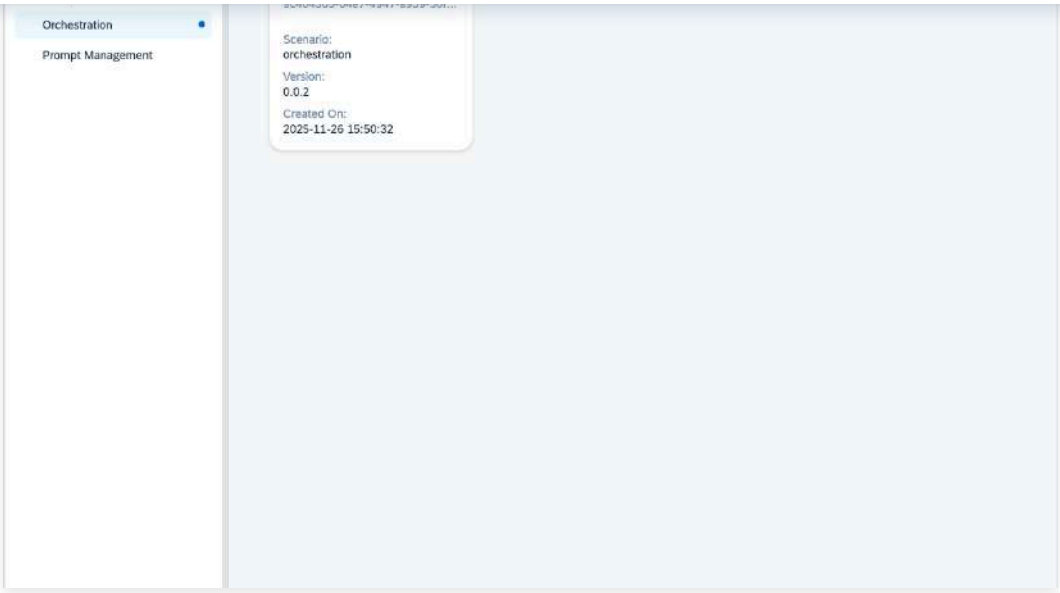
2. Expand **Generative AI Hub** and select **Orchestration** from the menu list to see the Orchestration Workflow.



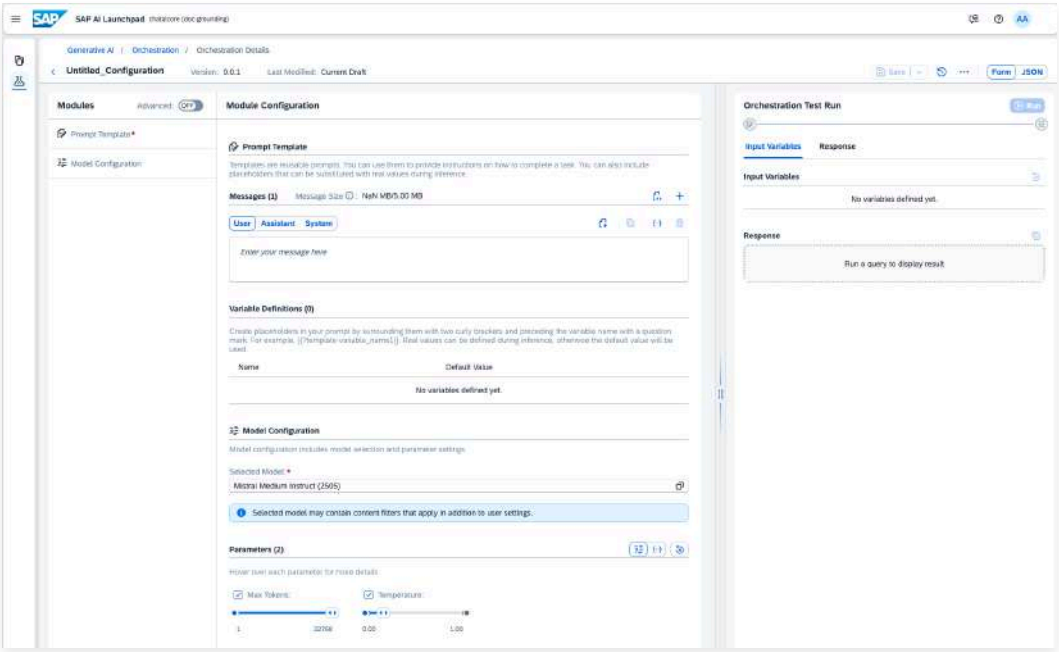
3. The *Orchestration Configurations* page will open, where any existing orchestration workflow configurations are displayed.
4. Select the **Create** button to create a new configuration.

Basic Trials

/ Workbooks / SAP Generative AI Hub with AI Launchpad, Orchestration and Document ...



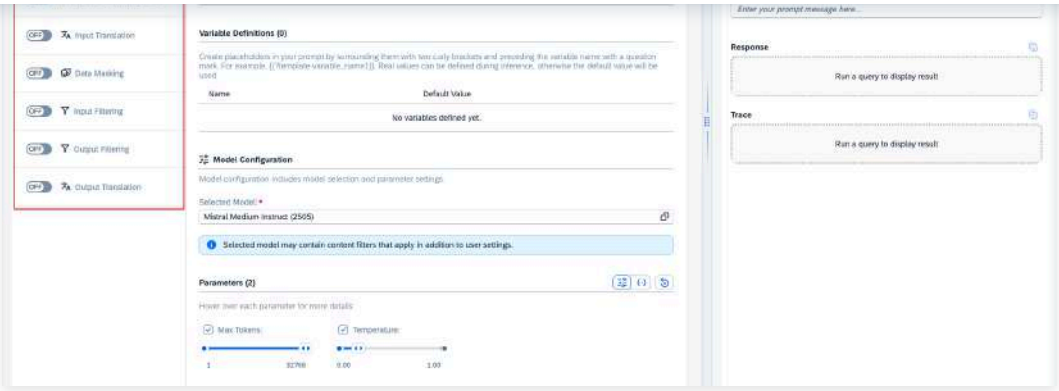
5. The *Orchestration Configuration* page will open where you can configure multiple settings for a range of different modules such as Grounding, Data Masking, Filtering and Tanslation.



6. Toggle the **Advanced** setting to **ON** to see the different modules available. You will notice that some of these are mandatory, while others are optional:

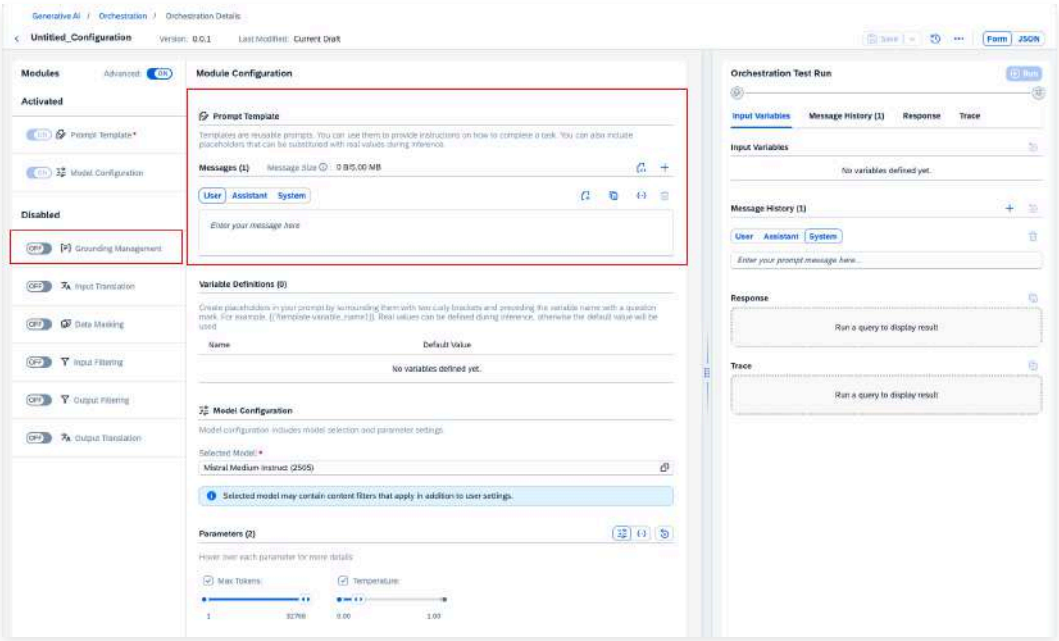
Basic Trials

/ Workbooks / SAP Generative AI Hub with AI Launchpad, Orchestration and Document ...



