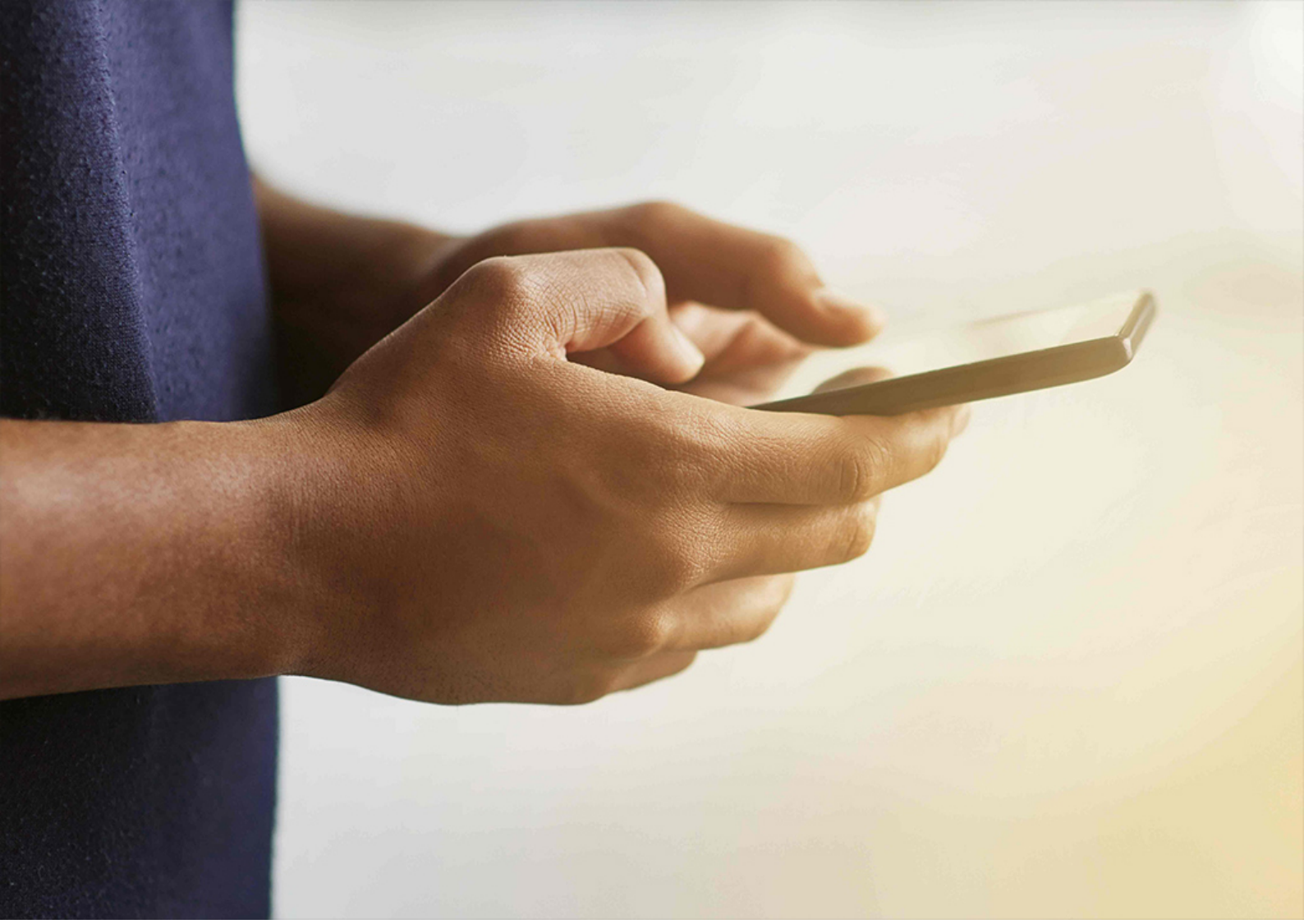


INTERNAL
2023-12-20

Generative AI Hub with SAP AI Core

THE BEST RUN SAP

Content

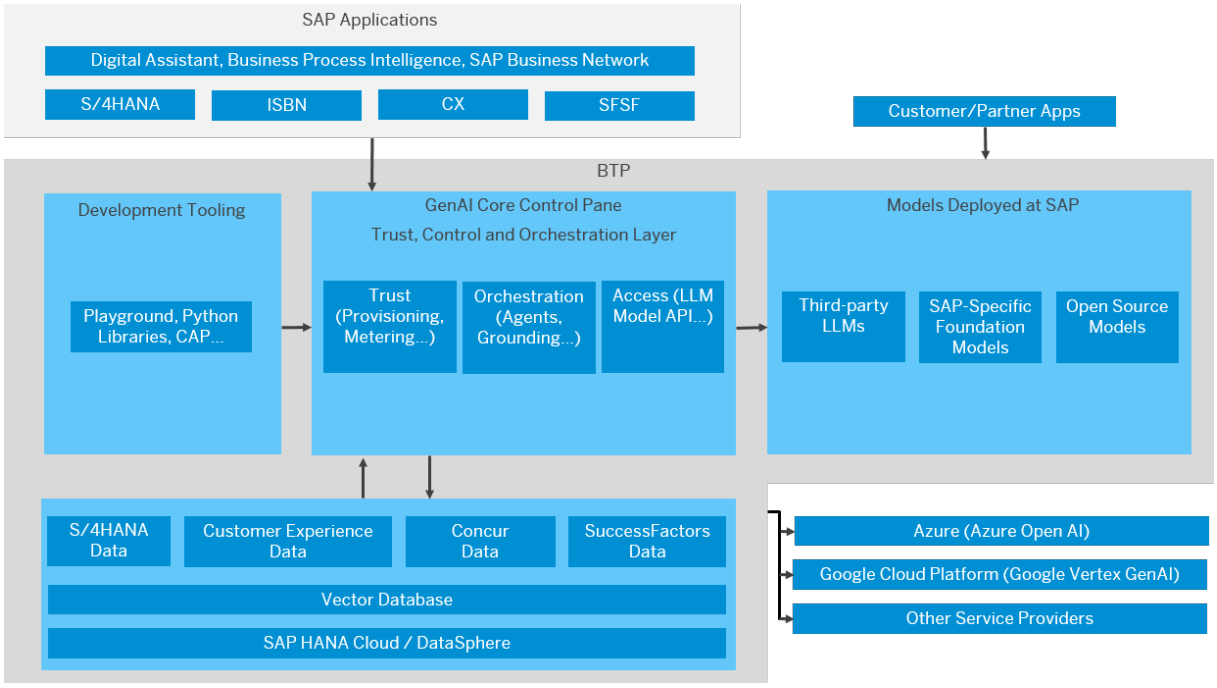
1 Generative AI Hub in SAP AI Core

The Generative AI hub incorporates generative AI into your AI activities in SAP AI Core and SAP AI Launchpad.

LLMs are self-supervised, deep learning models that have been trained on vast amounts of unlabeled data. They leverage AI technology and industrial-scale computational resources to learn complex language patterns and semantic knowledge bases for natural language processing (NLP) tasks. They parse input, such as prompts, and by predicting a target word, can return contextually relevant responses written in natural language. A single LLM can perform multiple NLP tasks by using different input formats and output modes.

LLMs are general models but can be fine-tuned with additional embeddings for specialized or domain-specific use cases.

SAP AI Core and the Generative AI hub help you to integrate LLMs and AI into new business processes in a cost-efficient manner.



Generative AI Hub Architecture Overview

2 Service Plans

The SAP AI Core service plan you choose determines pricing, conditions of use, resources, available services, and hosts. The Generative AI hub in SAP AI Core is available only through the `sap-internal` service plan.

> ⚠ Caution
>
> The `sap-internal` service plan is available only for internal consumption and on eu10 canary.

The `sap-internal` service plan enhances the capabilities provided in the `standard` service plan. Specifically, it provides access to the `foundation-models` global AI scenario. This scenario, which is managed by SAP AI Core, includes serving templates for deployments with integrated LLM access.

If you're new to SAP AI Core, choose the `sap-internal` service plan during your initial setup. For more information, see Add a Service Plan for Generative AI to the Global Account [page 11].

If you already have an SAP AI Core tenant on a `standard` or `free` tier service plan, you can update the service plan to `sap-internal`. For more information, see Update an Existing Plan of SAP AI Core to Include Generative AI [page 12].

> → Tip
>
> If you update to the `sap-internal` service plan, you can still use your original service key.

2.1 Resource Plans for Large Language Models

You can configure SAP AI Core to use different infrastructure resources for different tasks. SAP AI Core provides several preconfigured infrastructure bundles called "resource plans" for this purpose.

> i Note
>
> These resource plans are internal and should not be shared outside of the organisation or used for productive use with external customers.

You can choose from the following GPU resource plans to deploy large language models (LLMs) in SAP AI Core:

Resource Plans in SAP AI Core

Resource Plan ID	GPUs	CPU Cores	Memory GBs	Code to Allocate Resources in Workflow Templates
Infer2-L	1 a10g	15	57.9	`ai.sap.com/resourcePlan: infer2.l`

Resource Plan ID	GPUs	CPU Cores	Memory GBs	Code to Allocate Resources in Workflow Templates
`Infer2-4XL`	4 a10g	47	182.7	`ai.sap.com/resourcePlan: infer2.4xl`
`Train2-8XL`	8 a100	95	1118.7	`ai.sap.com/resourcePlan: train2.8xl`
`Train2-8XXL`	8 a100	95	1118.7	`ai.sap.com/resourcePlan: train2.8xxl`

The capacity units for the GPU resource plans to deploy LLMs are as follows:

Capacity Units in SAP AI Core

Resource Plan ID	Capacity Units (Billable Units per Hour)
Infer2-L	3.6917
`Infer2-4XL`	11.9707
`Train2-8XL`	67.1944
`Train2-8XXL`	82.8221

> **i** Note
>
> For information about resource plans that do not pertain to large language models, see Choose a Resource Plan in SAP AI Core.

2.2 Models and Scenarios in the Generative AI Hub

Scenarios

Access to the large language models is provided under the global AI scenario `foundation-models`, which is managed by SAP AI Core. Individual models are provided as executables in the form of serving templates, and accessed by choosing the corresponding template for the desired model.

The following scenarios are available:

Scenario Number	Global Scenario	Executable ID	Description
1	foundation-models	azure-openai	The Azure OpenAI Service provides REST API access to OpenAI's LLMs.

Scenario Number	Global Scenario	Executable ID	Description
2	foundation-models	aicore-opensource	Opensource models hosted and accessed via SAP AI Core.

Models

The following models are supported:

Executable ID	Model Name	Model Version	Deprecation (as Specified by Model Provider)	Region	Request Limit (Requests per Minute)
azure-openai	gpt-35-turbo	0613	2025-07-05	• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	120
azure-openai	gpt-35-turbo-16k	0613	2024-15-01	• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	96
azure-openai	gpt-4	0613	2024-15-01	• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	18
azure-openai	gpt-4-32k	0613	2024-15-01	• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	78
azure-openai	text-embedding-ada-002	2	2025-02-02	• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	138
aicore-opensource	tiiuae--falcon-40b-instruct			• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	138

> **i Note**
>
> Rate limits are applied at tenant level.

> **i Note**
>
> In addition to the generally available models, there are experimental and preview models maintained by IES. Experimental and preview models have their own data guidelines which differ from those for generally available models from SAP AI Core.
>
> The guidelines for experiemental and preview models are:
>
> - You can send **public data** to all models.
> - You can send **internal data** to all models, except to those in 'preview'.
> - You cannot save any prompt where **confidential** data is sent to the model.
> - You should never send **personal data** to any model. This includes but is not limited to, SAP customer data, personal data of SAP customers and personal data of SAP employees.

Models from Azure OpenAI are accessed through a private instance of the `chat-completions API`. For more information, see Azure OpenAI Chat Completions API Documentation .

Open Source models are hosted by SAP AI Core and can be accessed via OpenAI compatible API schema.

For more information on the Generative AI Hub in SAP AI Core, see the SAP AI Core documentation.

> **i Note**
>
> The following topics are out of the scope of this document:
>
> - Advanced consumption patterns such as working with a textual knowledge base (such as embeddings)
> - Complex orchestration of LLM calls
> - Training own models

Related Information

Azure Chat Completions Documentation
Tiiuae Falcon 40b Instruct Documentation

2.3 Metering and Pricing for the Generative AI Hub

The use of large language models (LLMs) in the Generative AI hub is metered using GenAI tokens and capacity units.

The Generative AI Hub is available only as part of the Extended service plan.

A GenAI token corresponds to a block of 1,000 tokens from the LLM service provider. Its cost varies depending on the model used and the type of token (input or output).

A capacity unit is the number of GenAI tokens multiplied by a fixed amount, and is used to calculate the monetary value of your LLM usage. The fixed amount is currently 2.6925 (subject to change).

> **i Note**
>
> Prices on this page are internal and should not be shared outside of the organisation or used for productive use with external customers.

The following table provides the conversion rates between tokens from the LLM service provider and GenAI tokens. The rates apply to blocks of 1,000 input and output tokens. You can refer to these values to calculate the total number of GenAI tokens that you consume. You then multiply the number of consumed tokens by the fixed-rate capacity unit to obtain the monetary value.

> **i Note**
>
> Values indicated are subject to change.

Model	GenAI Input Tokens (for 1,000 Tokens)	GenAI Output Tokens (for 1,000 Tokens)
GPT-35-Turbo	0.00094	0.00122
GPT-35-Turbo-16K	0.00180	0.00238
GPT-4	0.01735	0.03462
GPT-4-32K	0.03462	0.06917
text-embedding-ada-002	0.00013	0.00000
tiiuae--falcon-40b-instruct	0.00045	0.00081

> **Example**
>
> This example uses the GPT-35-Turbo model. For a given request, x input tokens are consumed and y output tokens are consumed. The corresponding metrics are:
>
> GenAI tokens (nonbillable metric) = (x/1000) * 0.00094 + (y/1000) * 0.00122
> Capacity units (billed amount) = GenAI tokens * 2.6925

Charges associated with use of SAP AI Core may also apply. For more information, see SAP AI Core Metering and Pricing.

LLM Metering for AI Unit Consumption

AI Units

SAP's portfolio of AI offerings and products can meter usage in AI units using the Unified Metering service.

An AI unit is a commercial construct that covers the costs for all AI-enabled features and offerings and that helps to provide a unified customer experience.

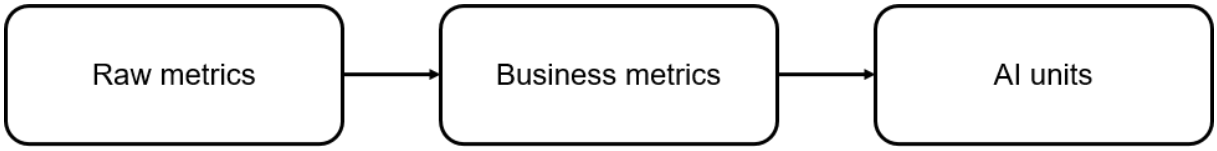
Customers should be able to see their current usage of AI units and their balance on a dashboard in the SAP4ME portal.

The notion of AI units requires a specific business metric from the application/service provider. Business metrics can be thought of as business objects or entities that act as a unit from a consumption or business

value perspective. AI units can be converted to/from based on the business metric through the billing component.

AI Units for GenAI Scenarios

Applications introducing GenAI features can use capabilities from the Generative AI hub and use it to meter their usage. Typical flow of billing in AI Units will involve:



Business metrics are converted to AI units in a central billing component using metadata from a central system of record. The logic for converting raw metrics to business metrics (for example, the number of input/output tokens to the number of chat sessions) must be provided by the application/service provider.

Applications that use capabilities from the Generative AI hub directly can report business metrics to the Unified Metering service:

1. Application teams maintain the rules for calculating business metrics in the Generative AI hub.
2. The teams submit the required metadata along with API requests (such as `/chat/completions`) as request headers to the Generative AI hub.
3. The Generative AI hub calculates the business metric and reports the resulting metric to the Unified Metering services.

Billing Metadata

Applications can send metadata along with API requests to the Generative AI hub using these headers:

- `X-USECASE-ID`
  This header maps to the business metric name. This is an identifier of the business metric, from which the generative AI use case can be derived, and which is maintained in an SAP global system of record.
- `X-BUSINESS-CONTEXT`
  This header maps to the business context. Additional business context is added as additional dimensions to the metering record.
- `X-LOCALTENANT-ID`
  This header maps to the tenant ID. This can either be a local tenant ID or a global tenant ID. The local tenant ID is later mapped to a global tenant ID, following the unified services concept. The commercial mapping is done centrally by the Unified Services, which maps the LoB's proprietary tenant (also known as the “local tenant”) and the customer’s data (also known as the “business metadata”), thereby linking the tenant to the actual business entity of the customer in SAP (the SAP CRM tenant).
- `X-PRODUCT-TYPE`
  This header maps to the tenant product type. Together with the tenant ID, this identifies a tenant globally. The mapping follows this approach and thus requires service metadata to be maintained.

In addition, applications can send an event ID, which is an application-determined identifier. If an event ID is provided and requests already exist with the same event ID, the Generative AI hub reports metrics only for the latest request. This prevents metrics reports from being duplicated due to the application triggering numerous retries.

2.4 Add a Service Plan for Generative AI to the Global Account

Generative AI models are available only with the `sap-internal` or `extended` service plan, which you configure in your global account during the initial setup phase.

Prerequisites

- You're a new user of SAP AI Core.

