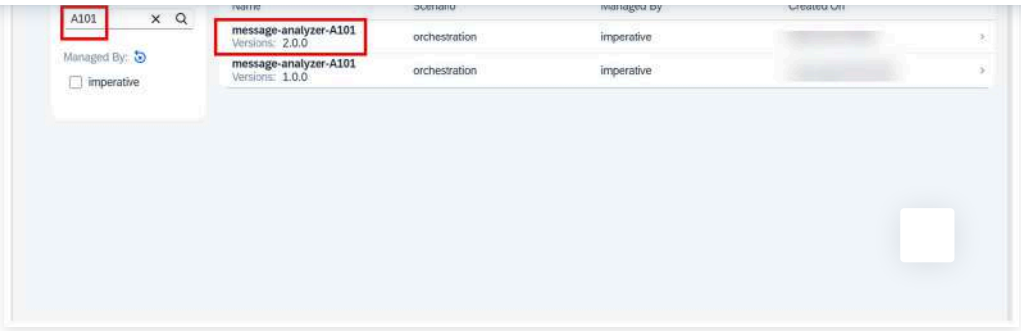


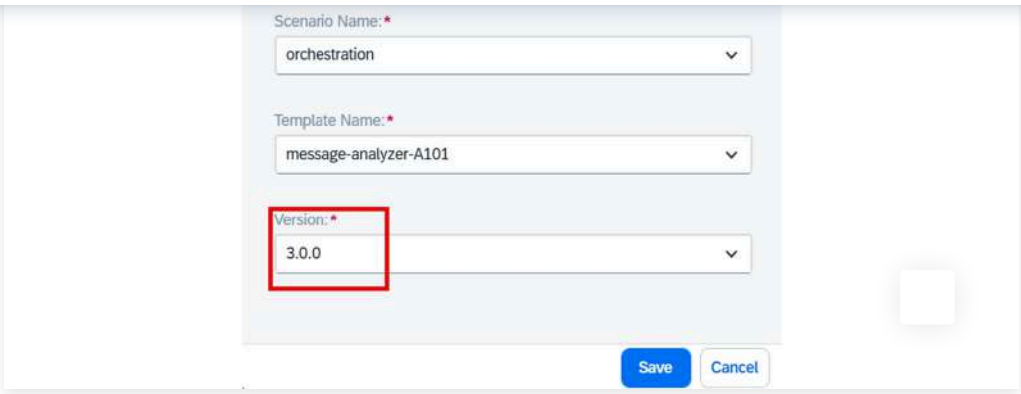
6. Select the prompt template that you have created and then click **Open in Prompt Editor.**
7. Use the following prompt in the user role.

Code Snippet

```
5  If any field's value cannot be determined fr
6  </Instructions>
7
8  <OutputFormat>
9  {
10   "Complaint_ID": "string (e.g., AUTO-GEN-00
11   "Complaint_Type": "enum (Plumbing, HVAC, E
12   "Urgency": "enum (High, Medium, Low)",
13   "Problem_Description": "string (concise su
14   "Affected_Location": "string (e.g., Apartm
15   "Customer_Sentiment": "enum (Very Negative
16   "Suggested_Initial_Action": "string (clear
17 }
18 </OutputFormat>
19
20 <UserQuery>
21 {{?user_email_placeholder}}
22 </UserQuery>
23
```

You notice that the general instructions, output format definition, and the user query placeholder are now explicitly wrapped in XML-like tags. This provides clear visual cues and structural guidance to the LLM.

8. Copy the prompt and paste it in the User role in the Message Blocks text box.
9. Click the **Save Template** button. The Save Template dialog box is displayed.
10. Change the Version to 3.0.0.