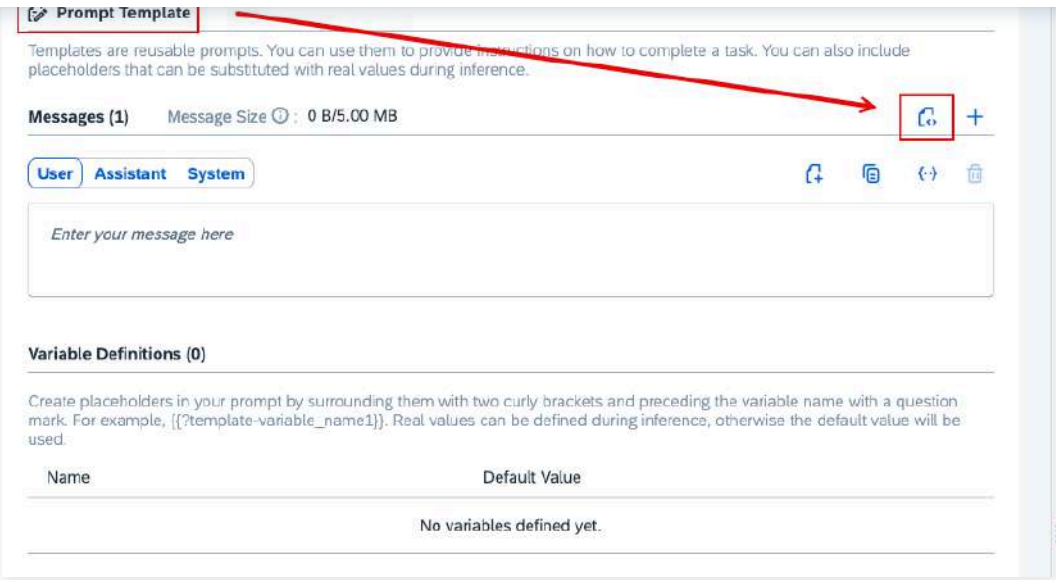
Let's first observe the effects of querying an LLM with no context and no frame of reference.

7. Ensure the *Grounding Management* feature is disabled (switch is set to **Off**) then go to **Prompt Template** section.



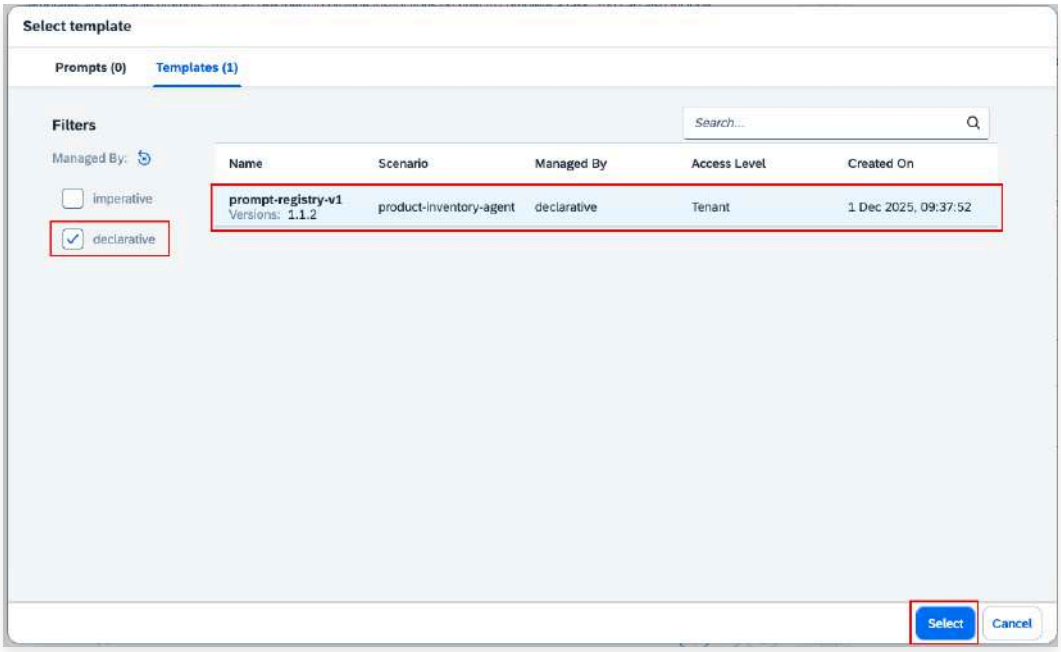
Using templates enables you to compose prompts, define placeholders, choose the LLM and configure it with required settings and parameters. It can also be enhanced with placeholders or variables to enable different prompts using the same basic configuration. Prompt Templates can be uploaded from a JSON file, the prompt library, or via the Prompt Registry using Github.

Let's upload a saved template which has been pre-configured and loaded into the prompt registry



The list of currently available templates will appear. It's possible to filter by the type of template (either imperative or declarative) or the search box can be used to find a specific template.

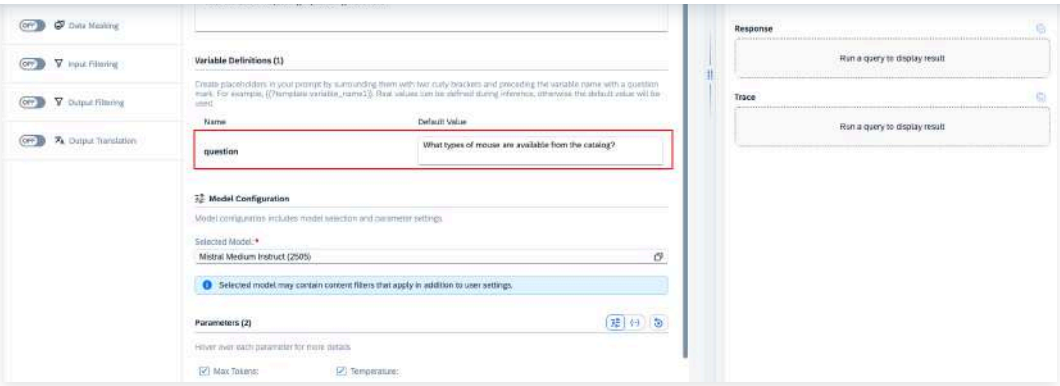
9. In the *Filter* section, select the checkbox next to **declarative** to reduce the list size. Then select the template named **prompt-registry-v1** and choose **Select** to add it to the templating module.