Context

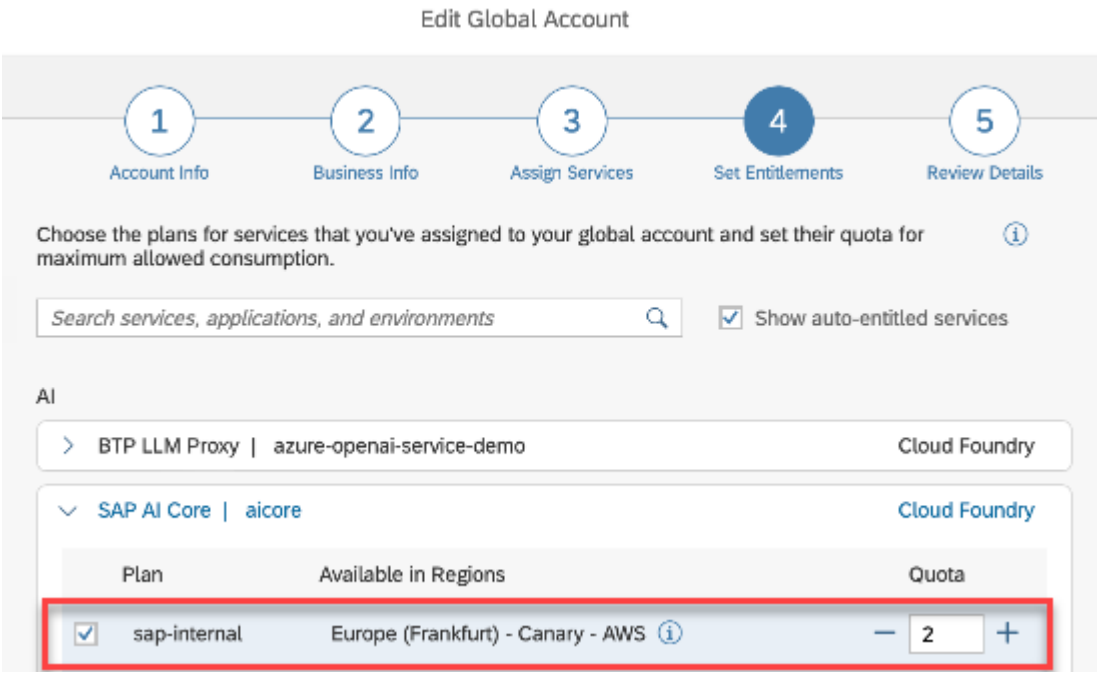
To add the `sap-internal` or `extended` plan, set the quota in your global account.

> **i** Note
>
> The sap-internal plan is internal and should not be shared outside of the organisation or used for productive use with external customers.

Procedure

Complete the initial setup for SAP AI Core as described in Add a Service Plan for Generative AI to the Global Account [page 11]. When you set your entitlements, include the `sap-internal` or `extended` service plan and set the applicable quota.

> **i** Note
>
> - The `sap-internal` and `extended` plans include the features of the `standard` plan, with the addition of the generative AI capabilities. If `sap-internal` or `extended` is included, it is not necessary to have a `standard` or `free` instance.
> - The `sap-internal` plan is internal and should not be shared outside of the organisation or used for productive use with external customers. For productive use for an external customer, choose the `extended` plan. For internal use with canary in region EU10, choose the `sap-internal` plan.



2.5 Update an Existing Plan of SAP AI Core to Include Generative AI

Generative AI models are available only with the `sap-internal` or `extended` service plan. You can switch service plans by updating your instance of SAP AI Core.

Prerequisites

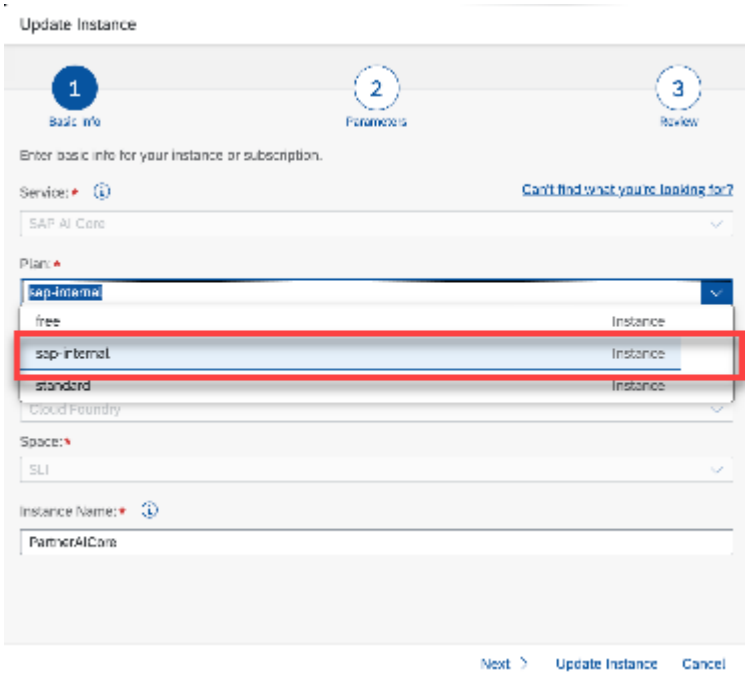
You're an existing user of SAP AI Core.

> **!** Restriction
>
> - If you have an existing `sap-internal` or `extended` plan, it is not possible to update to `standard` or `free`.
> - If you have an existing `standard` plan, it is **not** possible to create a new instance with the `sap-internal` or `extended` plan. You must update your standard instance to `sap-internal` or `extended` .
> - The sap-internal plan is internal and should not be shared outside of the organisation or used for productive use with external customers.

Procedure

1. In your SAP BTP cockpit, choose *Instances and Subscriptions*.
2. Under *Environments*, choose *Update* and add the `sap-internal` or `extended` service plan.

   The `sap-internal` plan is internal and should not be shared outside of the organisation or used for productive use with external customers. For productive use for an external customer, choose the `extended` plan. For internal use with canary in region EU10, choose the `sap-internal` plan.



3 Tutorials

The activation and consumption steps are also available online in a step by step tutorial.

For more information, see Prompt LLMs in the Generative AI Hub in SAP AI Core and SAP AI Launchpad.

4 Activating the Generative AI Hub

4.1 Create a Deployment for a Generative AI Model in SAP AI Core

You make a generative AI model available for use by creating a deployment. You can do so one time for each model and model version, and for each resource group that you want to use with Generative AI Hub. The deployment URL that is generated can be reused.

Prerequisites

- You have an SAP AI Core service instance and service key. For more information, see SAP AI Core Initial Setup Documentation.
- You're using the `sap-internal` or `extended` service plan. For more information, see Service Plans [page 5] and .
- You have completed the client authorization for your preferred user interface. For more information, see Use a Service Key in SAP AI Core.

Context

You make a model available for use by creating a deployment. You can do so one time for each model and model version. The model deployment includes the `modelName` and `version` of the model you want to access. After the deployment is complete, you have a `deploymentUrl`, which can be used across your organization to access the model version.

Using the API

Procedure

1. Decide which LLM you want to deploy and note the following information:
   - Executable ID
   - Model name

- Model version

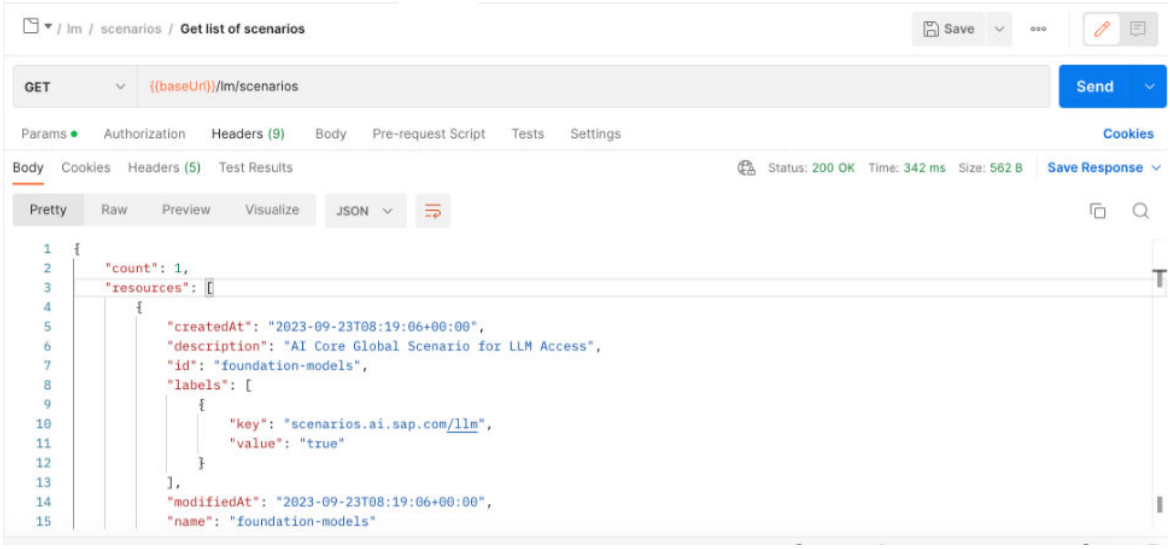
> **i Note**
>
> - Instead of specifying a model version, using “latest” will use the latest version of the model available in SAP AI Core.
> - Where model version is not listed, it is not applicable.

Executable ID	Model Name	Model Version	Deprecation (as Specified by Model Provider)	Region	Request Limit (Requests per Minute)
azure-openai	gpt-35-turbo	0613	2025-07-05	• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	120
azure-openai	gpt-35-turbo-16k	0613	2024-15-01	• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	96
azure-openai	gpt-4	0613	2024-15-01	• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	18
azure-openai	gpt-4-32k	0613	2024-15-01	• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	78
azure-openai	text-embedding-ada-002	2	2025-02-02	• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	138
aicore-open-source	tiiuae--falcon-40b-instruct			• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	138

2. Check that you have access to the scenario containing generative AI by sending a GET request to `{{apiurl}}/v2/lm/scenarios`.

   Set the *Authorization* header with **`Bearer $TOKEN`** and set your resource group.

> **i Note**
>
> You must use the same resource group for all of your generative AI activities. To use a different resource group, these steps must be repeated for each resource group.



The scenarios listed contain a scenario with the id `foundation-models`.

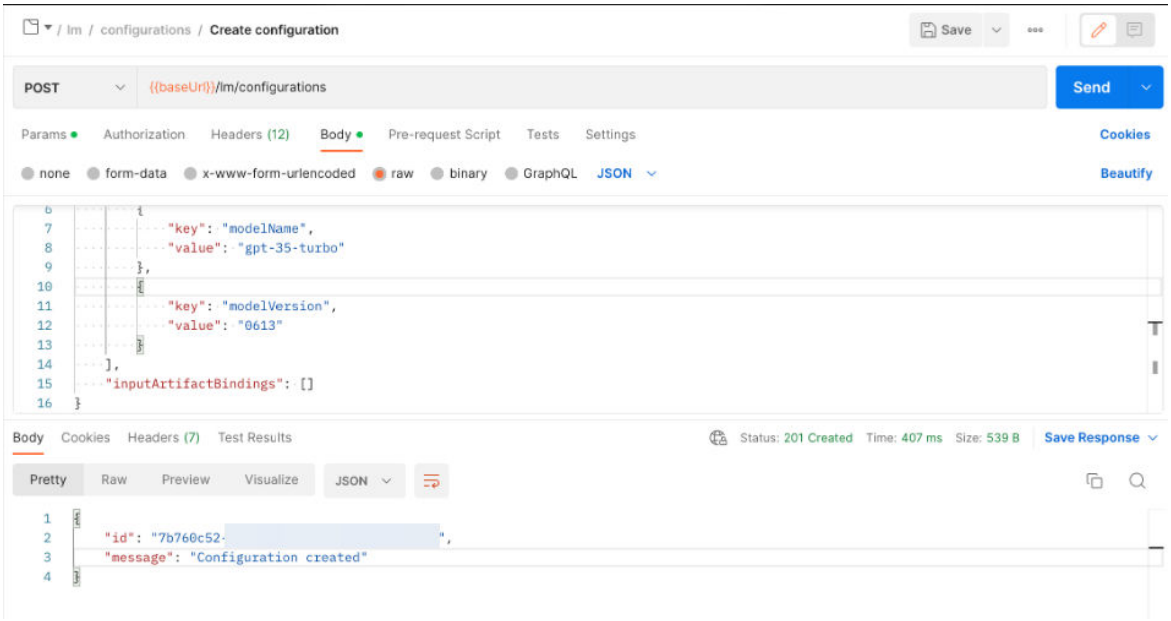
3. Create a configuration by sending a POST request to the endpoint `{{apiurl}}/v2/lm/configurations`.

   Include details of the model to which you want to provide access by passing in the following parameters:

   - `name` is your free choice of identifier.
   - `executableId`, `modelName`, and `modelVersion` are provided in the table above.
   - `scenarioId` must be `foundation-models`.
   - `versionId` is your own version reference.

   Sample Code

```
{
    "name": "yourNameChoice",
    "executableId": "azure-openai",
    "scenarioId": "foundation-models",
    "versionId": "0.0.1",
    "parameterBindings": [
        {
            "key":"modelName",
            "value":"gpt-35-turbo"
        },
        {
            "key": "modelVersion",
            "value": "0613"
        }
  ],
    "inputArtifactBindings": []
}
```



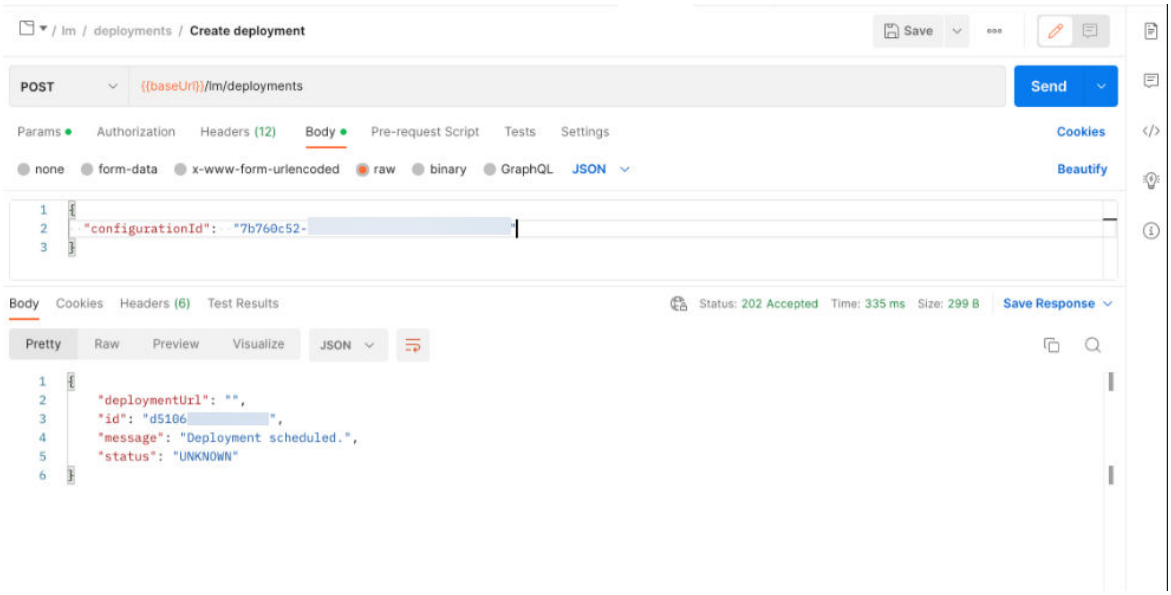
> → Tip
>
> You can specify the value **`latest`** for the `modelVersion` to use the most recent model version available in SAP AI Core.

You receive a unique `configuationId` in the response.

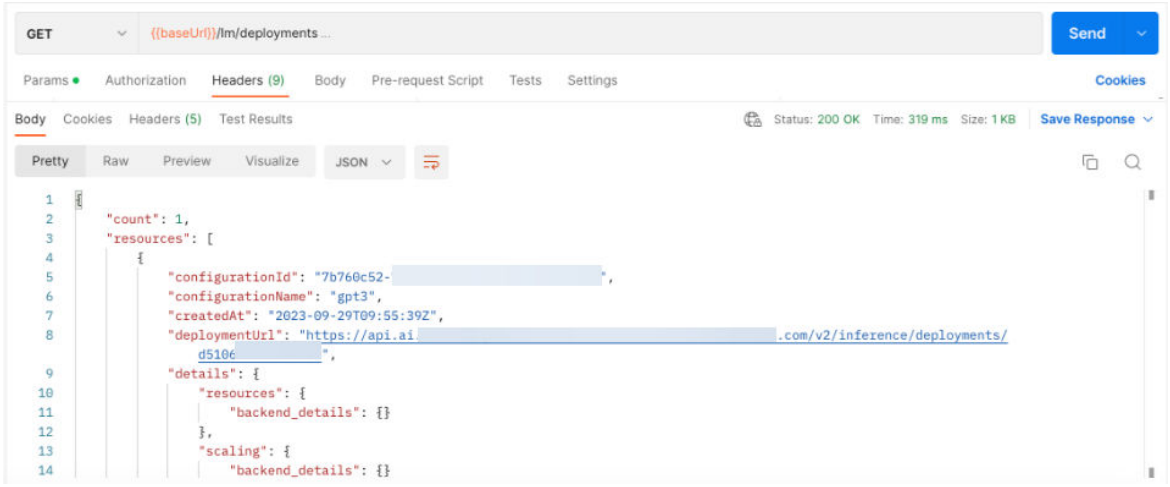
4. Create a deployment by sending a POST request to the endpoint `{{apiurl}}/v2/lm/deployments`.

   Include the `configuationId` from the previous step in your request.

> ⁖≡ Sample Code
>
> ```
> {
>  "configurationId": "yourConfigurationId"
> }
> ```



5. Retrieve the details of your deployment by sending a GET request to the endpoint`{{apiurl}}/v2/lm/deployments`.



Next Steps

When the deployment is running, the model can be accessed using the `deploymentUrl` provided in the response. For more information, see Consume Generative AI Models Using SAP AI Core [page 25].