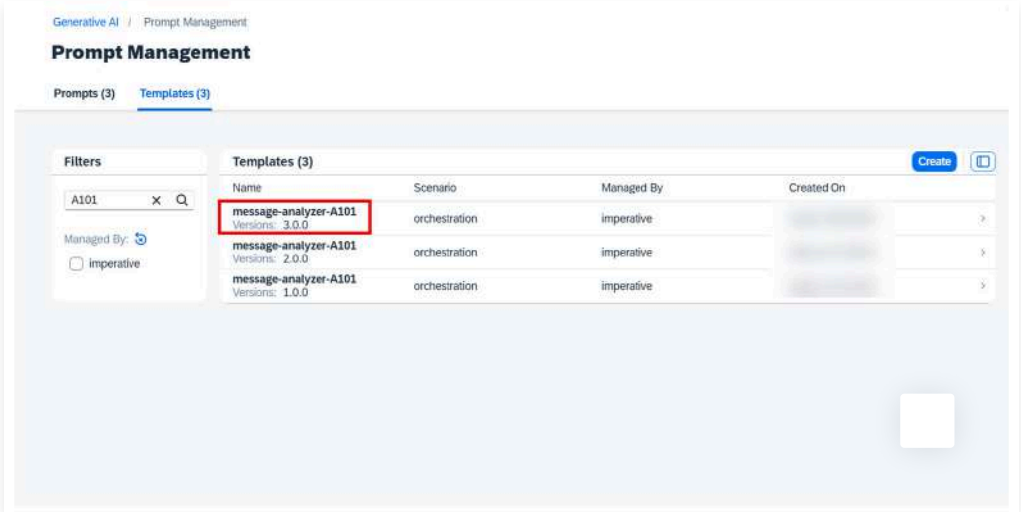
11. Click the **Save** button. The template is saved. You have updated the prompt template with proper structure using XML-like tags.

Task 4: Use your Prompt Template to Address your Business Problem

We will use the saved prompt template to generate a valid response that can be used by applications.

Steps

1. Ensure that you are logged on to generative AI hub.
2. Select **Prompt Management and then Templates.** You can see your template here. You can also search for it, if needed.
3. Select the latest version of the template which is 3.0.0. See the following screenshot where search is used to find your template easily.



Quick links

Download Catalog (CSV, JSON, XLSX, XML)

SAP Learning Hub

SAP Training Shop

SAP Developer Center

SAP Community

Newsletter

Learning Support

Get Support

Share Feedback

Release Notes

About SAP

Company Information

Copyright

Trademark

Worldwide Directory

Careers

News and Press

Site Information

Privacy

Terms of Use

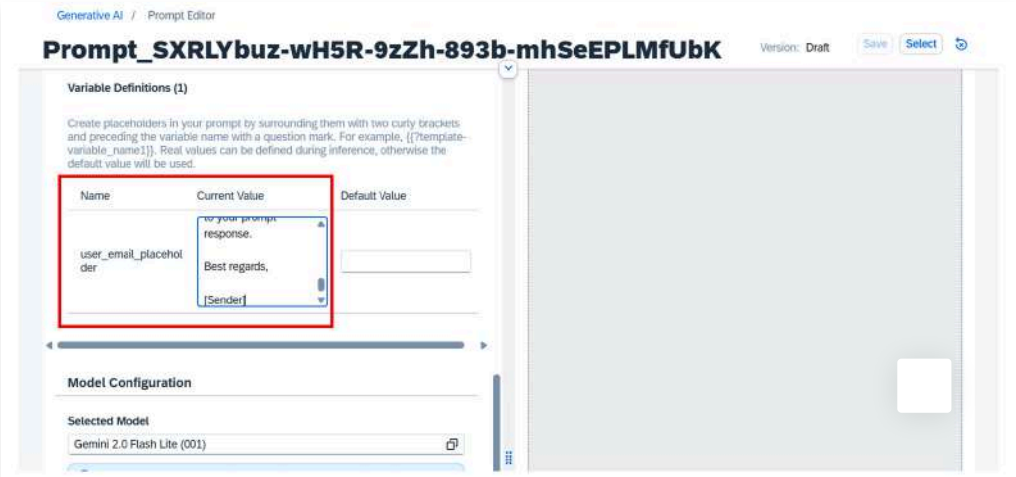
Legal Disclosure

Do Not Share/Sell My Personal Information (US Learners Only)