The template will now load up with pre-defined entries for both the **system** role and **user** role. You will also see a variable called **question** which has been defined and assigned a default value:

Basic Trials

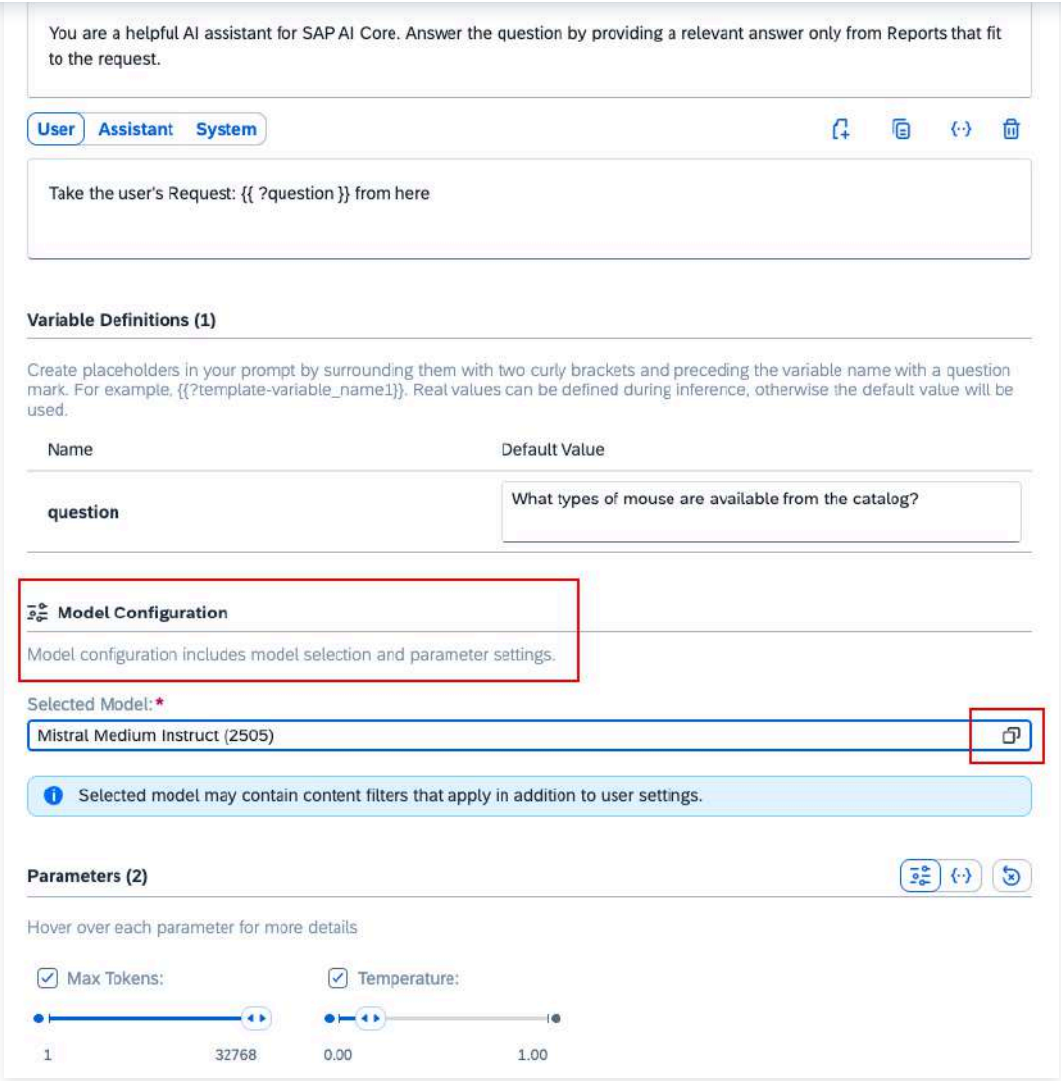
/ Workbooks / SAP Generative AI Hub with AI Launchpad, Orchestration and Document ...



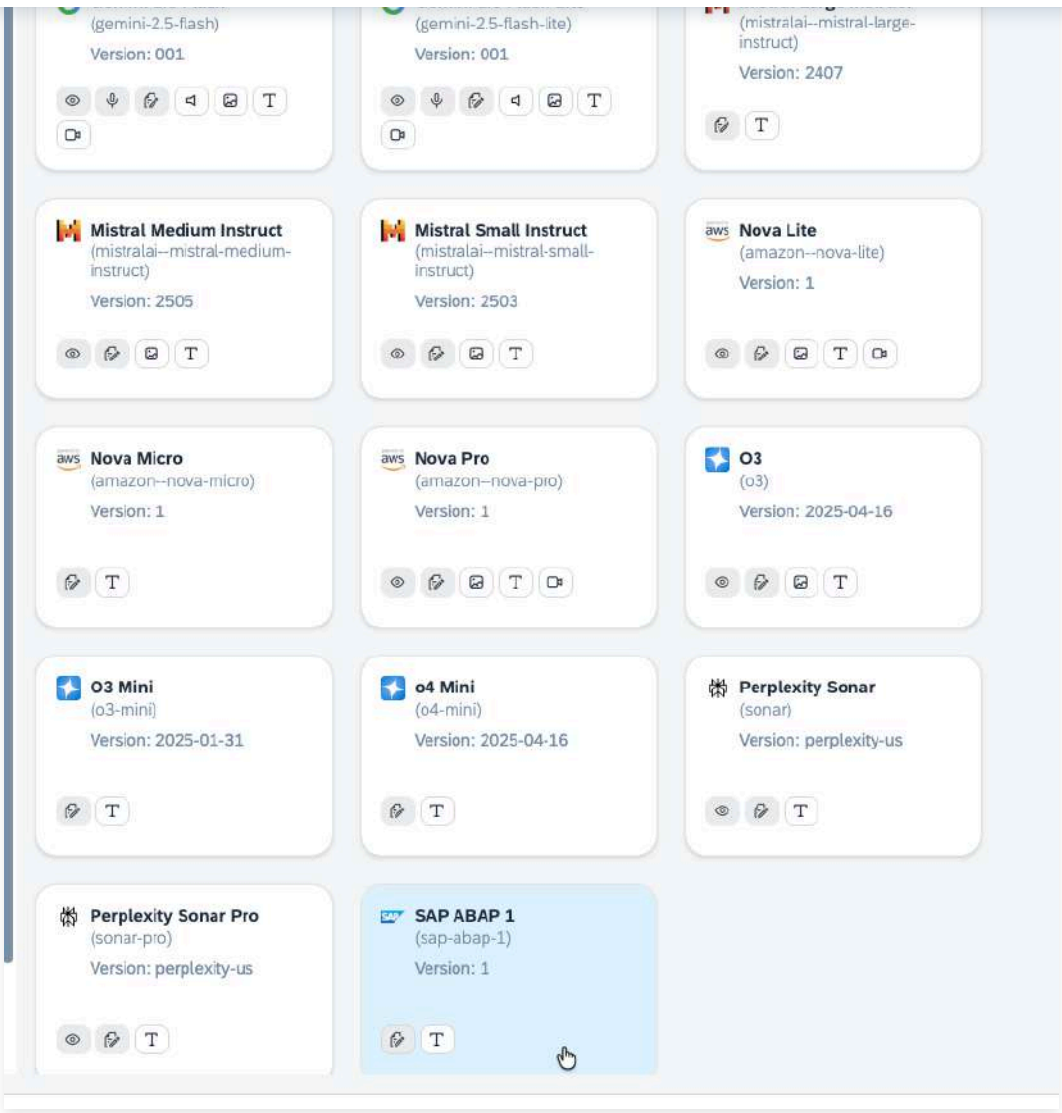
10. Scroll down to the **Model Configuration** section to choose a model from the available versions and define its parameters. A model has already been configured for this step, but you can choose from a large selection of other available models in here.