4.2 Create a Deployment for a Generative AI Model in SAP AI Launchpad

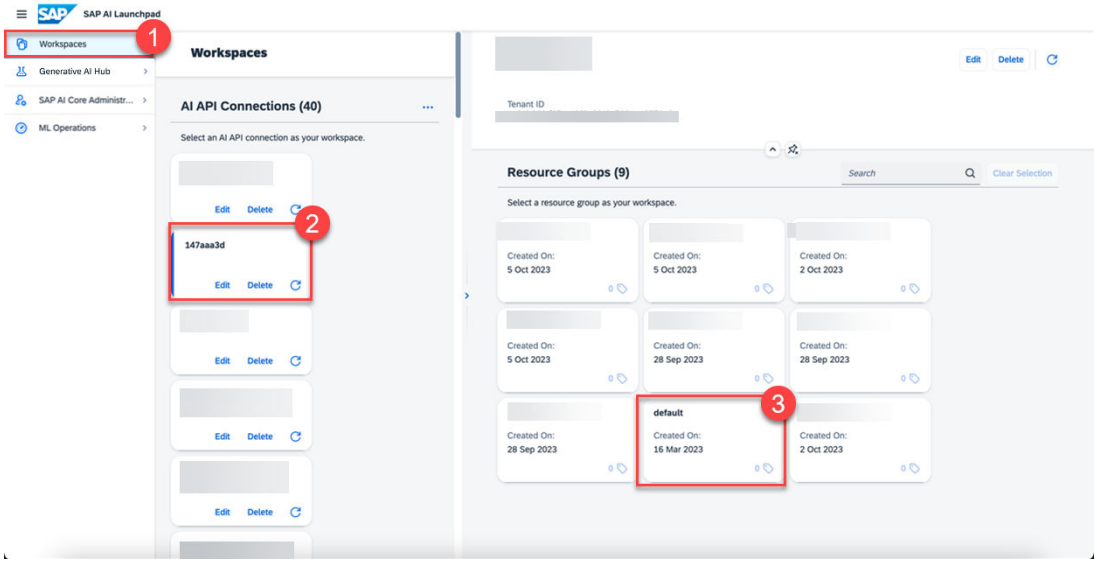
You make a model available for use by creating a deployment. You can do so one time for each model and model version, and for each resource group that you want to use with Generative AI Hub.

Prerequisites

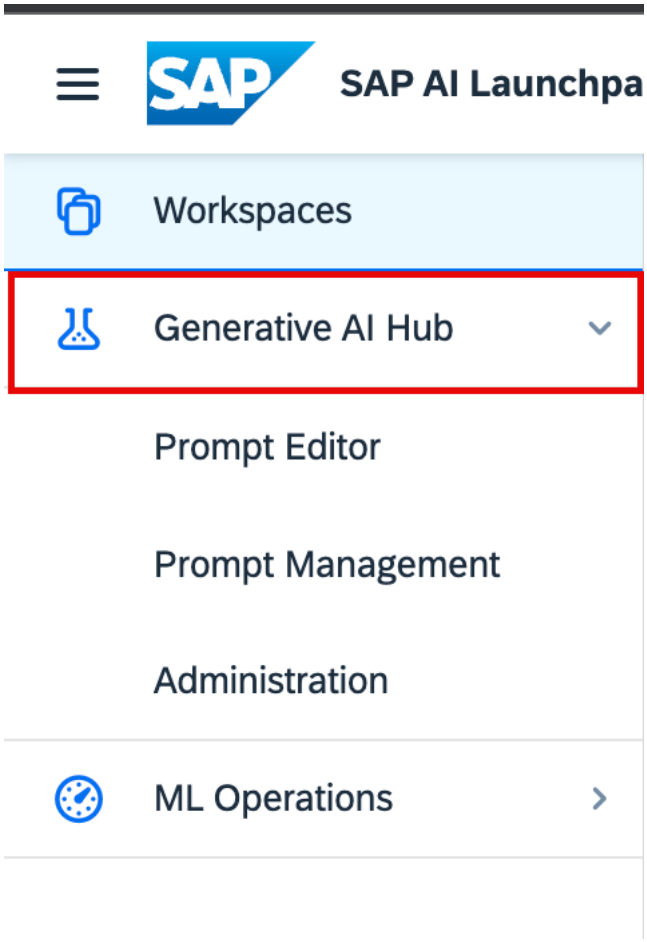
- You have an SAP AI Launchpad service instance and service key. For more information, see SAP AI Launchpad Initial Setup Documentation.
- You're using the `sap-internal` or `extended` service plan. For more information, see Service Plans [page 5].
- You have either the `mloperations_editor` or `scenario_deployment_editor` role, or you are assigned a role collection that contains one of these roles. For more information, see Roles and Authorizations.
- You have completed the client authorization for your preferred user interface. For more information, see Use a Service Key in SAP AI Core.

Procedure

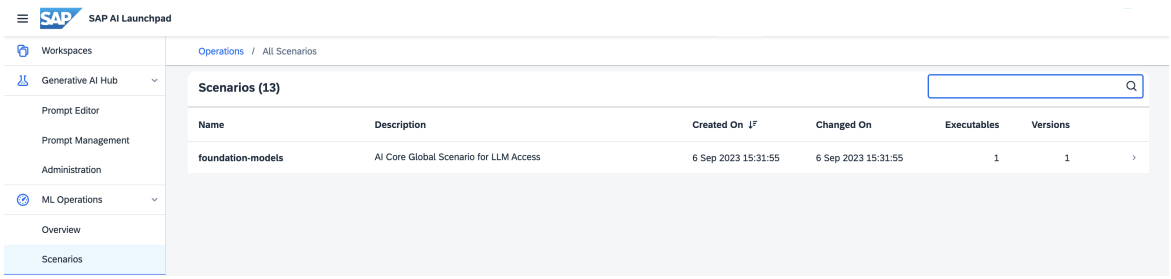
1. Select the connection to your SAP AI Core runtime in the *Workspaces* app and choose a resource group.



The *Generative AI Hub* app is now clickable in your side navigation panel and resource groups are listed.

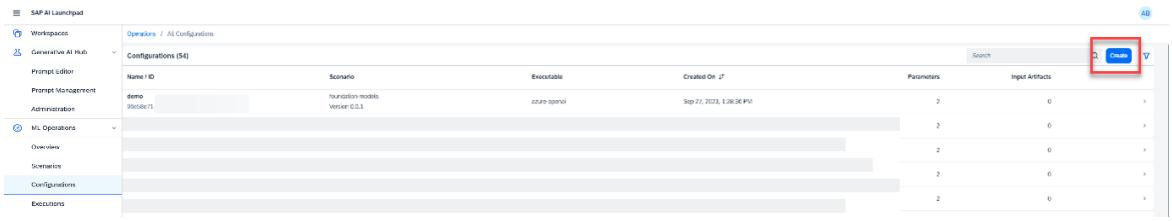


2. In the *ML Ops* app, choose *Scenarios* and check that a scenario called *foundation-models* scenario exists.

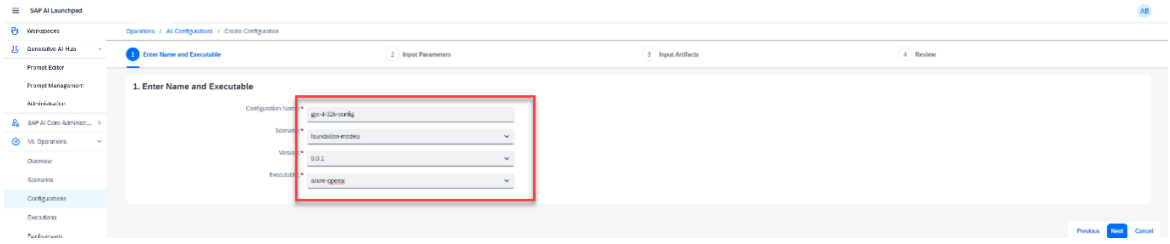


If the *foundation-models* scenario is not available, check your SAP AI Core tenant service plan.

3. In the *ML Ops* app, choose *Configurations* and click *Create*.



4. Enter a name for your configuration, choose the *foundation-models* scenario, enter a version number, and select the executable for your chosen model provider.

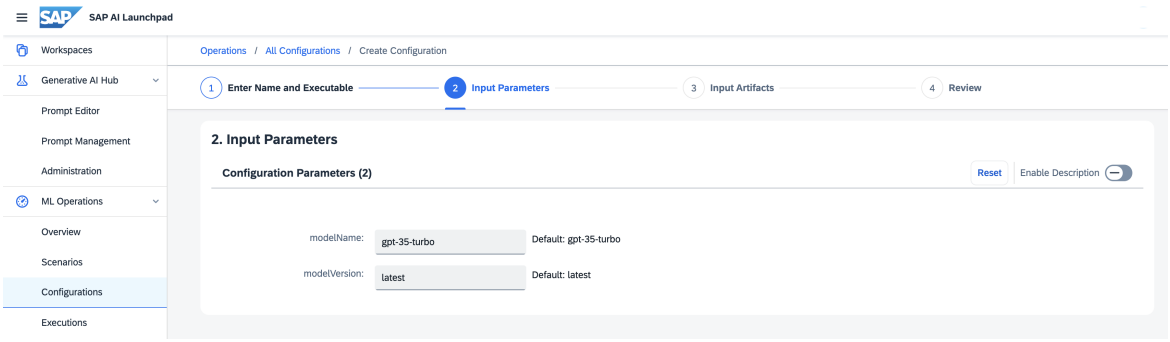


5. Enter the name and version (if applicable) of the model that you want to use.

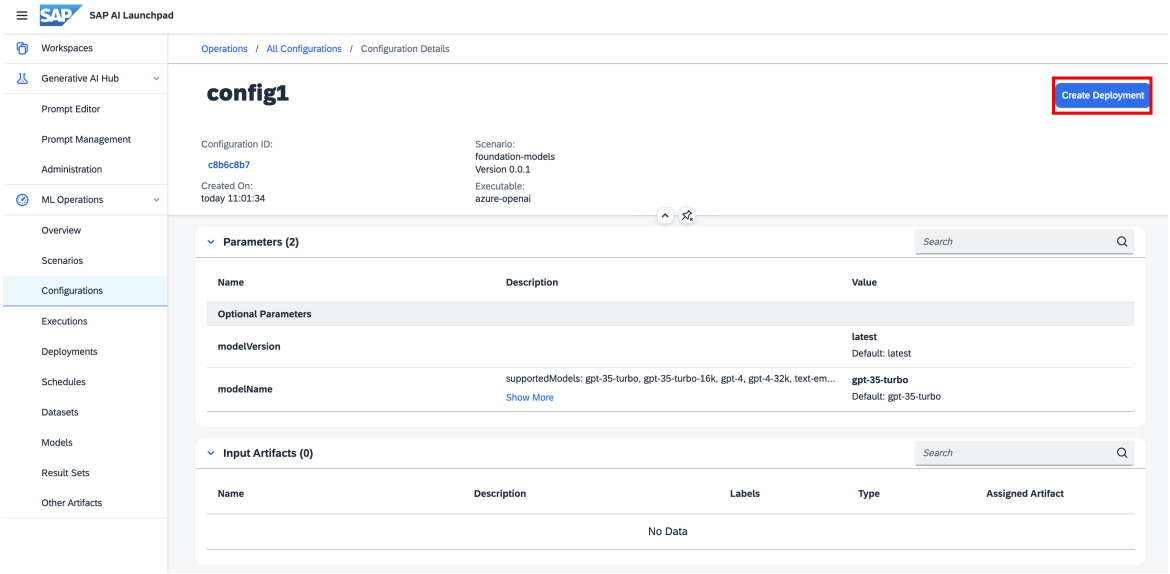
   The following models are available:

Executable ID	Model Name	Model Version	Deprecation (as Specified by Model Provider)	Region	Request Limit (Requests per Minute)
azure-openai	gpt-35-turbo	0613	2025-07-05	• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	120
azure-openai	gpt-35-turbo-16k	0613	2024-15-01	• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	96
azure-openai	gpt-4	0613	2024-15-01	• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	18
azure-openai	gpt-4-32k	0613	2024-15-01	• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	78
azure-openai	text-embedding-ada-002	2	2025-02-02	• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	138
aicore-open-source	tiiuae--falcon-40b-instruct			• US10 (mapped to Azure US East) • EU10 (mapped to Azure EU Central)	138

> **i Note**
>
> - Instead of specifying a model version, using “latest” will use the latest version of the model available in SAP AI Core.
> - Where model version is not listed, it is not applicable.



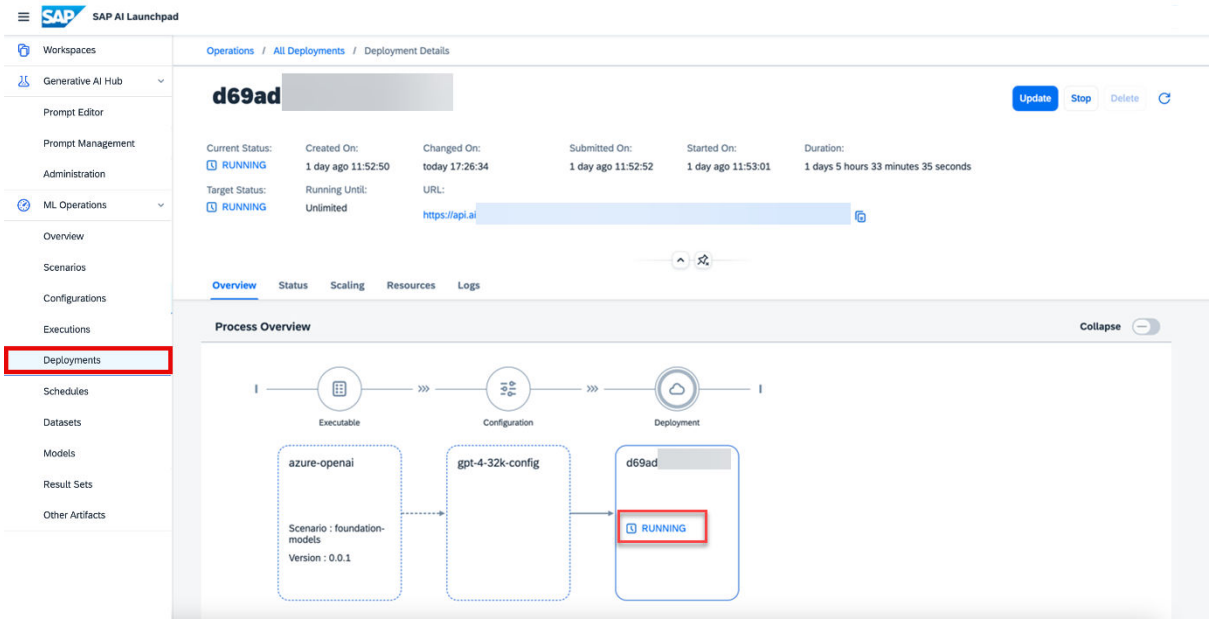
6. After you've created your configuration, select *Create Deployment*.



> **i Note**
>
> You must use the same resource group for all of your generative AI activities. To use a different resource group, these steps must be repeated for each resource group.

Next Steps

When the deployment is running, the model can be accessed using the *Generative AI Hub* app. For more information, see Prompt Experimentation [page 46].



If you want to remove a model, delete its deployment.

4.3 Model Lifecycle

Model versions have deprecation dates. Where a model version is specified in a deployment, the deployment will stop working on the deprecation date of that model version.

Implement one of the following model upgrade options:

- **Auto Upgrade:** Create or patch a generative AI configuration with `modelVersion latest`. When a new `modelVersion` is supported by SAP AI Core, existing generative AI deployments will automatically use the latest version of the given model.
- **Manual Upgrade:** Patch n generative AI configuration with your chosen replacement `modelVersion`. This model version will be used in generative AI deployments irrespective of updates to the models supported by SAP AI Core.

> **i** Note
>
> If `modelVersion` isn't specified, it will be `latest` by default. To upgrade manually, you **must** specify a `modelVersion`.

5 Consume Generative AI Models Using SAP AI Core

You consume a generative AI model by sending a request to the endpoint `{{deploymentUrl}}/chat/completions?api-version={{api-version}}`. LLMs can carry out natural language-related tasks such as answering questions, summarizing text, and extracting information from a body of text.

Prerequisites

- You have the deployment URL for your generative AI model. For more information, see Create a Deployment for a Generative AI Model in SAP AI Core [page 15].

Context

The body of your request must include the `messages` parameter that defines your query.

Ensure that you have the following headers set:

Header	Value
Authorization	Bearer $TOKEN
AI-Resource-Group	The resource group used in the activation steps

You can also include optional parameters such as:

Models	Parameters
Azure	• `max_tokens`: An integer that defines the maximum number of tokens allowed for the generated answer. The default value is 4,096. • `temperature`: A number between 0 and 2. Higher values make the output more random; lower values make it more focused and deterministic. • `frequency_penalty`: A number between -2.0 and 2.0. Positive values penalize new tokens based on their existing frequency in the text so far, decreasing the model's likelihood to repeat the same line verbatim. • `presence_penalty`: A number between -2.0 and 2.0. Positive values penalize new tokens based on whether they appear in the text so far, increasing the model's likelihood to talk about new topics. • `stop`: A string or array. Up to four sequences. If the output generates one of the stop values, it will stop generating content. • `stream`: A boolean value. True provides streaming support for LLM deployments, false does not. For more information, see Azure Chat Completions Documentation .
Falcon	• `max_tokens`: An integer that defines the maximum number of tokens allowed for the generated answer. The default value is 4,096. • `temperature`: A number between 0 and 2. Higher values make the output more random; lower values make it more focused and deterministic. • `frequency_penalty`: A number between -2.0 and 2.0. Positive values penalize new tokens based on their existing frequency in the text so far, decreasing the model's likelihood to repeat the same line verbatim. • `presence_penalty`: A number between -2.0 and 2.0. Positive values penalize new tokens based on whether they appear in the text so far, increasing the model's likelihood to talk about new topics. • `stop`: A string or array. Up to four sequences. If the output generates one of the stop values, it will stop generating content. For more information, see Tiiuae Falcon 40b Instruct Documentation .

If you want to remove a model, delete its deployment.

Prompt Examples

The following examples are provided for Postman. To make a prompt request through curl, adapt the following code:

```
curl --location '$DEPLOYMENT_URL/chat/completions?api-version=<yourVersion>' \
--header 'AI-Resource-Group: <Resource Group Id>' \
--header 'Content-Type: application/json' \
--header "Authorization: Bearer $TOKEN" \
--data '{
```

```
    "messages": [
        {
        "role": "user",
        "content": "sample input prompt"
        }
    ],
    "max_tokens": 100,
    "temperature": 0.0,
    "frequency_penalty": 0,
    "presence_penalty": 0,
    "stop": "null"
}'
```

Summarizing

You can provide the LLM with a text and ask for a summary of it.