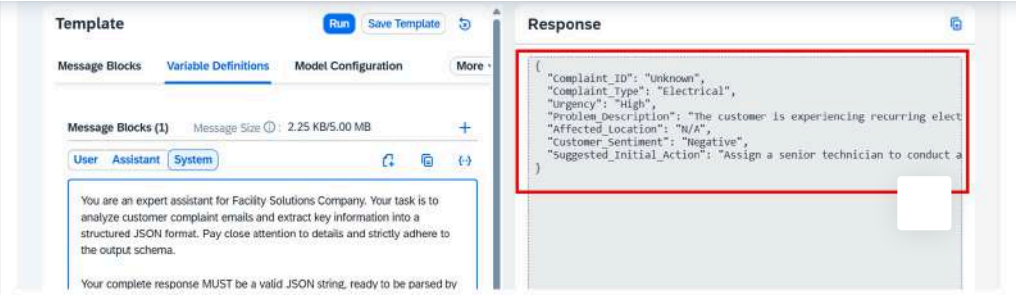
Preferências de Cookies

```
I hope this message finds you well. My name

We have been facing several recurring issues

To give you a clearer picture, we have had t

I am reaching out to request a more permanen

I trust that you understand the urgency of t

Thank you for your attention to this matter.

Best regards,

[Sender]
```

7. Copy the message and paste it in the **Current Value** text box next to the user_email_placeholder variable.



8. Scroll up and **Run** the prompt. A response is generated. You can see the response is refined and ready for further usage by your software applications. **Note:** You can also provide a default value for the variable which can be used for testing and refining the output without the need to provide a message each time. We will use this later.



You have used your prompt template to execute a prompt.

Continue to quiz

Was this lesson helpful? Yes No

Developing Basic Prompts for Common Queries

Objective

After completing this lesson, you will be able to build basic prompts using generative AI hub in SAP AI Launchpad.

Developing Basic Prompts for Common Queries

Once you have accessed the generative AI hub and deployed your orchestration service, you're ready to solve concrete business problems using the generative AI hub. A key skill for this interaction is **developing prompts**, which involves creating clear and effective instructions for an LLM.

This lesson will guide you through an iterative process of developing a basic prompt within SAP AI Launchpad to address a business challenge. We will focus on utilizing the system and user roles to provide clear instructions and context, ultimately producing structured, machine-readable output. By the end of this lesson, you will be able to construct and refine prompts that yield precise, actionable results, ready for integration into your enterprise applications.

Identifying the Business Scenario

Facility Management

Our focus for this learning journey is a Facility Solutions Company, a premier provider of comprehensive facility management, maintenance, and cleaning services for both residential and commercial properties. Their

The Business Problem:

The Facility Solutions company receives thousands of emails daily from customers regarding requests, complaints, and other inquiries. They maintain internal applications to address these customer communications, requiring efficient processing and prioritization to ensure timely and accurate responses.

However, categorizing these emails currently involves a manual process: transferring data from emails to internal applications, then manually categorizing and prioritizing each task. This process is time-consuming, error-prone, and creates bottlenecks in customer service.

The company is now turning to SAP's generative AI hub to automate and streamline this critical process. Throughout this learning journey, we will provide step-wise solutions to this problem.

Developing Prompts in SAP AI Launchpad

Developing a prompt to solve a business problem is an iterative ideation process. You start with a basic idea, test it, observe the LLM's response, and then refine your instructions until the output meets your specific business needs.

The best practice in modern prompt engineering is the use of distinct conversational roles: system and user. These roles provide clarity to the LLM about the nature of each piece of text it receives:

- The **system role** is used to set the overarching rules, persona, and constraints for the LLM. It defines who the LLM is, how it should behave, and what general guidelines it must follow throughout the interaction.
- The **user role** provides the specific query or input for that particular turn. This is where you put the dynamic content that the LLM needs to process.