Basic Trials

/ Workbooks / SAP Generative AI Hub with AI Launchpad, Orchestration and Document ...



11. Open up the list of available models and choose **SAP ABAP 1** for this exercise. Selecting the model will automatically add it to the *Model Configuration* section where further parameters can be tweaked.

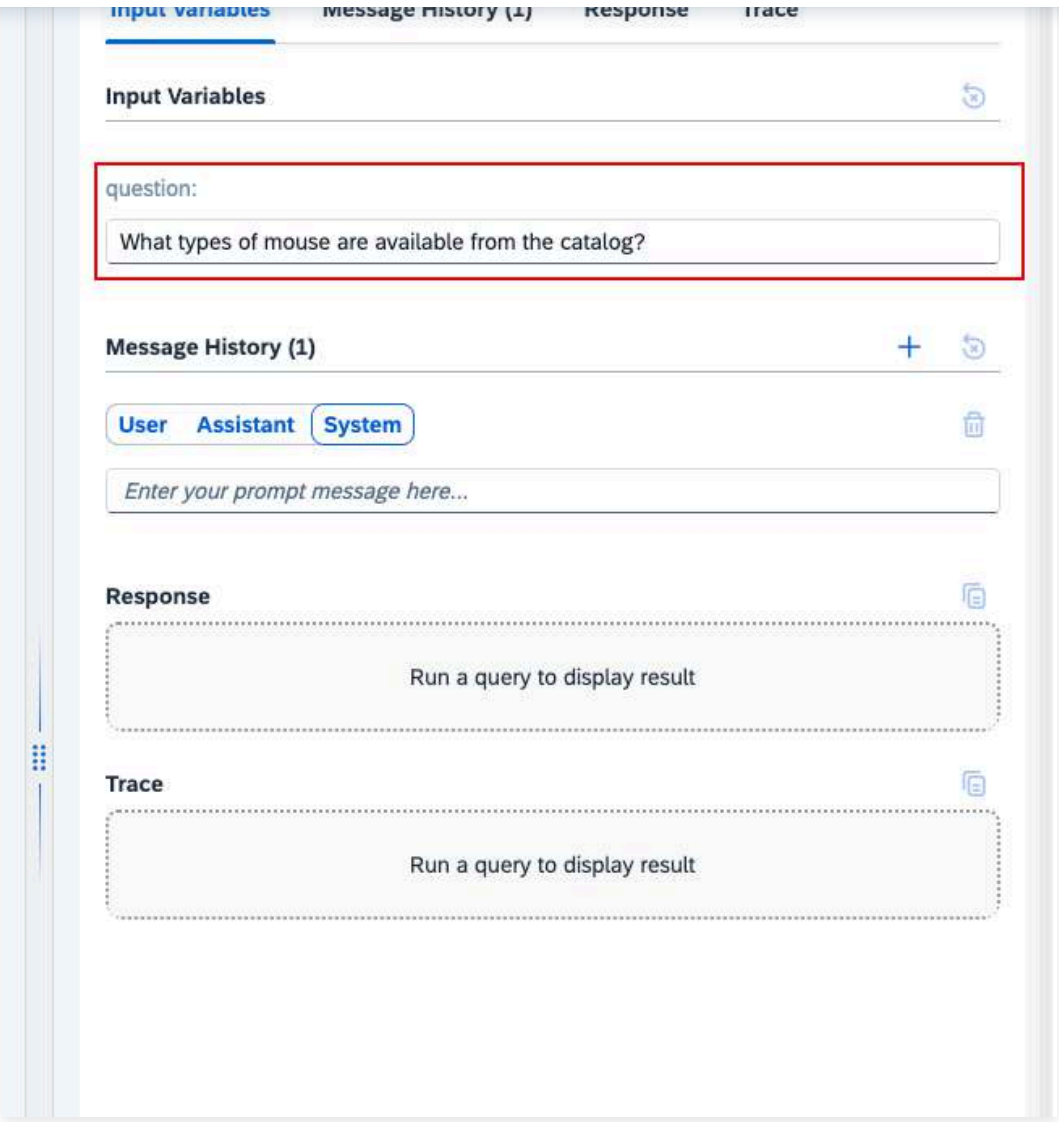


Once happy with the model selection and configuration, it is time to test out the prompt.

On the right hand side, you will see the **Orchestration Test Run** section, which has already been populated with the input variable from the template.

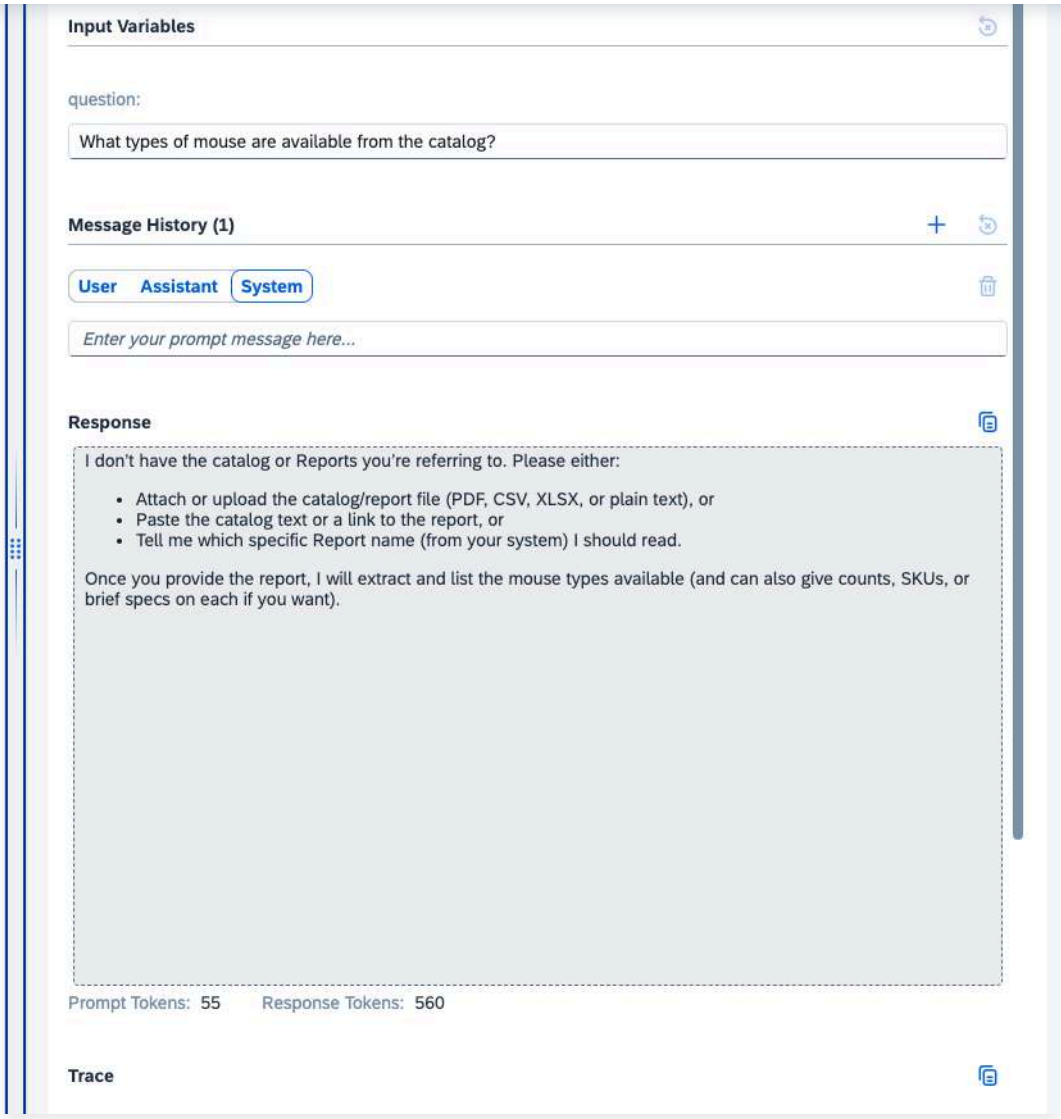
11. Select **Run** to send the query to your chosen LLM and observe the response.