Procedure

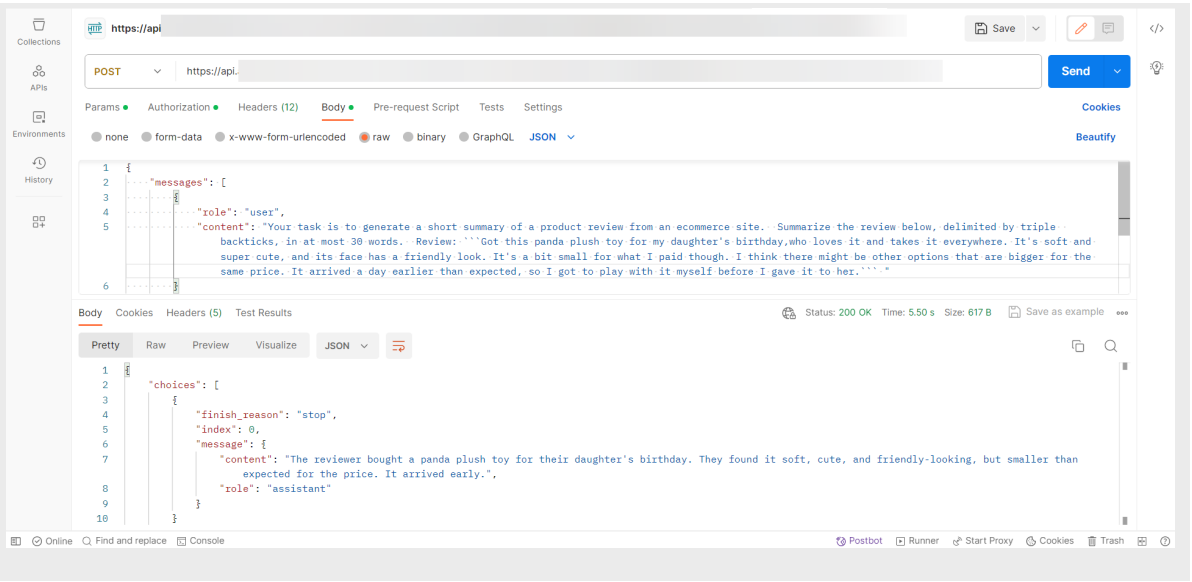
Send a POST request to the endpoint `{{deploymentUrl}}/chat/completions?api-version={{api-version}}`.

Include your query in the body. Mark the text to be summarized with triple back ticks (`).

> **i** Note
>
> Where the model does not have a version, the version is not needed in the endpoint.

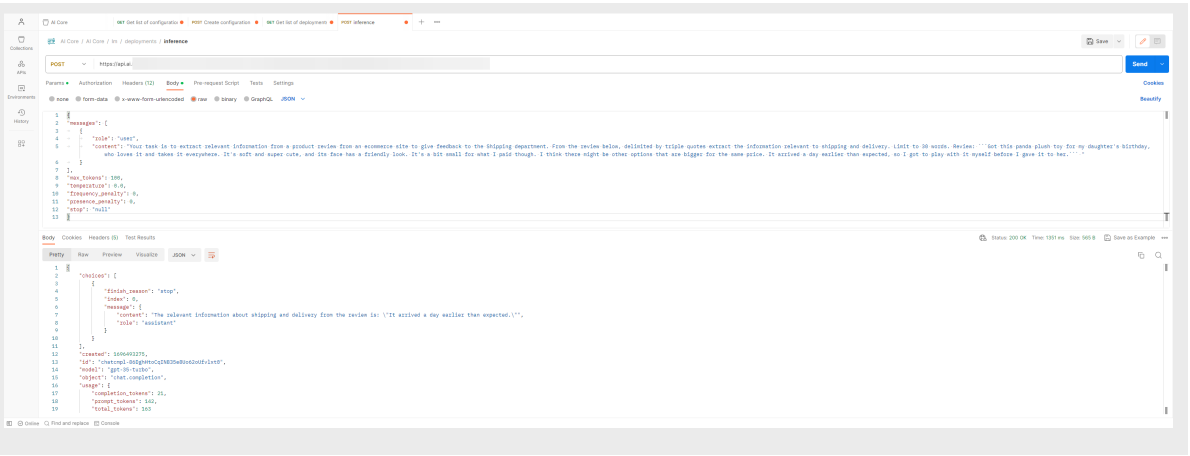
> **✤** Example
>
> This example generates a summary of a product review. Summaries can include topics that aren't related to the main topic.

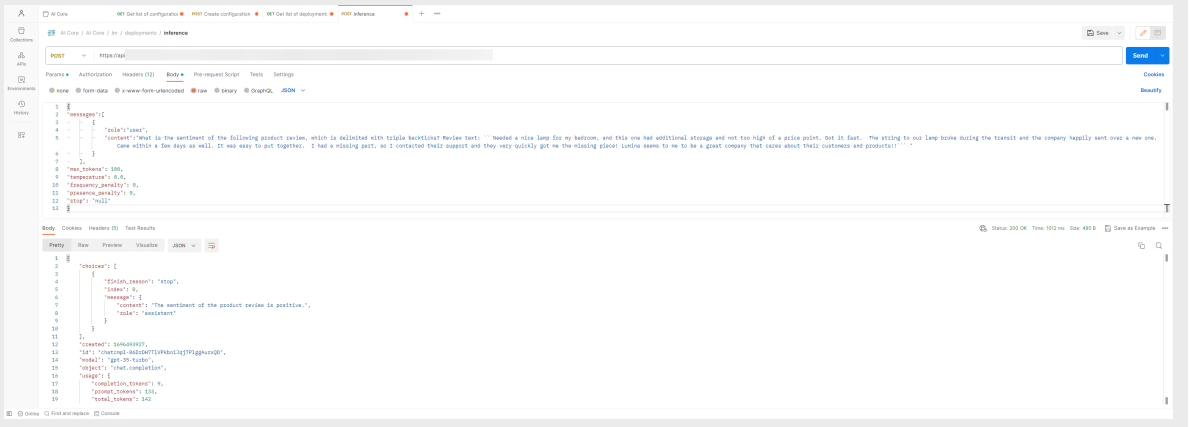
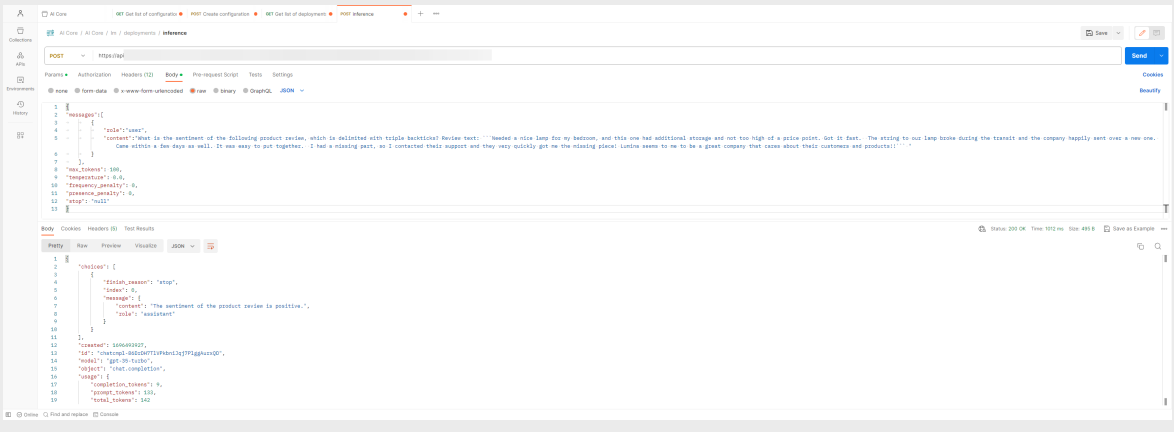
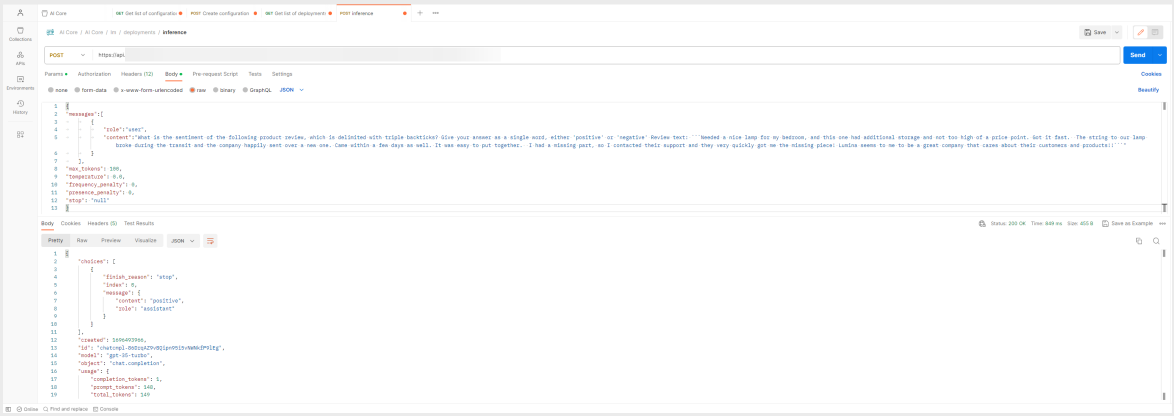
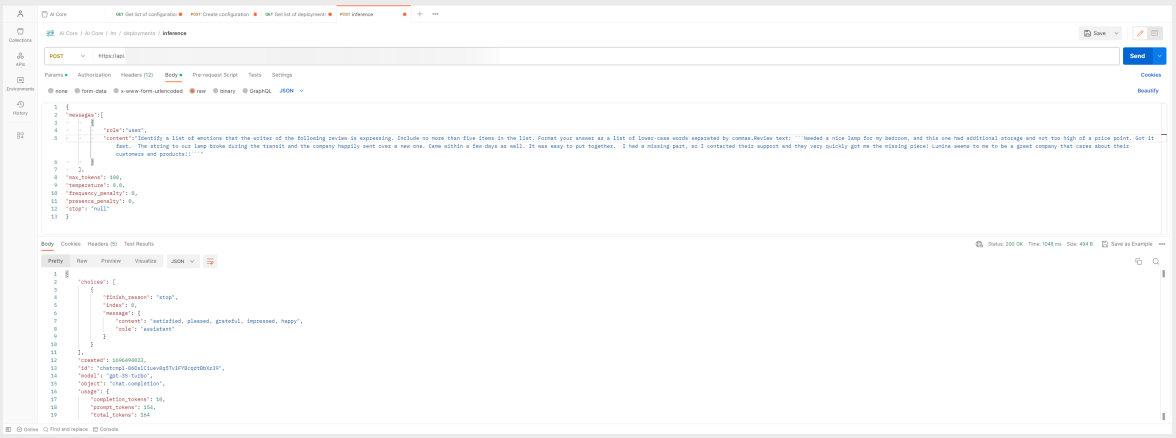
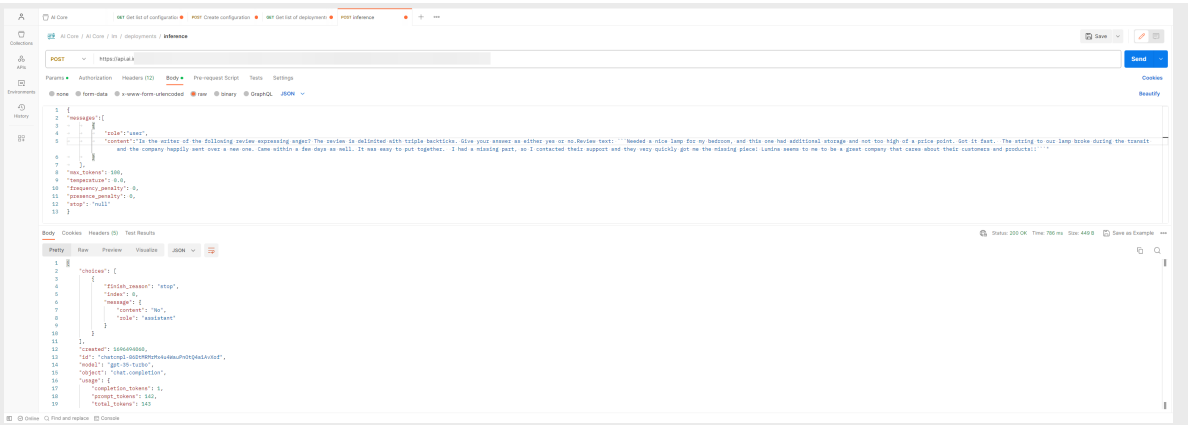
```
{
"messages": [
    {
        "role": "user",
        "content": "Your task is to generate a short summary of a product
review from an ecommerce site. Summarize the review below, delimited by
triple backticks, in at most 30 words. Review:```Got this panda plush toy for
my daughter's birthday, who loves it and takes it everywhere. It's soft and
super cute, and its face has a friendly look. It's a bit small for what I
paid though. I think there might be other options that are bigger for the
same price. It arrived a day earlier than expected, so I got to play with it
myself before I gave it to her. ```"
    }
],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```

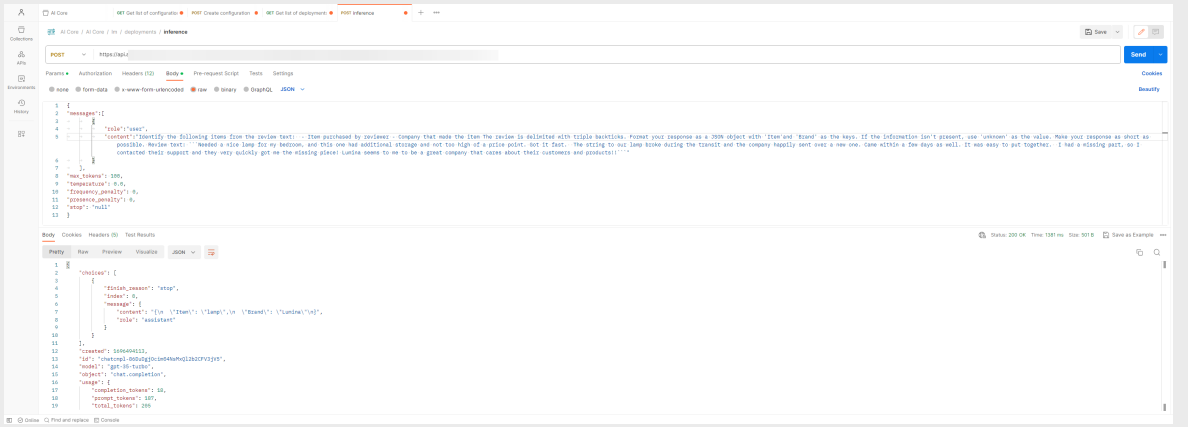


Example

In this example, the prompt is similar but has been refined by adding the intended recipient of the feedback (the purchasing department) and the reason for requesting it (to determine the price of the product).

```
{
"messages": [
    {
        "role": "user",
        "content": "Your task is to generate a short summary of a product
review from an ecommerce site to give feedback to the pricing department,
responsible for determining the price of the product. Summarize the review
below, delimited by triple backticks, in at most 30 words, and focusing
on any aspects that are relevant to the price and perceived value. Review:
```Got this panda plush toy for my daughter's birthday,who loves it and takes
it everywhere. It's soft and super cute, and its face has a friendly look.
It's a bit small for what I paid though. I think there might be other options
that are bigger for the same price. It arrived a day earlier than expected,
so I got to play with it myself before I gave it to her.``` "
    }
],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```
```



Example

This example uses “extract” instead of “summarize”.

```
{
"messages": [
    {
        "role": "user",
        "content": "Your task is to extract relevant information from a
product review from an ecommerce site to give feedback to the Shipping
department. From the review below, delimited by triple backticks extract the
information relevant to shipping and delivery. Limit to 30 words. Review:
```Got this panda plush toy for my daughter's birthday,who loves it and takes
it everywhere. It's soft and super cute, and its face has a friendly look.
It's a bit small for what I paid though. I think there might be other options
that are bigger for the same price. It arrived a day earlier than expected,
so I got to play with it myself before I gave it to her.``` "
    }
],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```

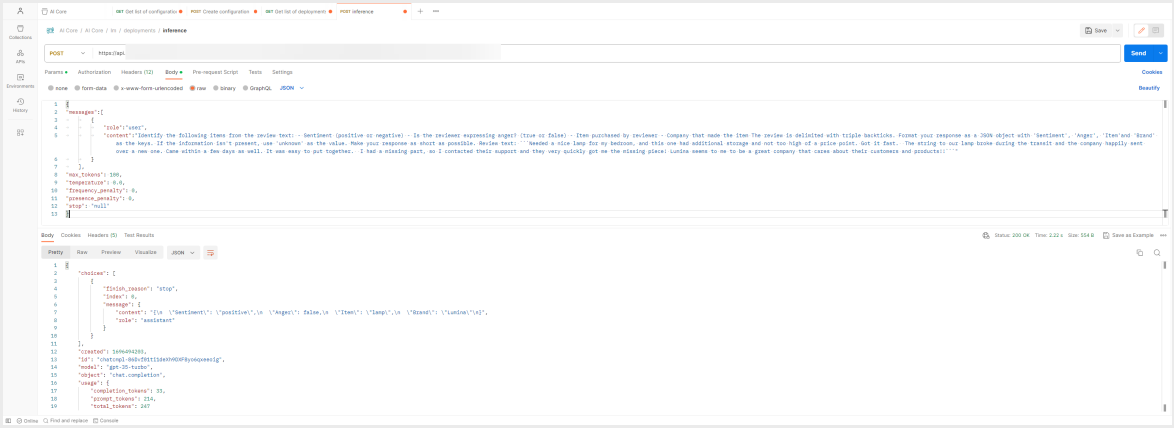

```

Inferencing

Inferencing uses the information in a given text to draw a conclusion.

Procedure

Send a POST request to the endpoint `{{deploymentUrl}}/chat/completions?api-version={{api-version}}`.

Include your query in the body. Mark the text to be inferenced with triple back ticks (`).

Example

This example performs a sentiment analysis on a product review.