Let's begin developing a prompt to extract urgency and sentiment from an incoming customer email. You can start with any model like Gemini models or an open source LLM, like Mistral AI, accessible via the generative AI hub. Open-source LLMs offer advantages like cost reduction, transparency, and customization, making them excellent choices for development.

Step 1: Develop a Basic Prompt for Urgency and Sentiment

JSON

```
[
  {
    "role": "system",
    "content": "You are a helpful assistant
  },
  {
      "role": "user"
"content": " Analyze the following message:
---
Subject: Urgent HVAC System Repair Needed

Dear Support Team,

I hope this message finds you well. My name

However, I am currently facing a pressing i

I have attempted to troubleshoot the proble

I kindly request that a repair team be disp

Thank you for your prompt attention to this

Best regards,
```

When you execute this prompt in SAP AI Launchpad, you may receive a lengthy, conversational response. For instance: "The urgency of this message is **high.** The sentiment conveyed is **negative,** as the customer is experiencing significant discomfort due to a malfunctioning HVAC system."

While the LLM correctly identified the urgency and sentiment, this verbose output is not directly useful for an application that needs to automatically process these values. We need to refine the prompt to produce more structured data.

Prompt Structure:

JSON

```
[
  {
    "role": "system",
    "content": "You are an expert customer
  },
    {
    "role": "user",
    "content": "Analyze the following messa
---
Subject: Urgent HVAC System Repair Needed

Dear Support Team,

I hope this message finds you well. My name

However, I am currently facing a pressing i

I have attempted to troubleshoot the proble

I kindly request that a repair team be disp

Thank you for your prompt attention to this

Best regards,
```

Now, the LLM's response will be more constrained, perhaps "Urgency: high, Sentiment: negative." This is better, but still requires parsing from a natural language string.

Step 3: Generate JSON Output for Urgency and Sentiment

Prompt Structure:

JSON

```
[
  {
    "role": "system",
    "content": "You are an expert customer

Your response MUST be a valid JSON string c
  },
  {
    "role": "user",
    "content": "Analyze the following messa
---
Subject: Urgent HVAC System Repair Needed

Dear Support Team,

I hope this message finds you well. My name

However, I am currently facing a pressing i

I have attempted to troubleshoot the proble

I kindly request that a repair team be disp

Thank you for your prompt attention to this
```

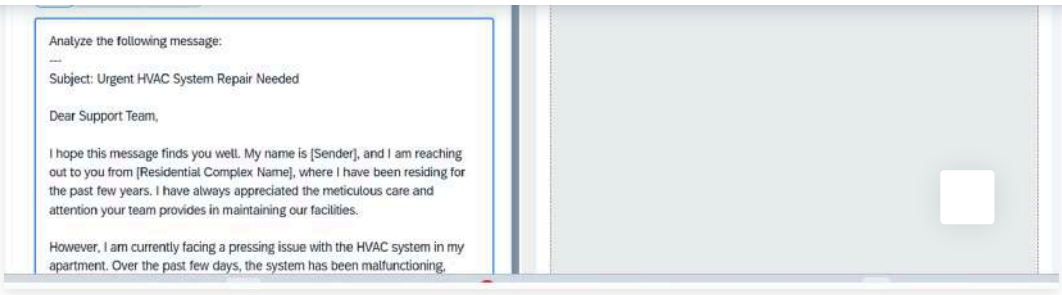
This prompt will yield a JSON output like: {"urgency": "high", "sentiment": "neutral"}. This is a significant improvement!

Step 4: Ensure Correct JSON Formatting

Even with instructions for JSON, LLMs can sometimes include extraneous characters like markdown code blocks (json ...) or extra whitespace, which

JSON

```
[
  {
    "role": "system",
    "content": "You are an expert customer

Your complete response MUST be a valid JSON
  },
  {
    "role": "user",
    "content": "Analyze the following messa
---
Subject: Urgent HVAC System Repair Needed

Dear Support Team,

I hope this message finds you well. My name

However, I am currently facing a pressing i

I have attempted to troubleshoot the proble

I kindly request that a repair team be disp

Thank you for your prompt attention to this
```



As illustrated in the screenshot, this prompt gives clear instructions to provide a clean JSON format without any quotes or extra whitespaces, making it robust for software consumption.