Basic Trials

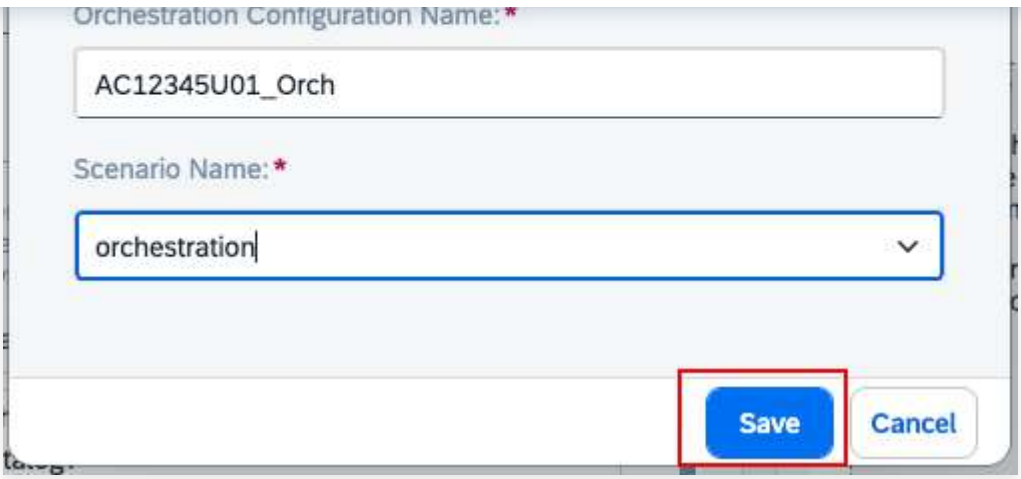


As this example uses a very targeted query with no context (looking for items in a specific catalog), the response from the model will either be an admission of failure to find the items, or it will make up an answer with generic options. This will depend on the model and version chosen, along with the parameters defined.

12. Save your current configuration by selecting the **Save** button at the top of the screen.



13. Give your configuration the name **GE275967_Orch**, select **orchestration** as the Scenario Name and then select **Save**.



In the next section, we can offer the model more context for the query by providing it with a database of information from which to search before providing an answer.

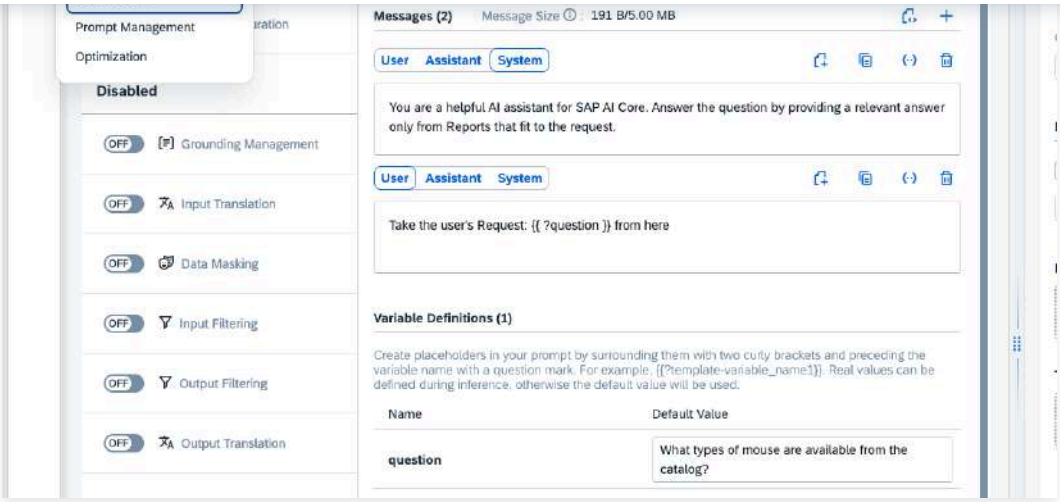
Grounding

Let's now 'ground' the LLM on a repository of data before asking the same question as before.

For this scenario, we have already created a set of embeddings from a product catalog which have been stored as vectors in a HANA Cloud Database.

1. Select the **Grounding Management** menu option within the *Generative AI Hub* section to view the available data repositories which can be used in this resource group (*doc-grounding*).

Basic Trials

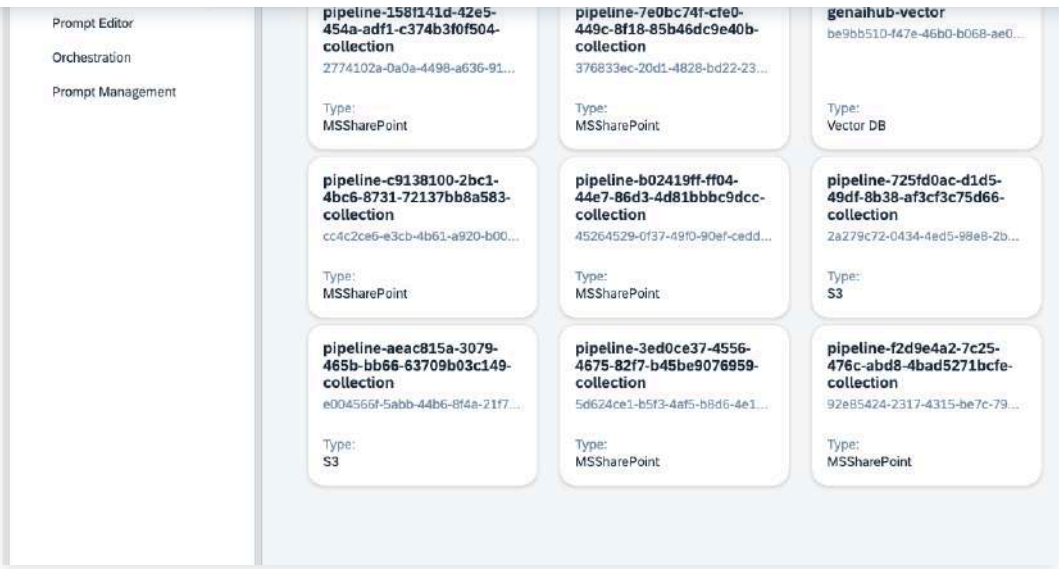


This is a collection of vectorized representations of various documents, csv files, images etc. which have been stored in MS Sharepoint, S3 buckets or directly in the vector store of SAP HANA Cloud itself.