```
{
"messages":[
        {
            "role":"user",
            "content":"What is the sentiment of the following product review,
which is delimited with triple backticks? Review text: ```Needed a nice lamp
for my bedroom, and this one had additional storage and not too high of a
price point. Got it fast.  The string to our lamp broke during the transit
and the company happily sent over a new one. Came within a few days as well.
It was easy to put together.  I had a missing part, so I contacted their
support and they very quickly got me the missing piece! Lumina seems to me to
be a great company that cares about their customers and products!!``` "
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```



Example

This example generates the sentiment as a one word response.

```
{
"messages":[
        {
            "role":"user",
            "content": "What is the sentiment of the following product
review, which is delimited with triple backticks? Give your answer as a
single word, either 'positive' or 'negative' Review text: ```Needed a nice
lamp for my bedroom, and this one had additional storage and not too high of
a price point. Got it fast.  The string to our lamp broke during the transit
and the company happily sent over a new one. Came within a few days as well.
It was easy to put together.  I had a missing part, so I contacted their
support and they very quickly got me the missing piece! Lumina seems to me to
be a great company that cares about their customers and products!!```"
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```



Example

This example analyzes the emotions expressed in the review.

```
{
"messages":[
        {
            "role":"user",
            "content": "Identify a list of emotions that the writer of the
following review is expressing. Include no more than five items in the list.
Format your answer as a list of lower-case words separated by commas.Review
text: ```Needed a nice lamp for my bedroom, and this one had additional
storage and not too high of a price point. Got it fast.  The string to
our lamp broke during the transit and the company happily sent over a new
one. Came within a few days as well. It was easy to put together.  I had
a missing part, so I contacted their support and they very quickly got me
the missing piece! Lumina seems to me to be a great company that cares about
their customers and products!!```"
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
```

```
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```



Example

This example detects whether anger is present.

```
{
"messages":[
        {
            "role":"user",
            "content": "Is the writer of the following review expressing
anger? The review is delimited with triple backticks. Give your answer as
either yes or no.Review text: ```Needed a nice lamp for my bedroom, and this
one had additional storage and not too high of a price point. Got it fast.
The string to our lamp broke during the transit and the company happily sent
over a new one. Came within a few days as well. It was easy to put together.
I had a missing part, so I contacted their support and they very quickly got
me the missing piece! Lumina seems to me to be a great company that cares
about their customers and products!!```"
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```



Example

This example detects product and company names from the customer review.

```
{
"messages":[
        {
            "role":"user",
            "content": "Identify the following items from the review text:
- Item purchased by reviewer - Company that made the item The review is
delimited with triple backticks. Format your response as a JSON object
with 'Item'and 'Brand' as the keys. If the information isn't present, use
'unknown' as the value. Make your response as short as possible. Review text:
```Needed a nice lamp for my bedroom, and this one had additional storage and
not too high of a price point. Got it fast.  The string to our lamp broke
during the transit and the company happily sent over a new one. Came within
a few days as well. It was easy to put together.  I had a missing part, so
I contacted their support and they very quickly got me the missing piece!
Lumina seems to me to be a great company that cares about their customers and
products!!```"
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```


```

Example

This example performs multiple tasks in a single query.

```
{
"messages":[
        {
            "role":"user",
            "content": "Identify the following items from the review text: -
Sentiment (positive or negative) - Is the reviewer expressing anger? (true or
false) - Item purchased by reviewer - Company that made the item The review
is delimited with triple backticks. Format your response as a JSON object
with 'Sentiment', 'Anger', 'Item'and 'Brand' as the keys. If the information
isn't present, use 'unknown' as the value. Make your response as short as
possible. Review text: ```Needed a nice lamp for my bedroom, and this one
had additional storage and not too high of a price point. Got it fast.  The
string to our lamp broke during the transit and the company happily sent over
a new one. Came within a few days as well. It was easy to put together.  I
had a missing part, so I contacted their support and they very quickly got me
the missing piece! Lumina seems to me to be a great company that cares about
their customers and products!!```"
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```

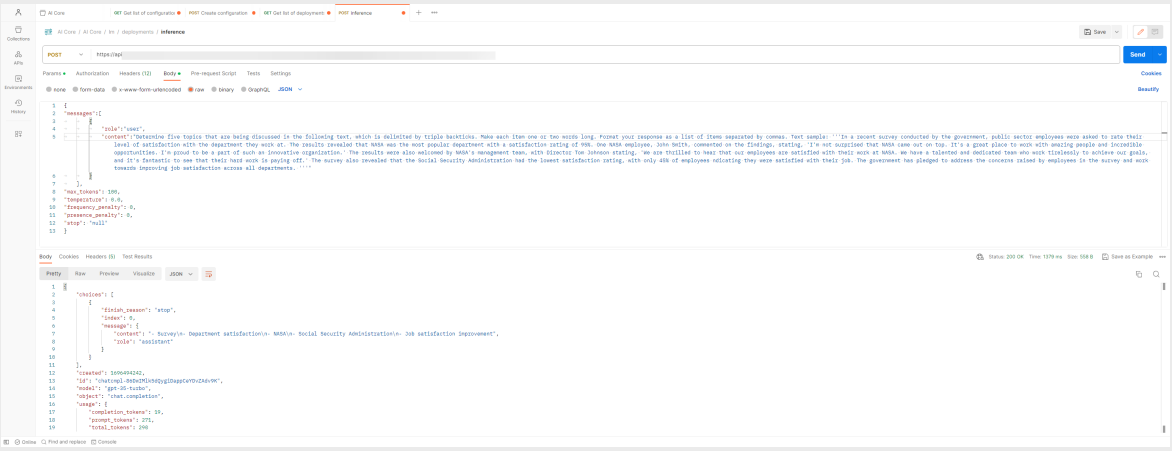


Example

This example identifies the five topics discussed in a story.

```
{
"messages":[
        {
            "role":"user",
            "content": "Determine five topics that are being discussed in the
following text, which is delimited by triple backticks. Make each item one or
two words long. Format your response as a list of items separated by commas.
Text sample: '''In a recent survey conducted by the government, public sector
employees were asked to rate their level of satisfaction with the department
they work at. The results revealed that NASA was the most popular department
with a satisfaction rating of 95%. One NASA employee, John Smith, commented
on the findings, stating, 'I'm not surprised that NASA came out on top. It's
a great place to work with amazing people and incredible opportunities. I'm
```

```
proud to be a part of such an innovative organization.' The results were
also welcomed by NASA's management team, with Director Tom Johnson stating,
'We are thrilled to hear that our employees are satisfied with their work at
NASA. We have a talented and dedicated team who work tirelessly to achieve
our goals, and it's fantastic to see that their hard work is paying off.' The
survey also revealed that the Social Security Administration had the lowest
satisfaction rating, with only 45% of employees ndicating they were satisfied
with their job. The government has pledged to address the concerns raised by
employees in the survey and work towards improving job satisfaction across
all departments. ''''"
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```



Transformations

Transformations transform a given text into another language or register.

Procedure

Send a POST request to the endpoint `{{deploymentUrl}}/chat/completions?api-version={{api-version}}`.

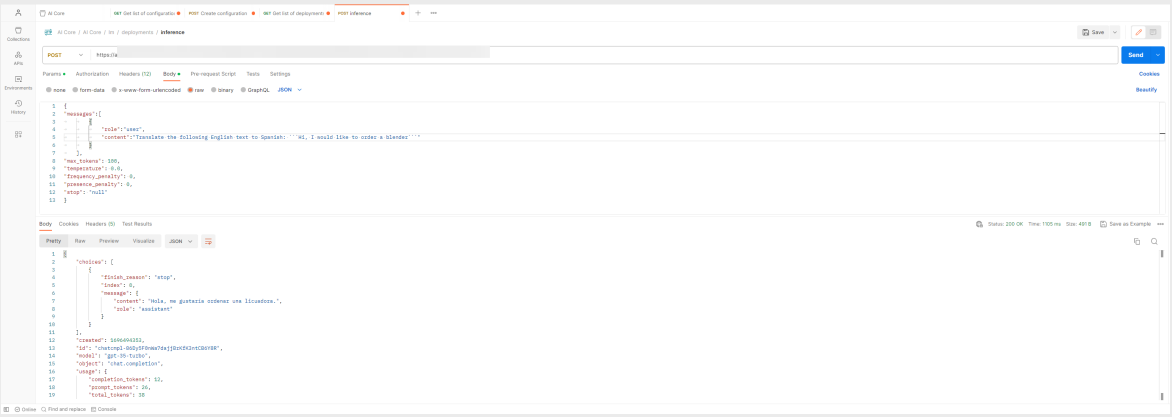
Include your query in the body. Mark the text to be transformed with triple back ticks (`).

Example

This example translates text from English to Spanish.

```
{
"messages":[
```

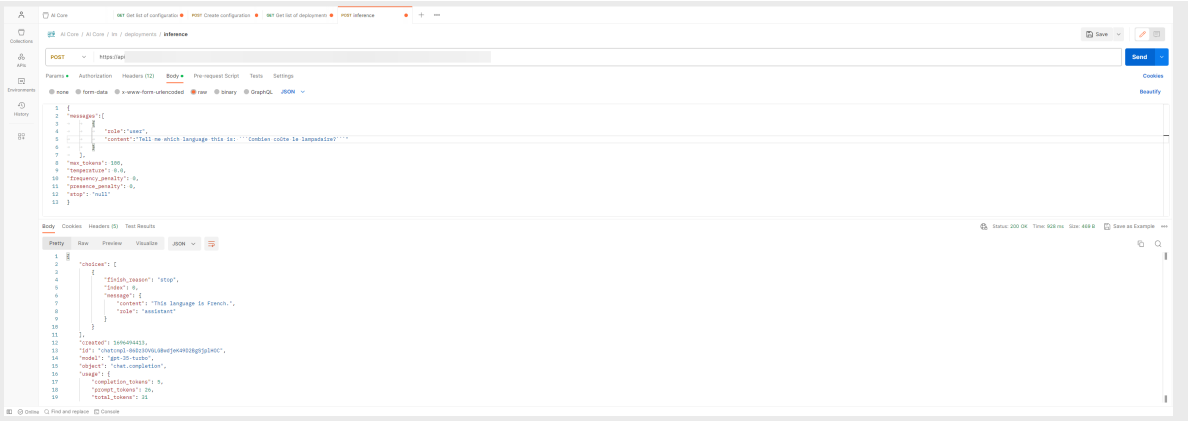
```
        {
            "role":"user",
            "content": "Translate the following English text to Spanish:
```Hi, I would like to order a blender```"
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```



Example

This example detects the language that the text is written in.

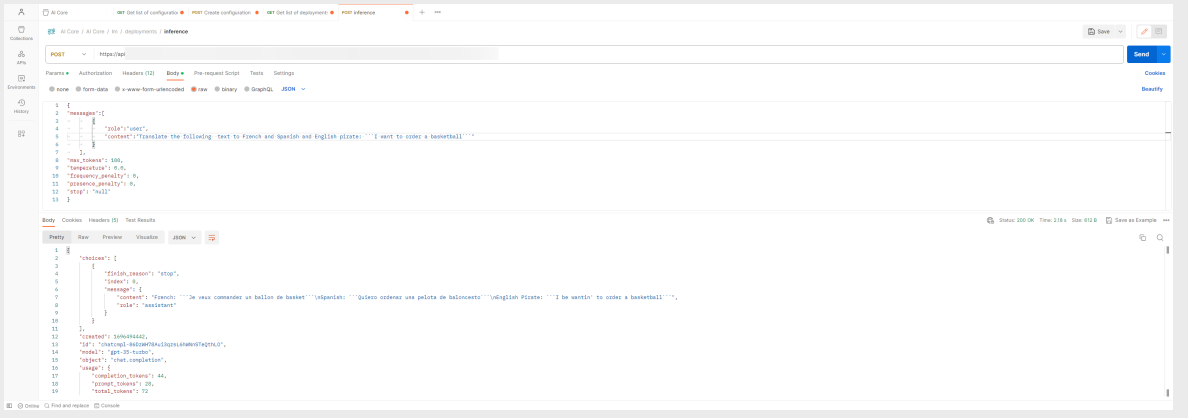
```
{
"messages":[
        {
            "role":"user",
            "content": "Tell me which language this is: ```Combien coûte le
lampadaire?```"
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```
```



Example

This example translates the given text into multiple languages.

```
{
"messages":[
        {
            "role":"user",
            "content": "Translate the following  text to French and Spanish
and English pirate: ```I want to order a basketball```"
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```

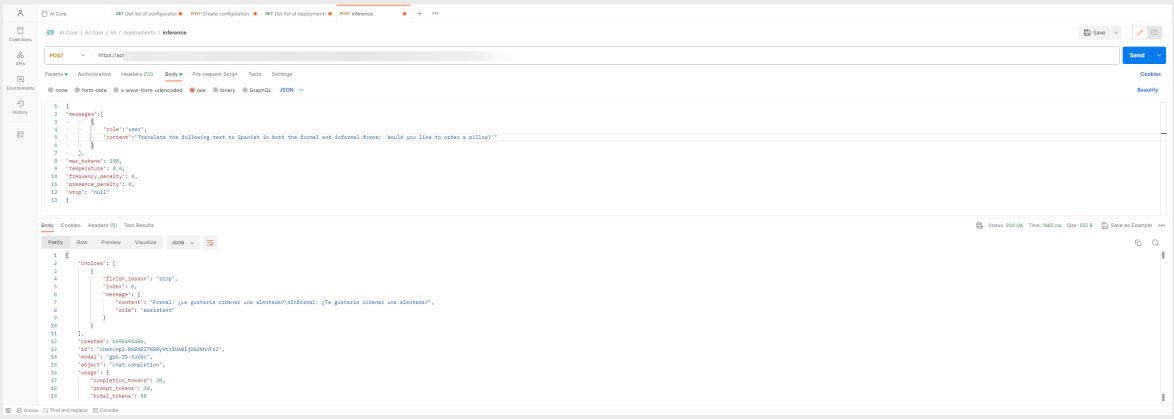


Example

This example translates both the language and register of the text.

```
{
"messages":[
        {
            "role":"user",
            "content": "Translate the following text to Spanish in both the
formal and informal forms: 'Would you like to order a pillow?'"
        }
```

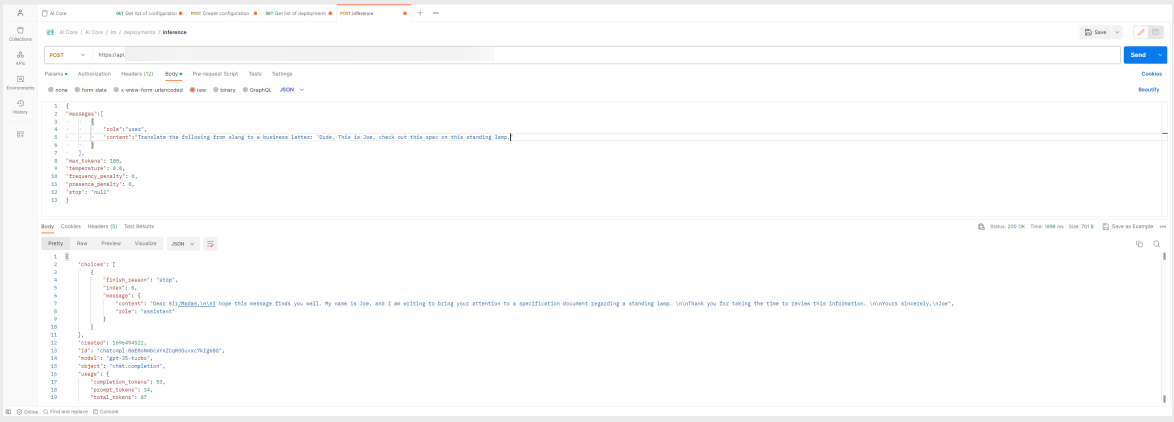
```
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```



Example

This example transforms the text to a formal register.