Step 5: Simple Categories Based on Business Functions

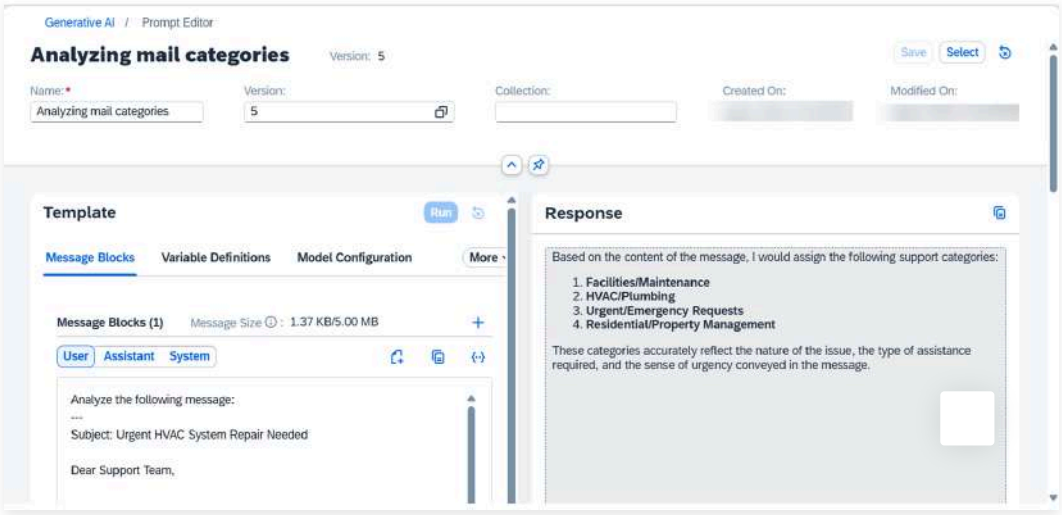
Beyond urgency and sentiment, we also need to categorize messages for business needs. Initially, we'll ask the LLM to simply assign relevant categories without providing a predefined list.

Prompt Structure:

JSON

```
[
  {
    "role": "system",
    "content": "You are an expert customer
  },
  {
    "role": "user",
    "content": "Analyze the following messa
---
Subject: Urgent HVAC System Repair Needed

Dear Support Team,

I hope this message finds you well. My name

However, I am currently facing a pressing i

I have attempted to troubleshoot the proble
```

```
24  Best regards,
```



Now, the LLM will assign categories. However, these categories might be free-form, potentially overlapping with other values like urgency, and not aligned with predefined internal categories.

Step 6: Assigning Values to Categories from a Predefined List

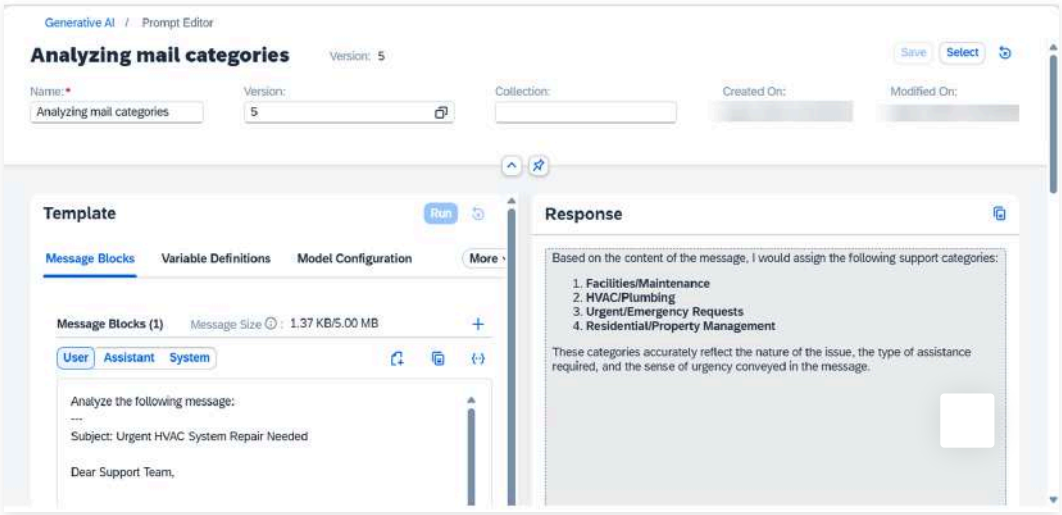
To ensure consistency and alignment with the company's internal processing, we need the LLM to select categories from a specific, predefined list. Aligning the output to business needs is a key step in addressing a business problem effectively.