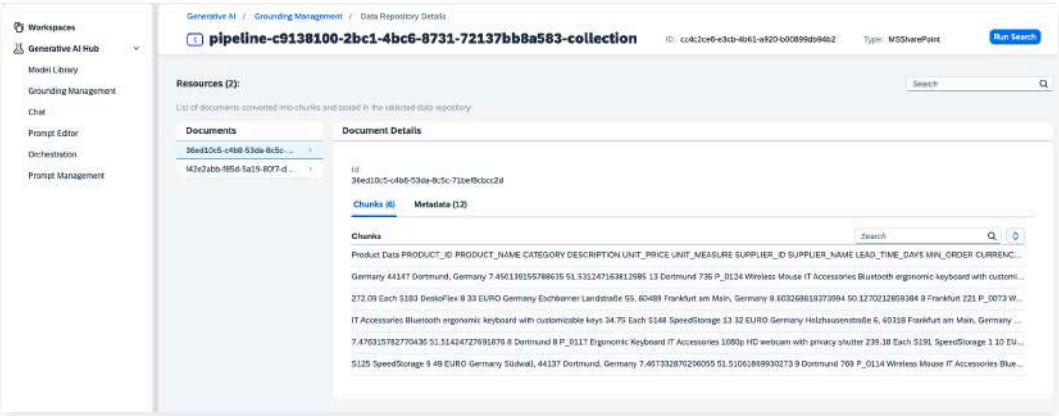
> **Note:** The repositories that you see may differ from those in the image below as this can depend on which tenant you are in and how many repositories have been pre-configured.

Basic Trials

/ Workbooks / SAP Generative AI Hub with AI Launchpad, Orchestration and Document ...

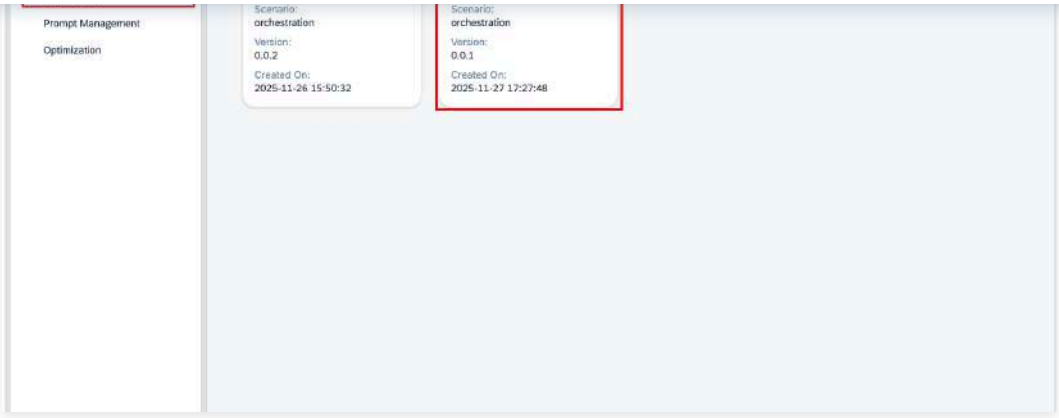


2. Select any of the repositories to view the data chunks. The repository we use for this workshop is called **pipeline-86a82ae8-e3a6-4f68-97b1-63568965f174-collection**. Select to open the data and observe the format.

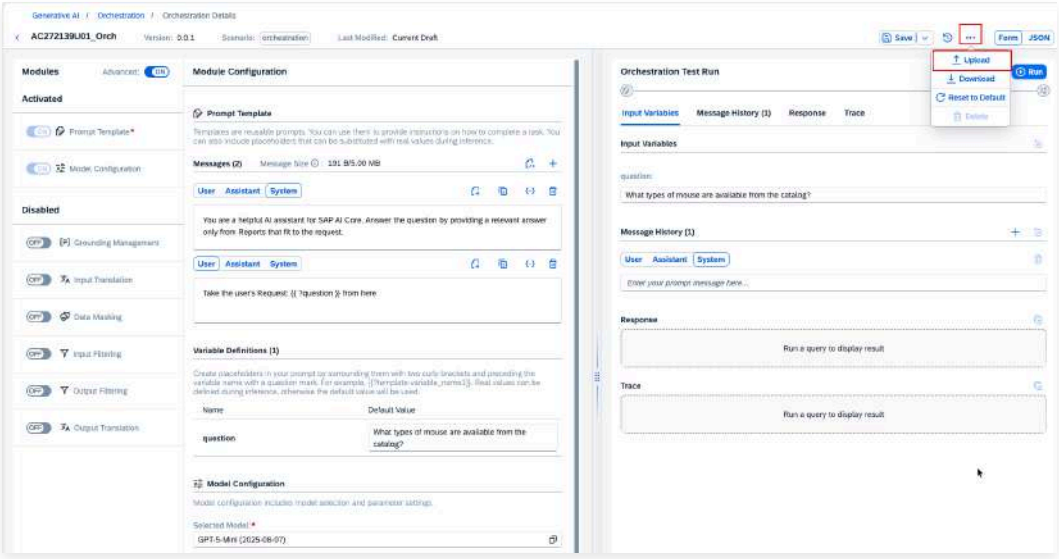


3. Download the following JSON template file , and save it locally on your device. Feel free to inspect the file and observe the format and layout.
4. Return to the orchestration workflow by selecting **Orchestration** from the menu and then choose your previously created configuration called **GE275967_Orch**.

Basic Trials



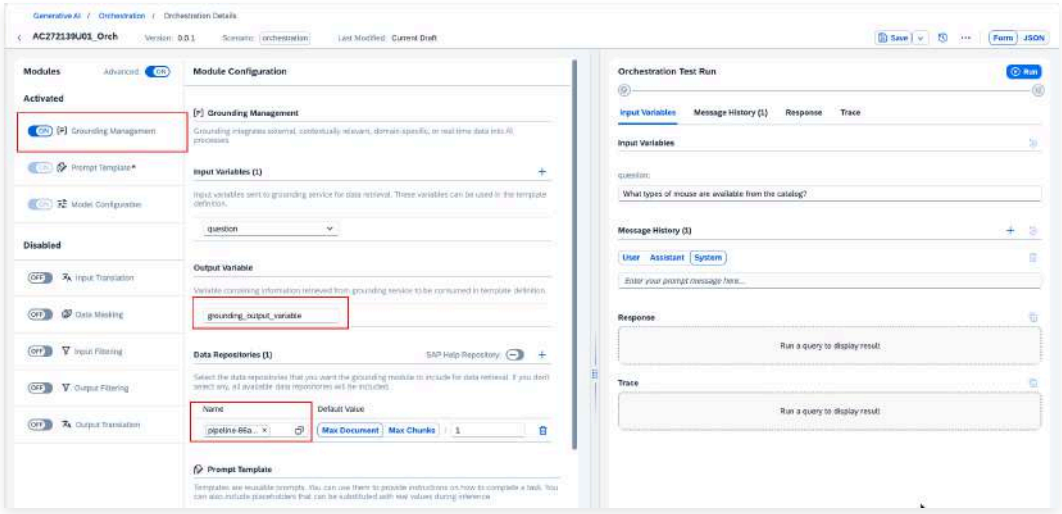
5. In the orchestration configuration screen, select the three dots at the top right and then choose **Upload** which will lead to a pop-up window asking for a file to be uploaded.