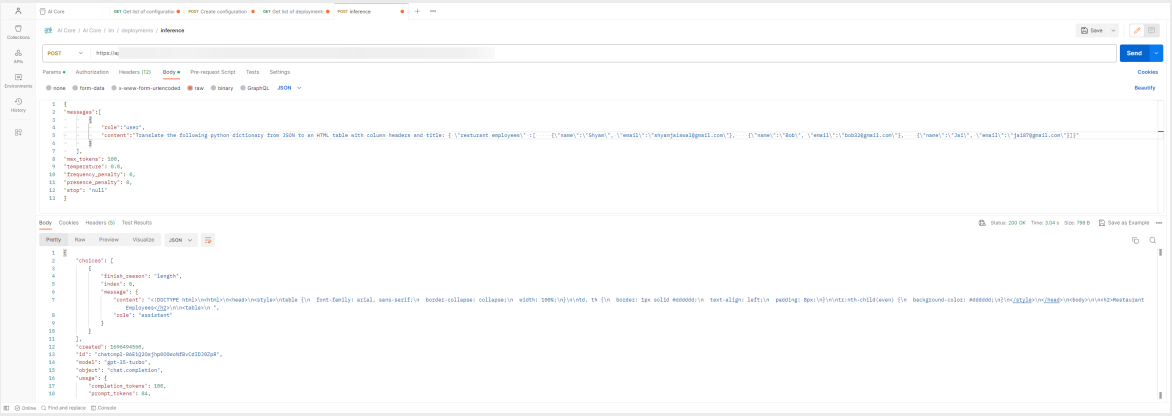
```
{
"messages":[
        {
            "role":"user",
            "content": "Translate the following from slang to a business
letter: 'Dude, This is Joe, check out this spec on this standing lamp.'"
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```



Example

This example translates between output formats. The prompt describes both the input and output format.

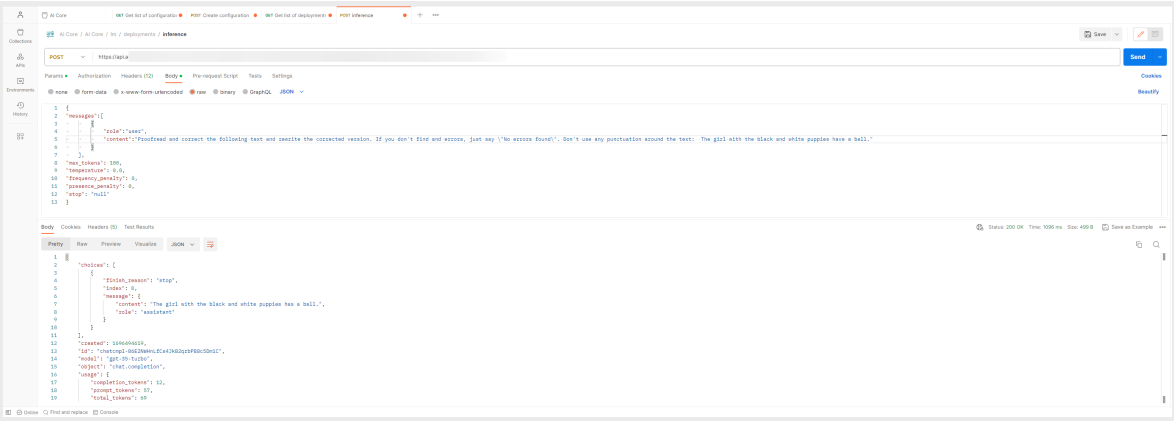
```
{
"messages":[
        {
            "role":"user",
            "content": "Translate the following python dictionary from JSON
to an HTML table with column headers and title: { \"resturant employees\" :
[     {\"name\":\"Shyam\", \"email\":\"shyamjaiswal@gmail.com\"},
{\"name\":\"Bob\", \"email\":\"bob32@gmail.com\"},    {\"name\":\"Jai\",
\"email\":\"jai87@gmail.com\"}]}"
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```



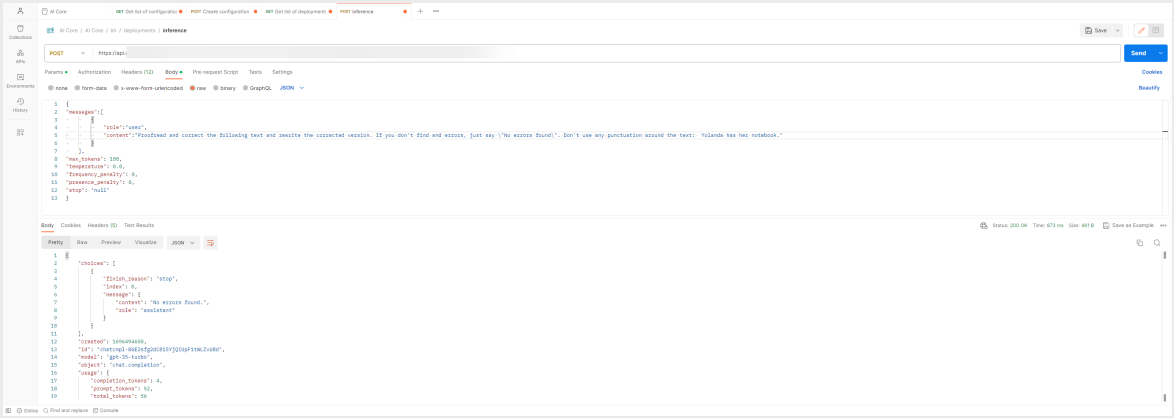
Example

In these examples, a text is proofread. The text can be proofread and corrected, or simply proofread.

```
{
"messages":[
        {
            "role":"user",
            "content": "Proofread and correct the following text and rewrite
the corrected version. If you don't find and errors, just say \"No errors
found\". Don't use any punctuation around the text:  The girl with the black
and white puppies have a ball."
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```

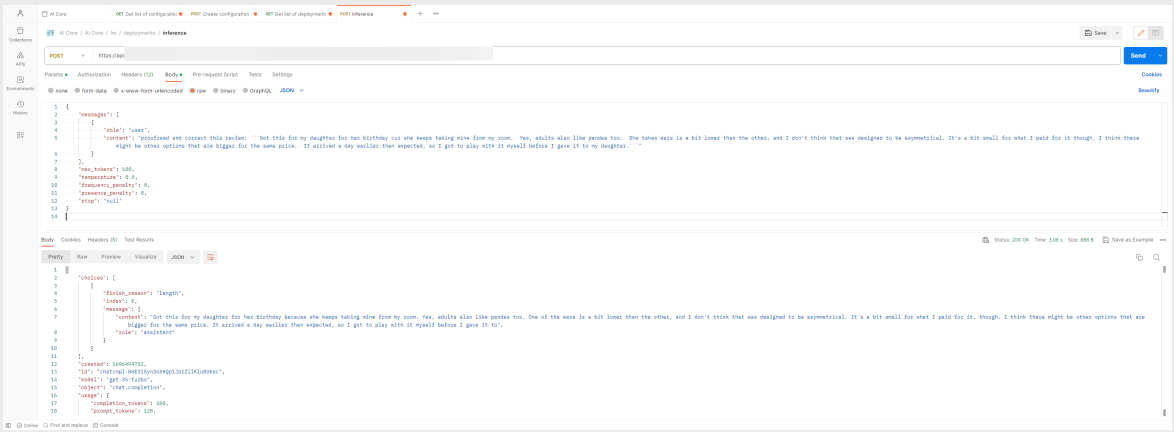


```
{
"messages":[
        {
            "role":"user",
            "content": "Proofread and correct the following text and rewrite
the corrected version. If you don't find and errors, just say \"No errors
found\". Don't use any punctuation around the text:  Yolanda has her
notebook."
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```



```
{
"messages":[
        {
            "role":"user",
            "content": "proofread and correct this review: ```Got this for
my daughter for her birthday cuz she keeps taking mine from my room.  Yes,
adults also like pandas too.  She takes ears is a bit lower than the other,
and I don't think that was designed to be asymmetrical. It's a bit small for
what I paid for it though. I think there might be other options that are
bigger for the same price.  It arrived a day earlier than expected, so I got
to play with it myself before I gave it to my daughter.```"
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
```

```
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```



Expansions

Expansions generate text based on a prompt.

Procedure

Send a POST request to the endpoint `{{deploymentUrl}}/chat/completions?api-version={{api-version}}`.

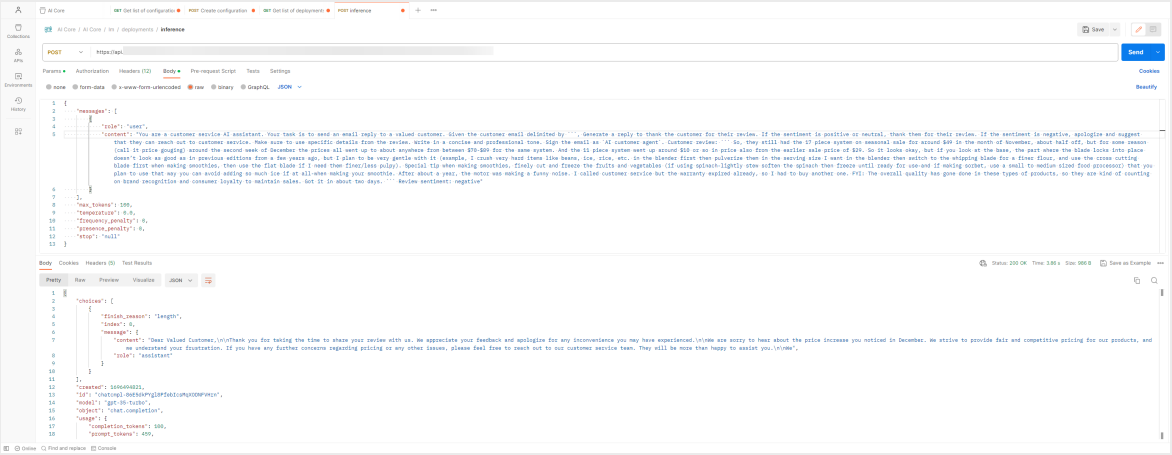
Include your query in the body.

Example

This example generates an automated reply to a customer email.

```
{
"messages":[
        {
            "role":"user",
            "content": "You are a customer service AI assistant. Your task
is to send an email reply to a valued customer. Given the customer email
delimited by ```, Generate a reply to thank the customer for their review.
If the sentiment is positive or neutral, thank them for their review. If
the sentiment is negative, apologize and suggest that they can reach out to
customer service. Make sure to use specific details from the review. Write
in a concise and professional tone. Sign the email as `AI customer agent`.
Customer review: ```So, they still had the 17 piece system on seasonal sale
for around $49 in the month of November, about half off, but for some reason
(call it price gouging) around the second week of December the prices all
went up to about anywhere from between $70-$89 for the same system. And the
11 piece system went up around $10 or so in price also from the earlier sale
price of $29. So it looks okay, but if you look at the base, the part where
```

```
the blade locks into place doesn’t look as good as in previous editions from
a few years ago, but I plan to be very gentle with it (example, I crush very
hard items like beans, ice, rice, etc. in the blender first then pulverize
them in the serving size I want in the blender then switch to the whipping
blade for a finer flour, and use the cross cutting blade first when making
smoothies, then use the flat blade if I need them finer/less pulpy). Special
tip when making smoothies, finely cut and freeze the fruits and vegetables
(if using spinach-lightly stew soften the spinach then freeze until ready
for use-and if making sorbet, use a small to medium sized food processor)
that you plan to use that way you can avoid adding so much ice if at all-
when making your smoothie. After about a year, the motor was making a funny
noise. I called customer service but the warranty expired already, so I had
to buy another one. FYI: The overall quality has gone done in these types
of products, so they are kind of counting on brand recognition and consumer
loyalty to maintain sales. Got it in about two days.``` Review sentiment:
negative"
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```



Chatbot

Context

Chatbots use input and give output in the form of conversations.

Procedure

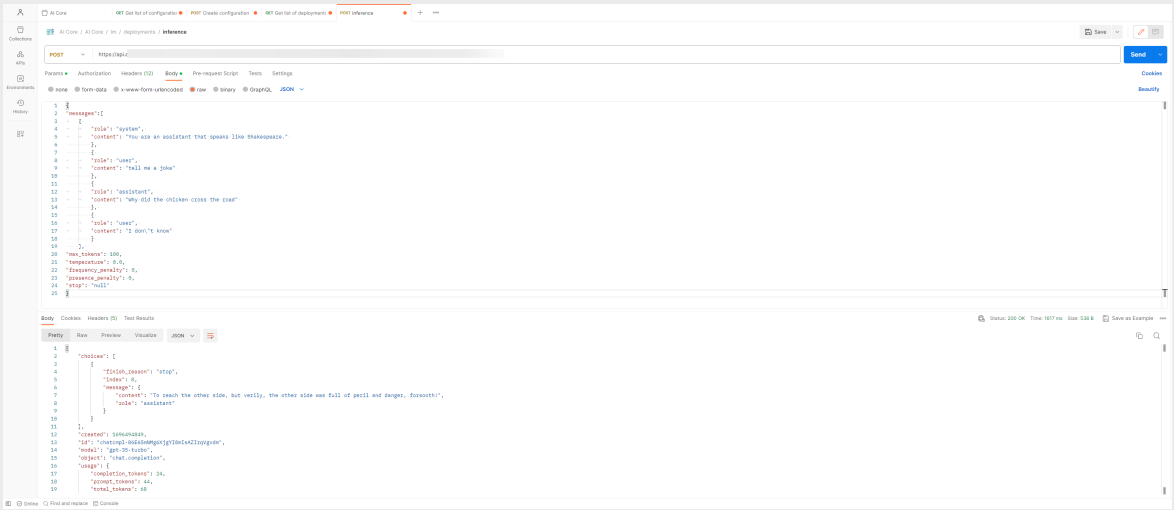
Send a POST request to the endpoint `{{deploymentUrl}}/chat/completions?api-version={{api-version}}`.

Include your query in the body. To provide more context or set a precedent, include examples of the desired outputs in your prompt.

Example

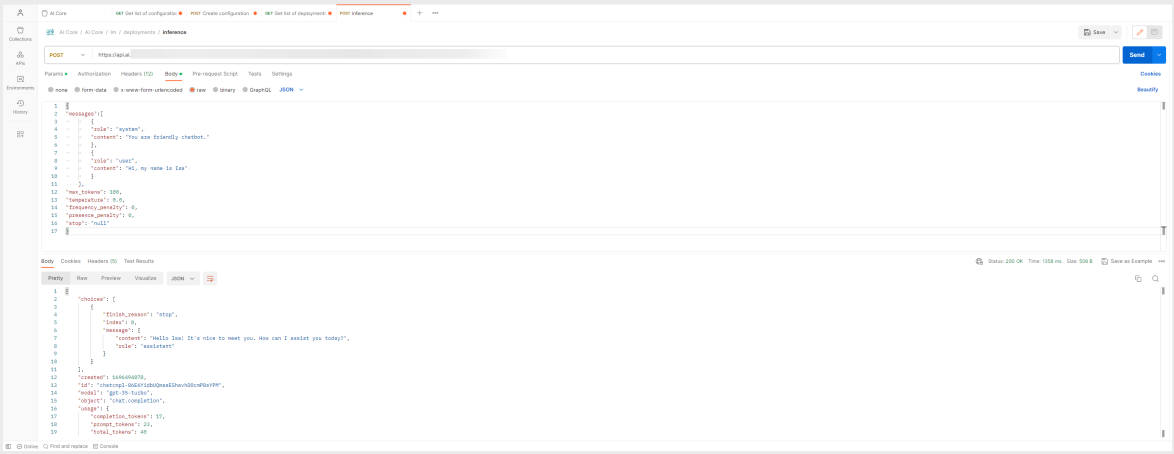
The following examples include few shot prompts for a chatbot.

```
{
"messages":[
    {
        "role": "system",
        "content": "You are an assistant that speaks like Shakespeare."
        },
        {
        "role": "user",
        "content": "tell me a joke"
        },
        {
        "role": "assistant",
        "content": "Why did the chicken cross the road"
        },
        {
        "role": "user",
        "content": "I don\"t know"
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```

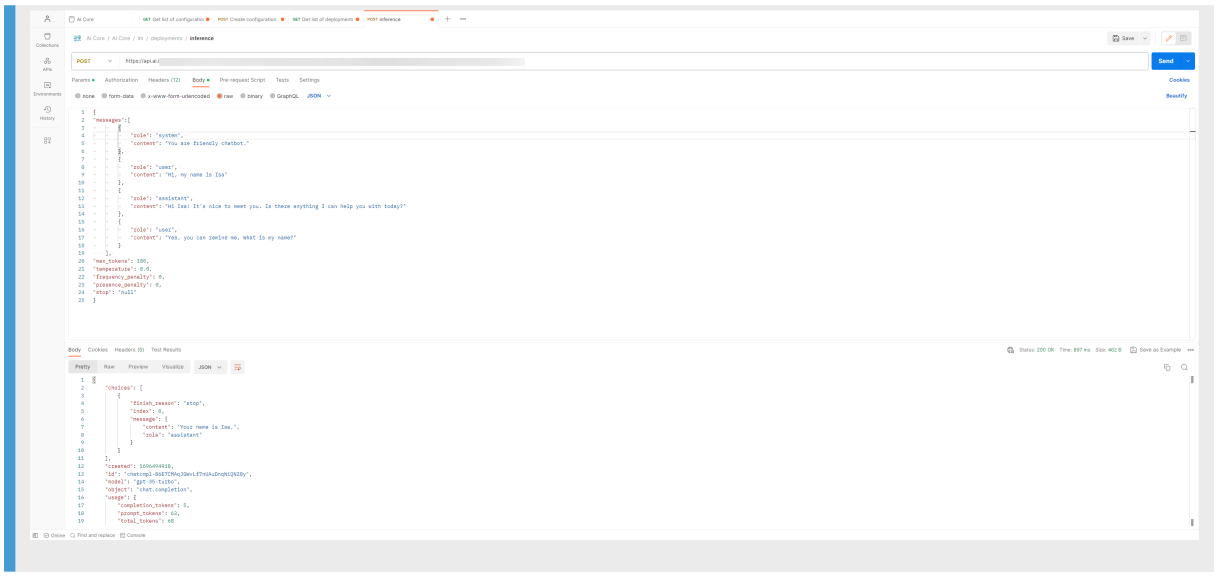


```
{
"messages":[
        {
        "role": "system",
        "content": "You are friendly chatbot."
        },
        {
        "role": "user",
        "content": "Hi, my name is Isa"
        }
    ],
"max_tokens": 100,
```

```
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```



```
{
"messages":[
        {
            "role": "system",
            "content": "You are friendly chatbot."
        },
        {
            "role": "user",
            "content": "Hi, my name is Isa"
        },
        {
            "role": "assistant",
            "content": "Hi Isa! It's nice to meet you. Is there anything I
can help you with today?"
        },
        {
            "role": "user",
            "content": "Yes, you can remind me, What is my name?"
        }
    ],
"max_tokens": 100,
"temperature": 0.0,
"frequency_penalty": 0,
"presence_penalty": 0,
"stop": "null"
}
```



6 Consume Large Language Models Using SAP AI Launchpad

6.1 Prompt Editor

6.1.1 Prompt Experimentation

Prerequisites

- You have at least one deployment for a generative AI model is running. For more information, see Create a Deployment for a Generative AI Model in SAP AI Launchpad [page 20].
- You've selected the AI API connection and resource group that you used in the activation steps.
- You have the `genai_manager`, `prompt_manager`, `genai_experimenter` or `prompt_experimenter` role, or you are assigned a role collection that contains one of these roles. For more information, see Roles and Authorizations.
- Users with only the `genai_experimenter` or `prompt_experimenter` roles are not able to save prompts.

Context

> **i** Note
>
> In addition to the generally available models, there are experimental and preview models maintained by IES. Experimental and preview models have their own data guidelines which differ from those for generally available models from SAP AI Core.
>
> The guidelines for experiemental and preview models are:
>
> - You can send **public data** to all models.
> - You can send **internal data** to all models, except to those in 'preview'.
> - You cannot save any prompt where **confidential** data is sent to the model.
> - You should never send **personal data** to any model. This includes but is not limited to, SAP customer data, personal data of SAP customers and personal data of SAP employees.

> **⚠** Caution
>
> SAP does not take any responsibility for quality of the content in the input to or output of the underlying generative AI models, including but not limited to, bias, hallucinations, or inaccuracies. The user is responsible for verifying the content.

Procedure

1. Select the connection to your SAP AI Core runtime in the *Workspaces* app and choose the resource group that was used for your Generative AI Hub deployment.
2. In the side navigation, expand the *Generative AI Hub* and choose *Prompt Editor*.
3. Input your prompt:
   a. Enter your input data in the *Message* box.

      > **!** Restriction
      >
      > Do not submit sensitive information in prompts when using Generative AI Hub.

   b. **Optional:** Enter a name for your prompt.