Prompt Structure:

JSON

```
1  [
2    {
3      "role": "system",
4      "content": "You are an expert customer
5  `facility_management_issues`, `cleaning_ser
6    },
```

```
12
13  Dear Support Team,
14
15  I hope this message finds you well. My name
16
17  However, I am currently facing a pressing i
18
19  I have attempted to troubleshoot the proble
20
21  I kindly request that a repair team be disp
22
23  Thank you for your prompt attention to this
24
```



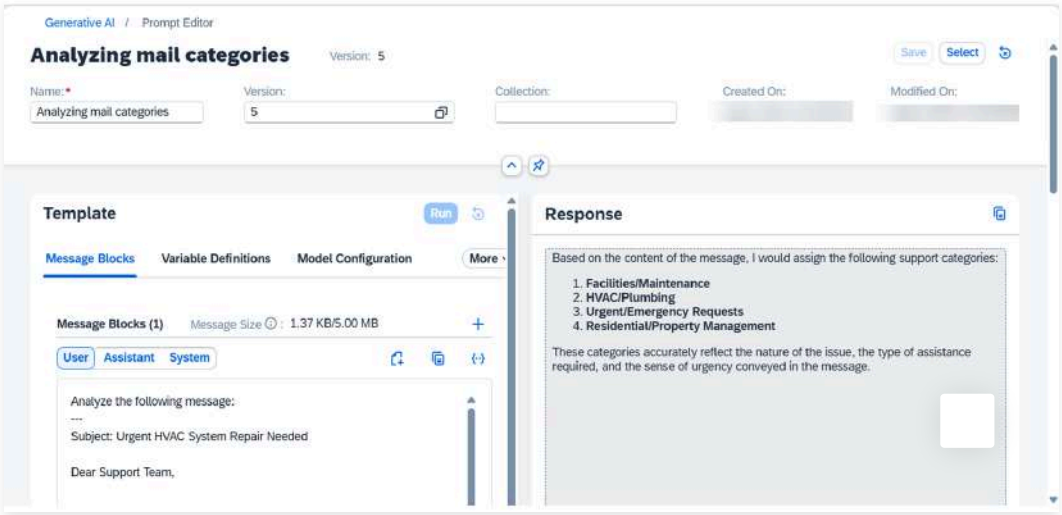
As shown in the screenshot, in the version 6, the LLM will now assign categories strictly from the provided list, which are streamlined for business processing.

Step 7: Generate JSON Output for Categories Values

Similar to Step 3, we need the category values in a JSON output for programmatic processing. We will also reiterate the importance of clean JSON formatting.