6. Choose the JSON file which you have saved in a previous step, and select **Open** to load it into the workflow.

Basic Trials

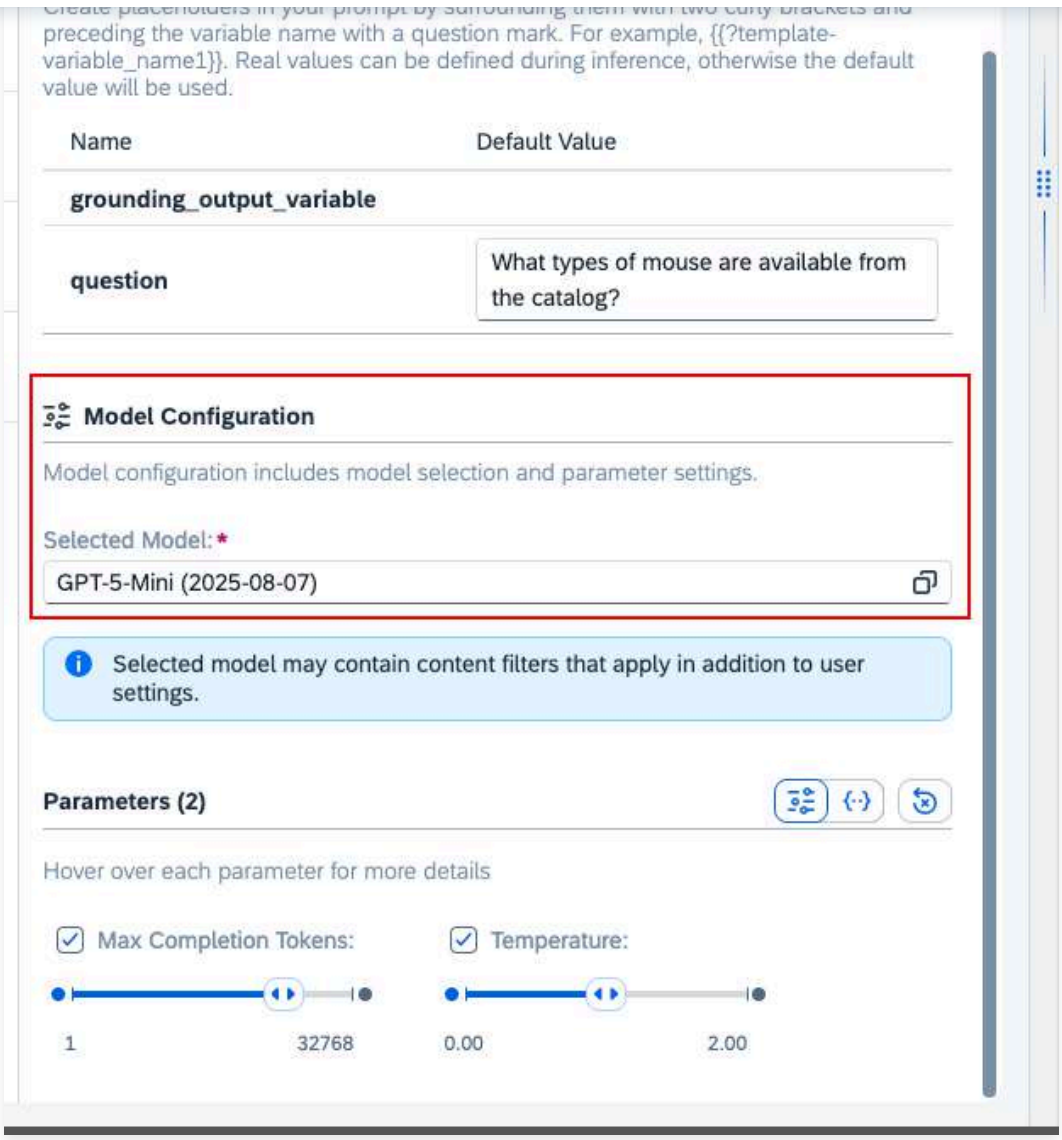
7. Observe that the *Grounding Management* module has now been activated and is enabled. There are also two new additions in the configuration section: An **Output Variable** and a **Data Repository**.



> **Note:** The Data Repository is the vectorized representation of an internal catalog of IT equipment and this forms the basis on which the LLM should formulate its response.
>
> The addition of an output variable tells the LLM to use the information retrieved from the data repository and store it in an output variable which can be used in a template.

Basic Trials

/ Workbooks / SAP Generative AI Hub with AI Launchpad, Orchestration and Document ...



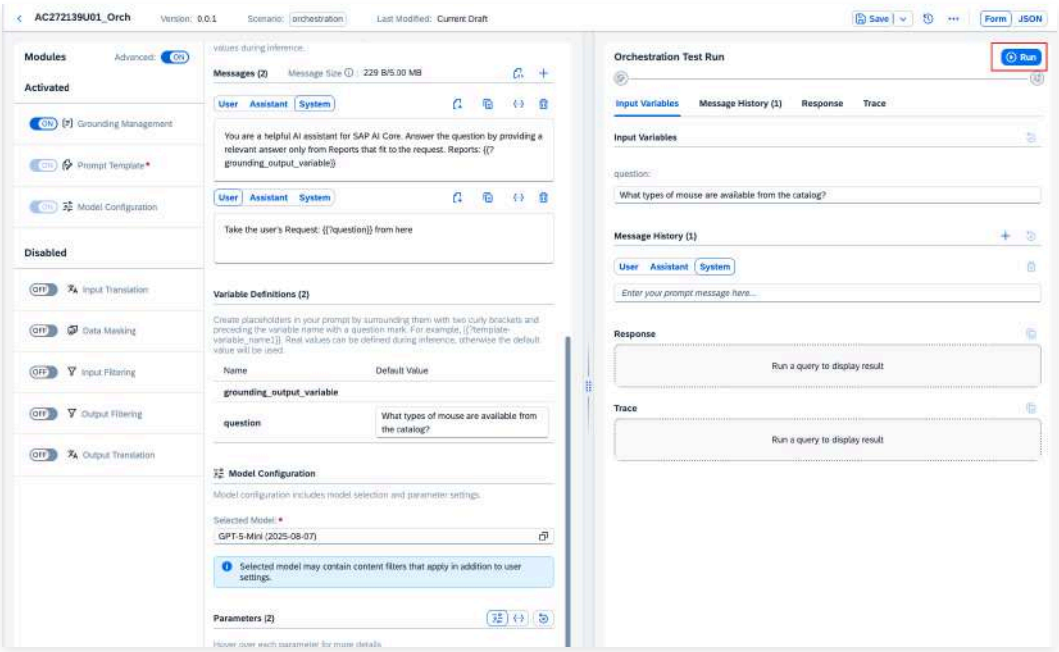
This is because the JSON configuration file has a section in which you can define the model and its parameters. In fact, there are many different options available to define in this configuration file, and it is an effective way of storing different orchestration workflows.

9. Select the JSON icon to view the JSON representation of the orchestration workflow. In particular, observe the addition of the data repository used for grounding:

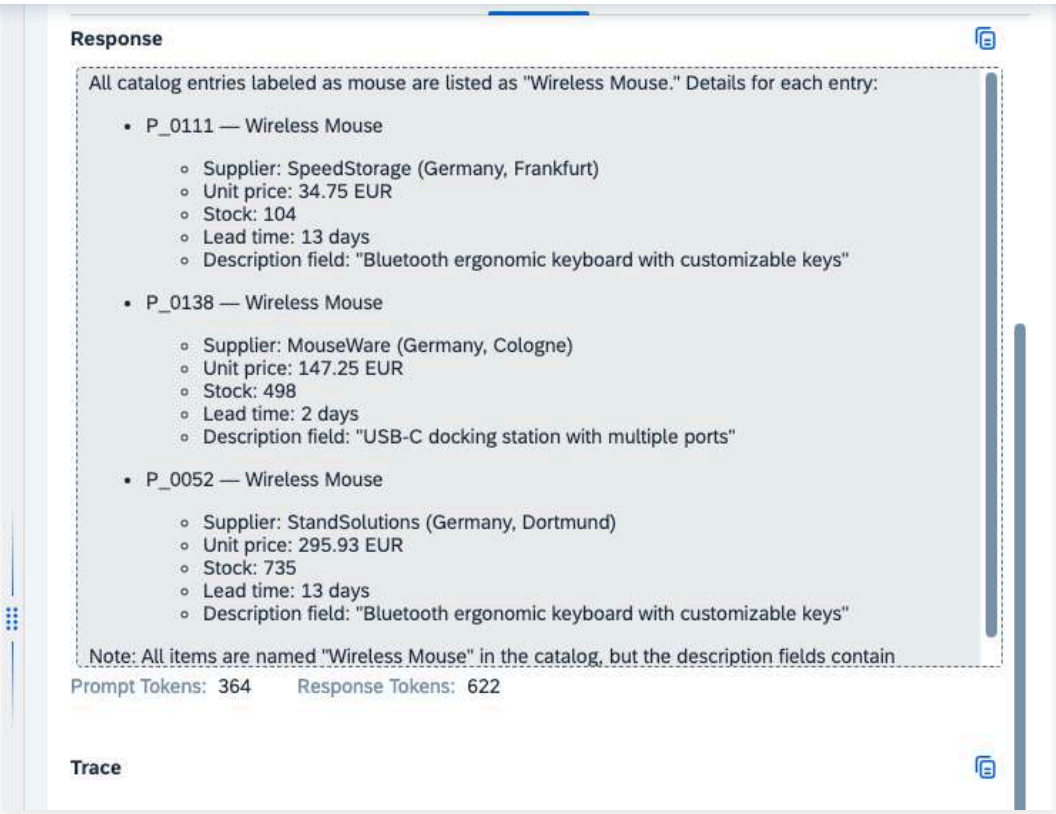
Basic Trials

/ Workbooks / SAP Generative AI Hub with AI Launchpad, Orchestration and Document ...

10. When ready, select the **Run** button again to test the query on a model which is now grounded on the IT equipment data repository and observe the results.



12. This time you should get a more targeted response suggesting the model was much more aware of the context behind the query:



This simplified example showcases how the grounding of a foundation model before running queries enhances its performance and reliability by aligning it with specific context and domain knowledge. This process involves customizing and fine-tuning the model using relevant data, which improves its understanding of nuanced language and concepts pertinent to specific applications.

Benefits include:

- Increased accuracy
- More relevant query results
- Reduced biases
- Better handling of specialized vocabularies

Ultimately, grounding ensures that the model's outputs are more precise and contextually appropriate, leading to higher overall effectiveness in various tasks like information retrieval, decision support, and predictive analytics.

Basic Trials

/ Workbooks / SAP Generative AI Hub with AI Launchpad, Orchestration and Document ...

SAP Business Technology Platform

Introduction to Orchestration

Objectives

After completing this lesson, you will be able to:

Understand SAP Gen AI Hub orchestration service

In the context of AI, orchestration involves coordinating various AI models, data sources, and computational resources to achieve a specific goal.

Key Functions:

- **Templating**
- **Grounding**
- **Translation**
- **Content filtering**
- **Data Masking**
- **Orchestration Workflow**
- **Configuration and Execution**
- **Harmonized API**

Orchestration Workflow

In a basic orchestration setup, you can combine different modules into a pipeline executed with a single API call. Each module's response serves as the input for the next module.

Grounding

Grounding integrates external, contextually relevant, domain-specific, or real-time data into AI processes. This data supplements the natural language processing capabilities of pre-trained models, which are trained on general material.

Grounding is a service designed to handle data-related tasks, for example grounding and retrieval, using vector databases.

The Pipeline API is proxied through the SAP AI Core generative AI hub, and incorporates vector stores, such as the managed SAP HANA Cloud database.

Grounding provides specialized data retrieval through vector databases, grounding the retrieval process using your own external and context relevant data. Grounding combines generative AI capabilities with the capacity to use real-time, precise data to improve decision-making and business operations, for specific business AI driven solutions.

Templating

The templating module is mandatory. It enables you to compose prompts and define placeholders. It then generates the final query that is sent to the model configuration module. Any placeholders that you define in your prompt can be filled at runtime. For example, the following template places the input text into the **{{ ?input }}** placeholder, for example:

```
"templating_module_config": {
  "template": [
    {
      "role": "user",
      "content": "{{ ?input }}"
    }
  ]
}
```

The **{{ ?input }}** placeholder is filled using the contents of the **input_params** input parameter:

You can also set default values for the placeholders. For example, to request either 5 paraphrases or a user-specific number of paraphrases for a given phrase, you can use the following configuration:

```
"templating_module_config": {
  "template": [
    {
      "role": "user",
      "content": "Create {{ ?number }} paraphrases
    }
  ],
  "defaults": {
    "number": 5
  }
},
//... other configuration
"input_params": {
  "number": "3",
  "phrase": "Please respond as soon as possible, th
}
```

As you will see in the next lesson, it's possible to upload a pre-defined template using a JSON config file, or via the Prompt Registry .

Data Masking

The data masking module is optional. It anonymizes or pseudonymizes personally identifiable information from the input.

- Anonymized data can not be unmasked as information about the original data is not retained. Personally identifiable information is replaced with a **MASKED_ENTITY** placeholder.
- Pseudonymized data can be unmasked in the response. personally identifiable information is replaced with a MASKED_ENTITY_ID placeholder.

> The masking service can mask personally identifiable information in the prompt. However, because it is using automated detection mechanisms, there is no guarantee that all personally identifiable information will be found and masked.
>
> Anonymization, replaces personally identifiable information in an irreversible manner. This results in a loss of context, which may limit the capability of the LLM to process the input.
>
> For example, if asked to write a story about Ben and Anna, with anonymization of profile-person, the LLM would be asked to write a story about **MASKED_PERSON** and **MASKED_PERSON** and could no longer distinguish between the two.

Input Filtering

The content filtering module is optional. It lets you filter the input and output based on content safety criteria.

The module supports the Azure Content Safety classification service. This service recognizes four distinct content categories: Hate, Violence, Sexual, and SelfHarm. For more information, see Harm categories in Azure AI Content SafetyInformation published on non-SAP site. Text can have more than one label (for example, a text sample can be classified as both Hate and Violence). The returned content categories include a severity level rating of 0, 2, 4, or 6. The value increases with the severity of the content.

If no content filter is configured in the orchestration configuration, the input is passed to the model configuration without filtering. Similarly, the response is returned without filtering. If no severity levels are set in the configuration (i.e. the config key is not given), the default level of 2 (low) is used for all categories.

Model Configuration

- Model Name: The name of the model to be used.
- Model Parameters: The parameters to be applied to the model.
- Model Version: The version of the model to be used, which can either be latest or a specific version number.

Output Filtering

The content filtering module is optional. It lets you filter the input and output based on content safety criteria.

The module supports the Azure Content Safety classification service. This service recognizes four distinct content categories:

- Hate
- Violence
- Sexual
- SelfHarm.

If no content filter is configured in the orchestration configuration, the input is passed to the model configuration without filtering. Similarly, the response is returned without filtering. If no severity levels are set in the configuration (i.e. the config key is not given), the default level of **2 (low)** is used for all categories.

Harmonized API

The harmonized API lets you use different foundation models without the need to change the client code. It does so by taking the OpenAI API as the standard and mapping other model APIs to it. This includes standardizing message formats, model parameters, and response formats. The harmonized API is integrated into the templating module, the model configuration, and the orchestration response.

In the OpenAI format, a prompt and its response can contain a list of messages with a role and content. The role can be system, user, or assistant and the content is the text of the message. The response can contain a list of choices with **index**, **messages**, and a **finish_reason**.

For more information see Harmonized API