      Not available to the `genai_experimenter` or `prompt_experimenter` roles.

   c. **Optional:** Choose a model.

      If you do not choose a model, the default model will be used.

   d. **Optional:** Enter a collection name.

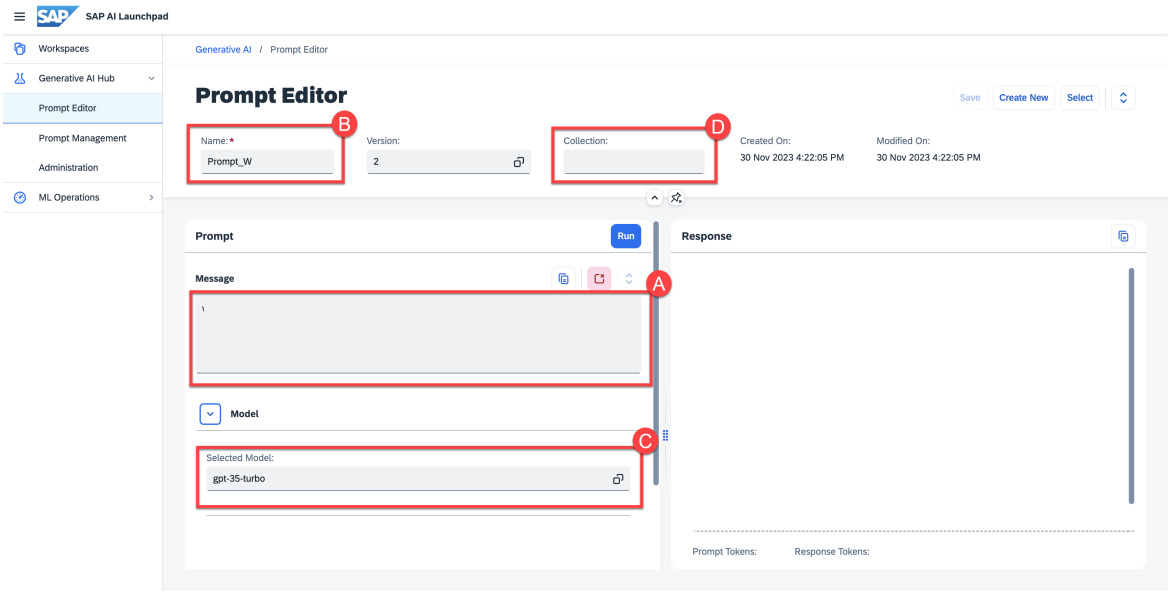
      Not available to the `genai_experimenter` or `prompt_experimenter` roles.

   e. **Optional:** Adjust the parameters, such as:

      - `Max Tokens:` An integer that defines the maximum number of tokens allowed for the generated answer. Max 2048 (4096 for some models), default 16.
      - `Temperature:` Higher values make the output more random; lower values make it more focused and deterministic. Min 0, max 2 (1 for non OpenAI models), step value .01, default 1.
      - `Frequency Penalty:` Tokens that are already used in the text are penalized, higher values make the model output less likely to repeat something already written. Min -2, max 2, step value .01, default 0. Not available for IES models.
      - `Presence Penalty:` Tokens that are already used in the text are penalized, higher values make the model output more likely to include new topics. Min -2, max 2, step value .01, default 0. Not available for IES models.
        For IES models, parameters have been normalised.

   f. **Optional:** Add meaningful tags and notes to the metadata.

      Not available to the `genai_experimenter` or `prompt_experimenter` roles.



4. Choose *Run*

Results

The response to your prompt will be generated.

Next Steps

- You can run your prompt again, make changes to the prompt, model, and parameters to change the outcome.

- You can save your prompt. For more information, see Save a Prompt [page 55].
  Not available to the `genai_experimenter` or `prompt_experimenter` roles.
- You can copy your prompt or response using the copy button.
- You can expand the *Message* field using the expand button.

Sample Prompts

Question Answering

You can ask the LLM a question and receive a response written in natural language.

> Example
>
> Prompt:
>
> ```
> What is python in the context of programing?
> ```

Summarizing

You can provide the LLM with a text and ask for a summary of it.

> Example
>
> Prompt:
>
> ```
> Your task is to generate a short summary of a product review from an ecommerce
> site. Summarize the review below, delimited by triple backticks, in at most 30
> words. Review: Got this panda plush toy for my daughter's birthday, who loves it
> and takes it everywhere. It's soft and super cute, and its face has a friendly
> look. It's a bit small for what I paid though. I think there might be other
> options that are bigger for the same price. It arrived a day earlier than
> expected, so I got to play with it myself before I gave it to her.
> ```

> Example
>
> This example specifies topic focus points.
>
> Prompt:
>
> ```
> Your task is to generate a short summary of a product review from an ecommerce
> site to give feedback to the pricing department, responsible for determining the
> price of the product. Summarize the review below, delimited by triple backticks,
> in at most 30 words, and focusing on any aspects that are relevant to the
> price and perceived value. Review: ```Got this panda plush toy for my daughter's
> birthday, who loves it and takes it everywhere. It's soft and super cute, and
> its face has a friendly look. It's a bit small for what I paid though. I think
> ```

```
there might be other options that are bigger for the same price. It arrived a day earlier than expected, so I got to play with it myself before I gave it to her.
```

Example

This example uses “extract” instead of “summarize”.

Prompt:

```
Your task is to extract relevant information from a product review from an ecommerce site to give feedback to the Shipping department. From the review below, delimited by triple quotes extract the information relevant to shipping and delivery. Limit to 30 words. Review: ```Got this panda plush toy for my daughter's birthday,who loves it and takes it everywhere. It's soft and super cute, and its face has a friendly look. It's a bit small for what I paid though. I think there might be other options that are bigger for the same price. It arrived a day earlier than expected, so I got to play with it myself before I gave it to her.```
```

Inferencing

Inferencing uses the information in a given text to draw a conclusion.

Example

This example performs a sentiment analysis on a product review.

Prompt:

```
What is the sentiment of the following product review, which is delimited with triple backticks? Review text: ```Needed a nice lamp for my bedroom, and this one had additional storage and not too high of a price point. Got it fast. The string to our lamp broke during the transit and the company happily sent over a new one. Came within a few days as well. It was easy to put together. I had a missing part, so I contacted their support, and they very quickly got me the missing piece! Lumina seems to me to be a great company that cares about their customers and products!!```
```

Example

This example generates the sentiment as a one word response.

Prompt:

```
What is the sentiment of the following product review, which is delimited with triple backticks? Give your answer as a single word, either 'positive' or 'negative' Review text: ```Needed a nice lamp for my bedroom, and this one had additional storage and not too high of a price point. Got it fast. The string to our lamp broke during the transit and the company happily sent over a new one. Came within a few days as well. It was easy to put together. I had a missing part, so I contacted their support and they very quickly got me the missing
```

```
piece! Lumina seems to me to be a great company that cares about their customers
and products!!```
```

Example

This example analyzes the emotions expressed in the review.

Prompt:

```
Identify a list of emotions that the writer of the following review is
expressing. Include no more than five items in the list. Format your answer as
a list of lower-case words separated by commas. Review text: ```Needed a nice
lamp for my bedroom, and this one had additional storage and not too high of a
price point. Got it fast. The string to our lamp broke during the transit and the
company happily sent over a new one. Came within a few days as well. It was easy
to put together. I had a missing part, so I contacted their support and they very
quickly got me the missing piece! Lumina seems to me to be a great company that
cares about their customers and products!!
```

Example

This example detects whether anger is present.

Prompt:

```
Is the writer of the following review expressing anger? The review is delimited
with triple backticks. Give your answer as either yes or no. Review text:
```Needed a nice lamp for my bedroom, and this one had additional storage and
not too high of a price point. Got it fast. The string to our lamp broke during
the transit and the company happily sent over a new one. Came within a few days
as well. It was easy to put together. I had a missing part, so I contacted their
support and they very quickly got me the missing piece! Lumina seems to me to be
a great company that cares about their customers and products!!```
```

Example

This example detects product and company names from the customer review.

Prompt:

```
Identify the following items from the review text: - Item purchased by reviewer
- Company that made the item The review is delimited with triple backticks.
Format your response as a JSON object with 'Item' and 'Brand' as the keys. If
the information isn't present, use 'unknown' as the value. Make your response as
short as possible. Review text: ```Needed a nice lamp for my bedroom, and this
one had additional storage and not too high of a price point. Got it fast. The
string to our lamp broke during the transit and the company happily sent over
a new one. Came within a few days as well. It was easy to put together. I had
a missing part, so I contacted their support and they very quickly got me the
missing piece! Lumina seems to me to be a great company that cares about their
customers and products!!``
```
```

Example

This example performs multiple tasks in a single query.

Prompt:

```
Identify the following items from the review text: - Sentiment (positive or negative) - Is the reviewer expressing anger? (true or false) - Item purchased by reviewer - Company that made the item The review is delimited with triple backticks. Format your response as a JSON object with 'Sentiment', 'Anger', 'Item' and 'Brand' as the keys. If the information isn't present, use 'unknown' as the value. Make your response as short as possible. Review text: ```Needed a nice lamp for my bedroom, and this one had additional storage and not too high of a price point. Got it fast. The string to our lamp broke during the transit and the company happily sent over a new one. Came within a few days as well. It was easy to put together. I had a missing part, so I contacted their support and they very quickly got me the missing piece! Lumina seems to me to be a great company that cares about their customers and products!!```
```

Example

This example identifies the five topics discussed in a story.

Prompt:

```
Determine five topics that are being discussed in the following text, which is delimited by triple backticks. Make each item one or two words long. Format your response as a list of items separated by commas. Text sample: '''In a recent survey conducted by the government, public sector employees were asked to rate their level of satisfaction with the department they work at. The results revealed that NASA was the most popular department with a satisfaction rating of 95%. One NASA employee, John Smith, commented on the findings, stating, 'I'm not surprised that NASA came out on top. It's a great place to work with amazing people and incredible opportunities. I'm proud to be a part of such an innovative organization.' The results were also welcomed by NASA's management team, with Director Tom Johnson stating, 'We are thrilled to hear that our employees are satisfied with their work at NASA. We have a talented and dedicated team who work tirelessly to achieve our goals, and it's fantastic to see that their hard work is paying off.' The survey also revealed that the Social Security Administration had the lowest satisfaction rating, with only 45% of employees indicating they were satisfied with their job. The government has pledged to address the concerns raised by employees in the survey and work towards improving job satisfaction across all departments.
```

Transformations

Transformations transform a given text into another language or register.

Example

This example translates text from English to Spanish.

Prompt:

```
Translate the following English text to Spanish: ```Hi, I would like to order a blender```
```

Example

This example detects the language that the text is written in.

Prompt:

```
Tell me which language this is: ```Combien coûte le lampadaire?```
```

Example

This example translates the given text into multiple languages.

Prompt:

```
Translate the following text to French and Spanish and English pirate: ```I want to order a basketball```
```

Example

These examples translate both the language and register of the text.

Prompt:

```
Translate the following text to Spanish in both the formal and informal forms: 'Would you like to order a pillow?
```

Prompt:

```
Translate the following from slang to a business letter: 'Dude, This is Joe, check out this spec on this standing lamp.
```

Example

This example translates between output formats. The prompt describes both the input and output format.

Prompt:

```
Translate the following python dictionary from JSON to an HTML table with column headers and title:
```

```
{ "resturant employees" :[
{"name":"Shyam", "email":"shyamjaiswal@gmail.com"},
{"name":"Bob", "email":"bob32@gmail.com"},
{"name":"Jai", "email":"jai87@gmail.com"}
]}
```

Example

In these examples, a text is proofread. The text can be proofread and corrected, or simply proofread.

Prompt:

```
Proofread and correct the following text and rewrite the corrected version.
If you don't find and errors, just say \"No errors found\". Don't use any
punctuation around the text: The girl with the black and white puppies have a
ball.
```

Prompt:

```
Proofread and correct the following text and rewrite the corrected version. If you
don't find and errors, just say \"No errors found\". Don't use any punctuation
around the text: Yolanda has her notebook.
```

Prompt:

```
Proofread and correct this review: ```Got this for my daughter for her birthday cuz
she keeps taking mine from my room. Yes, adults also like pandas too. She takes
ears is a bit lower than the other, and I don't think that was designed to be
asymmetrical. It's a bit small for what I paid for it though. I think there might
be other options that are bigger for the same price. It arrived a day earlier than
expected, so I got to play with it myself before I gave it to my daughter.```
```

Expansions

Expansions generate text based on a prompt.

Example

This example generates an automated reply to a customer email.

Prompt:

```
You are a customer service AI assistant. Your task is to send an email reply to
a valued customer. Given the customer email delimited by ```, Generate a reply
to thank the customer for their review. If the sentiment is positive or neutral,
thank them for their review. If the sentiment is negative, apologize and suggest
that they can reach out to customer service. Make sure to use specific details
from the review. Write in a concise and professional tone. Sign the email as
`AI customer agent`. Customer review: ```So, they still had the 17 piece system
on seasonal sale for around $49 in the month of November, about half off, but
for some reason (call it price gouging) around the second week of December the
prices all went up to about anywhere from between $70-$89 for the same system.
And the 11 piece system went up around $10 or so in price also from the earlier
sale price of $29. So it looks okay, but if you look at the base, the part where
the blade locks into place doesn’t look as good as in previous editions from a
few years ago, but I plan to be very gentle with it (example, I crush very hard
items like beans, ice, rice, etc. in the blender first then pulverize them in the
serving size I want in the blender then switch to the whipping blade for a finer
flour, and use the cross cutting blade first when making smoothies, then use the
flat blade if I need them finer/less pulpy). Special tip when making smoothies,
```

```
finely cut and freeze the fruits and vegetables (if using spinach-lightly stew
soften the spinach then freeze until ready for use-and if making sorbet, use
a small to medium sized food processor) that you plan to use that way you can
avoid adding so much ice if at all-when making your smoothie. After about a
year, the motor was making a funny noise. I called customer service but the
warranty expired already, so I had to buy another one. FYI: The overall quality
has gone done in these types of products, so they are kind of counting on brand
recognition and consumer loyalty to maintain sales. Got it in about two days.```
Review sentiment: negative.
```

6.1.2 Save a Prompt

Prerequisites

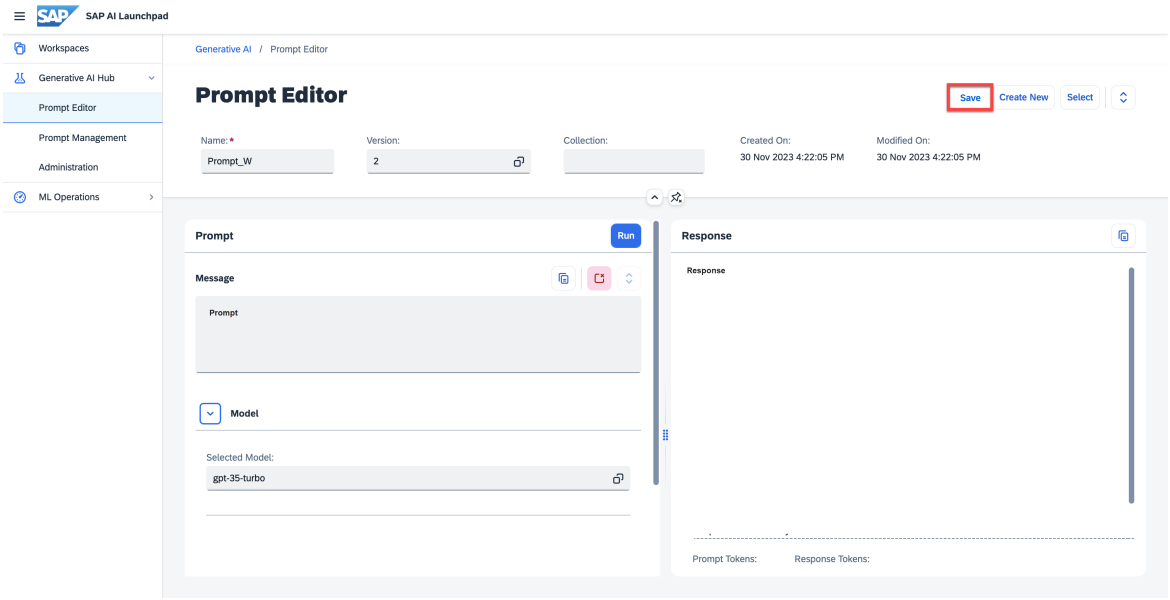
- You have run a prompt succesfully.
- You have either the `genai_manager` or `prompt_manager` role, or you are assigned a role collection that contains one of these roles. For more information, see Roles and Authorizations.
- Users with only the `genai_experimenter` or `prompt_experimenter` roles are not able to save prompts.

Context

- Prompts are saved in one region only, and can only be retrieved or deleted by an AI launchpad instance in that region.
- There is a storage limit applied at tenant level. If you reach this limit, you will receive an error message. Delete saved prompts to make space, or contact your administrator. Your prompt will not be saved and you will have to run it again to save it. You can use the copy function to paste it elsewhere for your reference.

Procedure

1. **Optional:** you can
   - Give your prompt a descriptive *name*.
   - Assign your prompt to a *Collection*.
   - Assign meaningful *Tags* and *Notes* as *Metadata*.
2. Choose *Save*.