Prompt Structure:

```
    "role": "system",
    "content": "You are an expert customer
`facility_management_issues`, `cleaning_ser

Your complete response MUST be a valid JSON
  },
  {
    "role": "user",
    "content": "Analyze the following messa
---
Subject: Urgent HVAC System Repair Needed

Dear Support Team,

I hope this message finds you well. My name

However, I am currently facing a pressing i

I have attempted to troubleshoot the proble

I kindly request that a repair team be disp
```

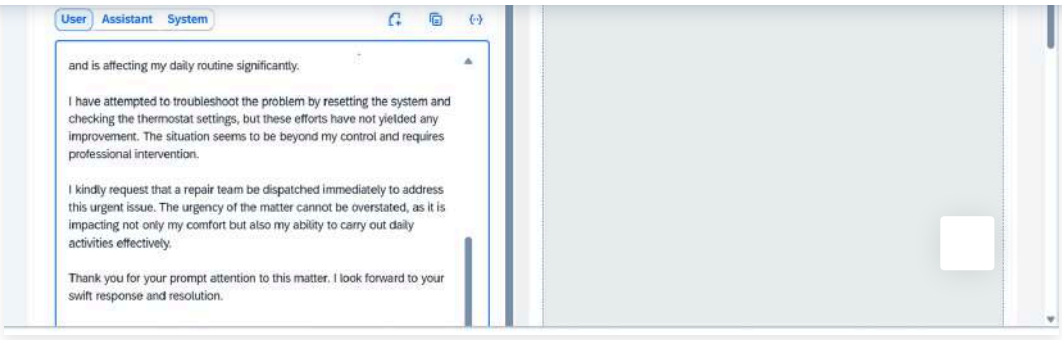


We have taken a step-by-step approach to arrive at proper values for urgency, sentiment, and categories in JSON format. Now, we consolidate all these instructions into a single, robust prompt. This combined prompt will serve as the final version for our automated email categorization system.

Prompt Structure:

JSON

```
[
  {
    "role": "system",
    "content": "You are an expert customer

For 'urgency', classify the message as one
For 'sentiment', classify the message as on
For 'categories', assign a list of the best
`facility_management_issues`, `cleaning_ser

Your complete response MUST be a valid JSON
  },
  {
    "role": "user",
    "content": "Analyze the following custo
---
Subject: Urgent HVAC System Repair Needed

Dear Support Team,

I hope this message finds you well. My name

However, I am currently facing a pressing i
```



As illustrated in the screenshot, this consolidated prompt delivers the output as a clean JSON, for example:

JSON

```
1 {"urgency": "high", "sentiment": "negative", "c
```

This output now consistently assigns urgency, sentiment, and categories to customer messages in a format that is immediately processable by software, fulfilling the business requirement. You have successfully developed a prompt that moves towards a solution to the problem.

Lesson Summary

You have successfully navigated the iterative process of developing basic prompts for common queries within the generative AI hub in SAP AI Launchpad. You've seen how defining clear instructions using the system and user roles, providing specific classification options, and explicitly requesting structured JSON output are crucial for creating machine-

Getting Started with Generative AI Hub

Objective

After completing this lesson, you will be able to assess the generative AI hub and deploy Large Language Models (LLMs) for enterprise use.

Getting Started with the Generative AI Hub: Access & Deployment

SAP's AI Foundation provides the operating system for enterprise-grade AI, with the generative AI hub as its core component for accessing Large Language Models (LLMs). The generative AI hub offers access to a wide range of models and the capability to orchestrate your AI workflows securely and reliably.

This lesson will guide you through the fundamental steps required to gain access to the generative AI hub and deploy LLMs for your specific use cases. You will learn how to set up your environment, manage resource groups, deploy the Orchestration Service (and, optionally, individual LLMs), and how to restrict model usage and strategically manage model upgrades. Mastering these access and deployment procedures is essential for unlocking the practical capabilities of generative AI within your SAP landscape and laying the groundwork for building robust enterprise AI applications.

Gain Access to the Generative AI Hub

To access the generative AI hub, begin by setting up your SAP Business Technology Platform (BTP) environment and provisioning the required AI

entry and management for all services on SAP BTP. It's the administrative umbrella under which your AI capabilities will reside.

To access your SAP BTP Enterprise Account's global account and make the initial configurations refer to Setup your Global Account of your SAP BTP Enterprise Account

2. **Provision SAP AI Core from the SAP BTP Cockpit:** The generative AI hub operates within SAP AI Core, which is a service you provision directly from your SAP BTP cockpit. This provisioning process **generates a service key.** This service key contains the necessary URLs and credentials that authenticate and authorize your access to your dedicated SAP AI Core instance.