6.2 Prompt Management

6.2.1 View a Saved Prompt

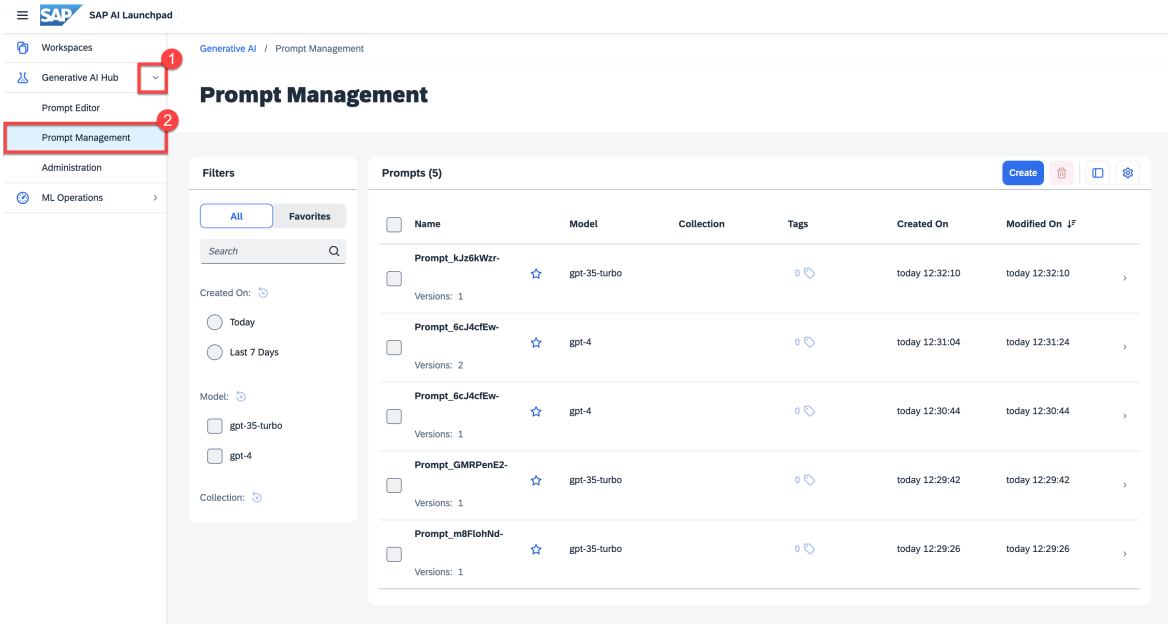
Prerequisites

- You have either the `genai_manager` or `prompt_manager` role, or you are assigned a role collection that contains one of these roles. For more information, see Roles and Authorizations.

> **i Note**
>
> Prompts are saved in one region only, and can only be retrieved or deleted by an AI launchpad instance in that region.

Procedure

1. Select the connection to your SAP AI Core runtime in the *Workspaces* app and choose the resource group that was used for your Generative AI Hub deployment.
2. In the *Workspaces* app, expand the Generative AI Hub and choose *Prompt Management*.



Your prompts will be listed. Use the filters to navigate to your desired prompt.

3. **Optional:** You can
   - Mark a prompt as favourite using the star button.
   - See the prompt details by selecting the prompt entry.
   - See version details by selecting the version entry from the prompt details.

6.2.2 Edit a Saved Prompt

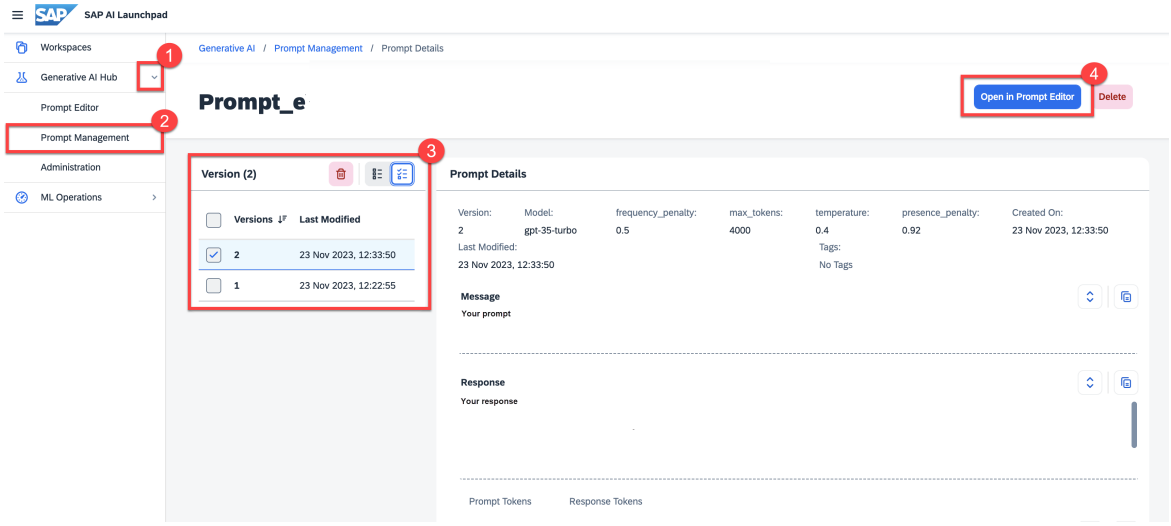
Prerequisites

- You've selected the AI API connection and resource group that you used in the activation steps.
- You have either the `genai_manager` or `prompt_manager` role, or you are assigned a role collection that contains one of these roles. For more information, see Roles and Authorizations.

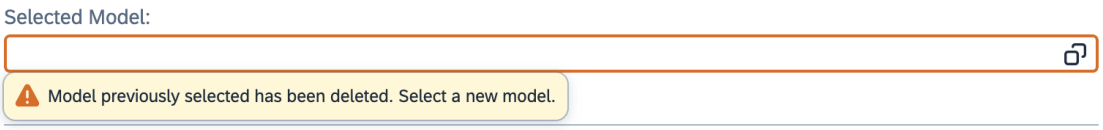
> **i Note**
>
> Prompts are saved in one region only, and can only be retrieved or deleted by an AI launchpad instance in that region.

Procedure

1. Navigate to your desired prompt. For more information, see View a Saved Prompt [page 56].
2. Choose the entry for the version you want to edit.
3. Choose *Open in Editor*. Make your changes.



> **Note**
>
> If the model used for your prompt is no longer available you will be notified when you open the prompt in the editor and you must choose another model.



Next Steps

- You can run your prompt again, making changes to the prompt message, parameters or model selection to change the outcome. For more information, see Prompt Experimentation [page 46].

6.2.3 Delete Prompts

Prerequisites

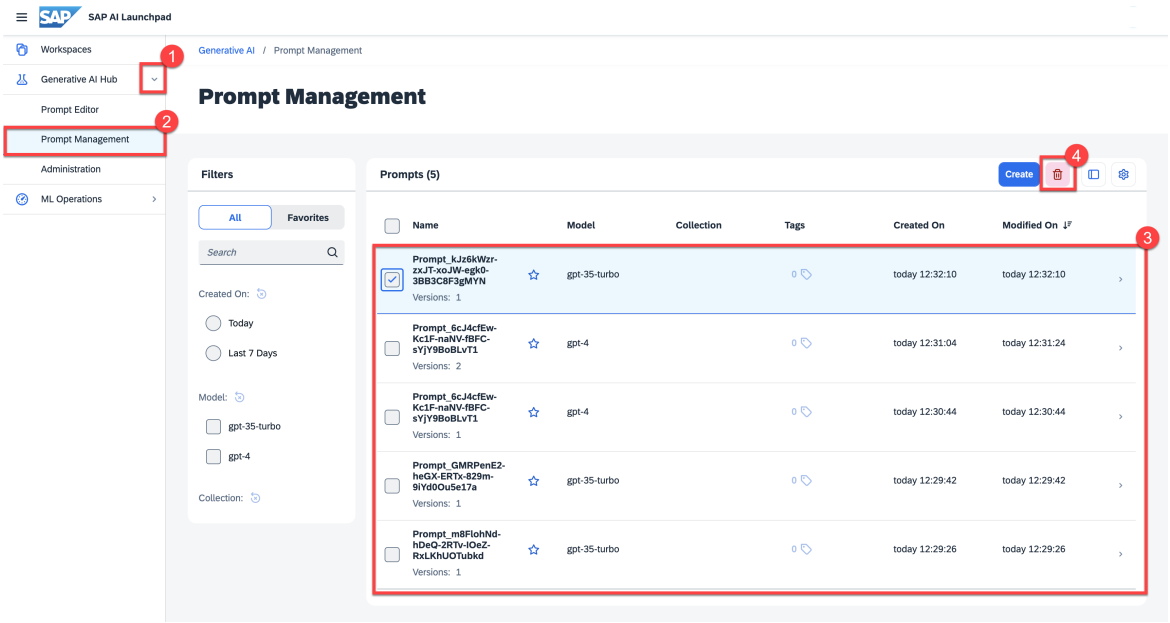
- You have either the `genai_manager` or `prompt_manager` role, or you are assigned a role collection that contains one of these roles. For more information, see Roles and Authorizations.

> **i Note**
>
> Prompts are saved in one region only, and can only be retrieved or deleted by an AI launchpad instance in that region.

Procedure

1. Navigate to *Prompt Management*. For more information, see View a Saved Prompt [page 56].
2. Choose the prompt yout want to delete using the check boxes. You can choose multiple prompts by choosing muliple check boxes.
3. Choose the *Delete* icon. Alternatively, you can choose *Delete* from the *Prompt Details* view.

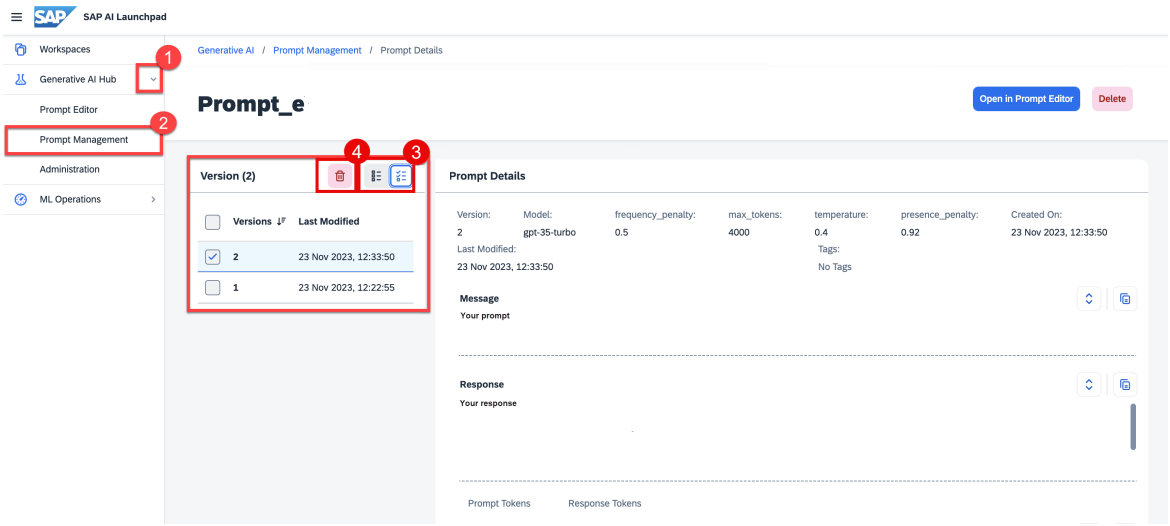
> **i Note**
>
> Deleting a prompt deletes all versions.



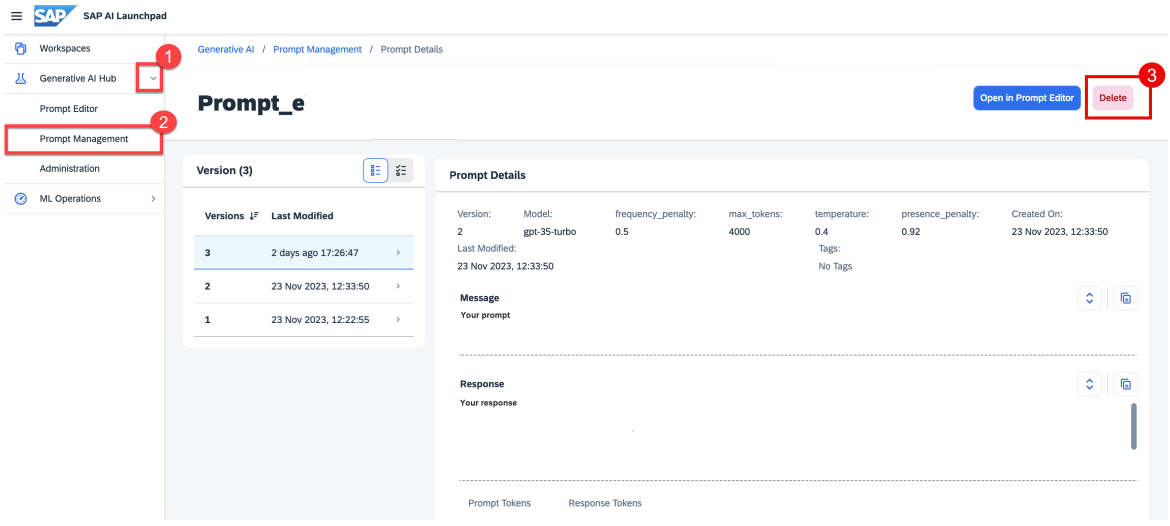
Delete a Prompt Version

Procedure

1. Navigate to your desired prompt. For more information, see View a Saved Prompt [page 56].
2. Switch the *Version* list to the *Select* view.
3. Choose the prompt version you want to delete using the check boxes. You can choose multiple prompt versions by choosing multiple check boxes.
4. Choose the *Delete* icon.



Alternatively, you can delete all prompt versions by choosing *Delete*.



6.3 Administration

6.3.1 Manual User Offboarding

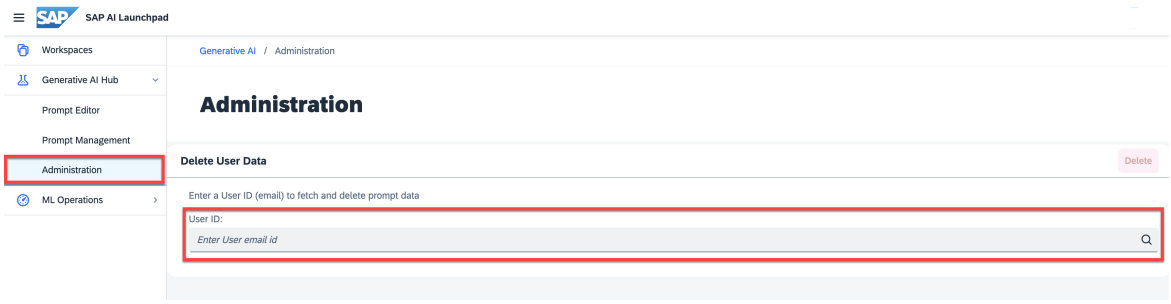
Prerequisites

- You have either the `genai_administrator` or `prompt_administrator` role, or you are assigned a role collection that contains one of these roles. For more information, see Roles and Authorizations.

> **i** Note
>
> - User data is saved in one region only, and can only be retrieved or deleted by an AI launchpad instance in that region.
> - In addition to manual user offboarding, prompt data can be deleted automatically. For more information, see Data Protection and Privacy.

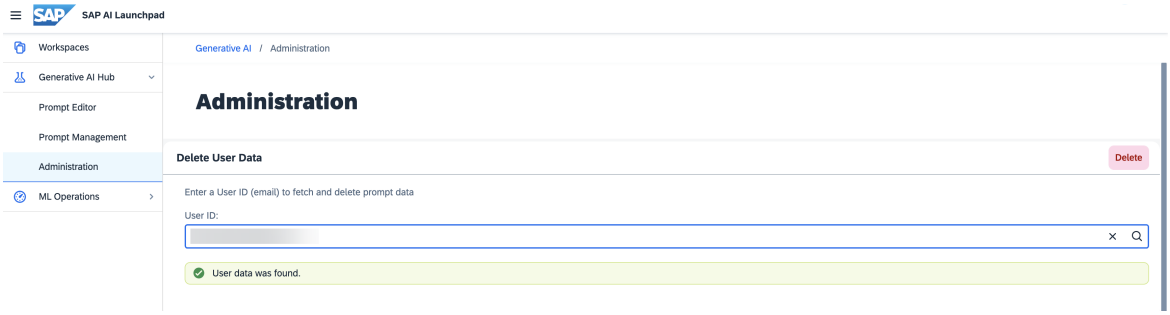
Procedure

1. Select the connection to your SAP AI Core runtime in the *Workspaces* app and choose the resource group that was used for your Generative AI Hub deployment.
2. In the *Workspaces* app, expand the Generative AI Hub and choose *Administration*.
3. Check for a user in the system by entering the user's email. Choose search.

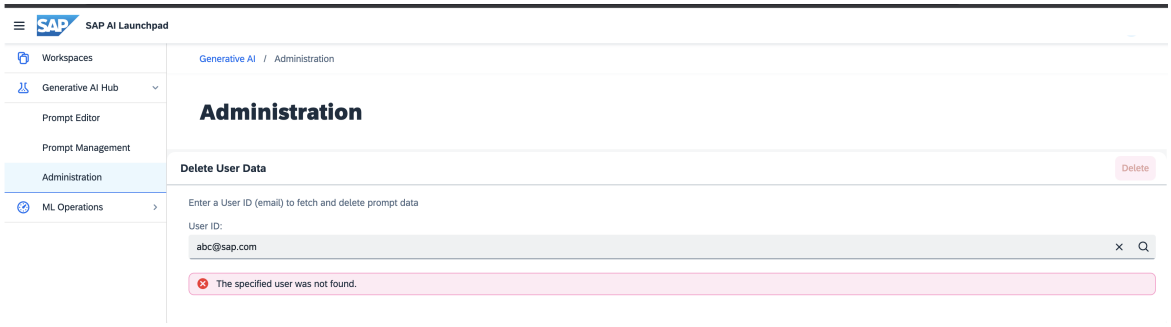


4. If the user is found, you can choose *Delete* to delete their entire data.

> **i** Note
>
> The user data won't be fetched.



If the user is not found, you will be informed.



7 Stopping or Deleting a Deployment

7.1 Stop or Delete a Deployment in SAP AI Core

Stop a Deployment

Procedure

Send a PATCH request to the endpoint `{{apiurl}}/v2/lm/deployments/{{deploymentid}}`. Include the following data in your request:

```
{
"targetStatus": "STOPPED"
}
```

Delete a Deployment

Procedure

Send a DELETE request to the endpoint `{{apiurl}}/v2/lm/deployments/{{deploymentid}}`.

7.2 Stop or Delete a Deployment in SAP AI Launchpad

Stop a Deployment

Procedure

1. Navigate to the deployment's details.
2. Choose *Stop* in the header. A *Warning* dialog box appears.

3. Choose *Stop* to stop running the deployment.

Delete a Deployment

Procedure

1. Navigate to the deployment's details.
2. Choose *Delete* in the header. A *Warning* dialog box appears.
3. Choose *Delete* to confirm the deletion.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

THE BEST RUN SAP