   See the following documentation for this process: Initial setup for SAP AI core

   To set up generative AI hub in SAP AI Core to start consuming LLMs, refer to Set up Generative AI Hub in SAP AI Core .

3. **Connect to SAP AI Core Tools:** Once SAP AI Core is provisioned, you need to establish a connection from your preferred tools. Generative AI hub can be primarily accessed through **SAP AI Launchpad,** a user interface for managing and monitoring your AI assets. For developers, programmatic access is also vital, allowing you to connect using tools like Bruno for API testing or Python SDKs for integrating AI capabilities directly into your applications.

   To create a connection between SAP AI Core and the tool of your choice (SAP AI Launchpad, API clients like Postman, SDK for languages like Python) refer to Set Up Tools to Connect With and Operate SAP AI Core .

4. **Securing Access via Roles and Authorizations:** Access to the generative AI hub via **SAP AI Launchpad** is managed through specific roles and authorizations. The generative AI hub is available in SAP AI Core exclusively with the **extended** service plan.

   To utilize the generative AI hub via **SAP AI Launchpad**, users must be assigned one of the following roles or have a role collection that includes one of these roles:

   genai_manager
   prompt_manager
   genai_experimenter

prompt_experimenter roles are not permitted to save prompts in SAP AI Launchpad .

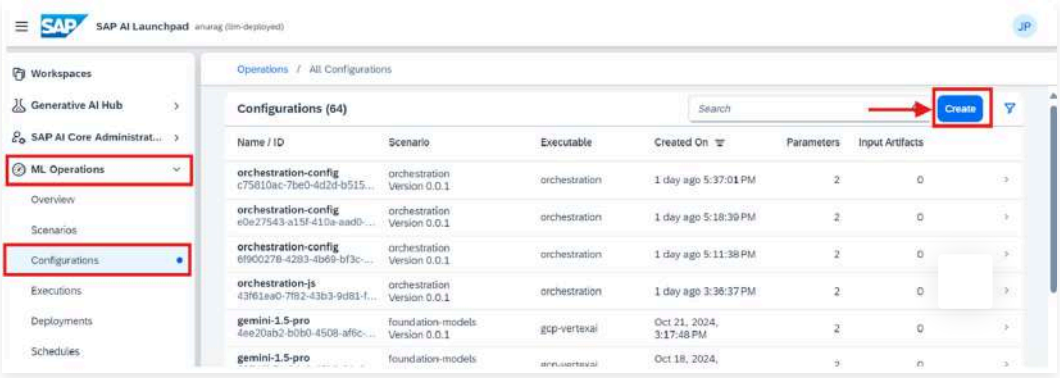
For additional details, refer to Roles and Authorizations .

Orchestration Deployment

Once you have accessed generative AI hub, the next step is to deploy the specific LLMs you intend to use. Deployment in the generative AI hub means instantiating a use-case-specific LLM configuration that your applications can then consume.

You can start with an orchestration deployment. There is no need to deploy individual models and their configuration. You can use a single orchestration deployment to configure and consume models in generative AI hub.

Select the Configuration section for your resource group.



Fill in Deployment Details, under **Configurations,** with the following details:

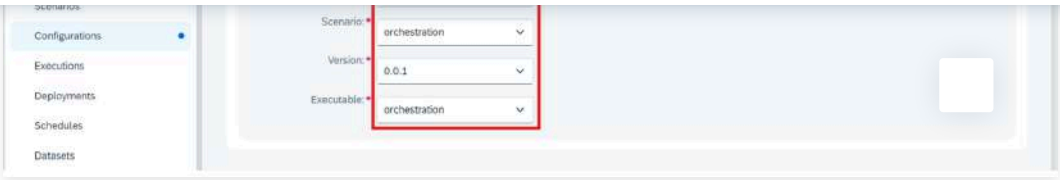
Name: "orchestration"

Executable: "orchestration"

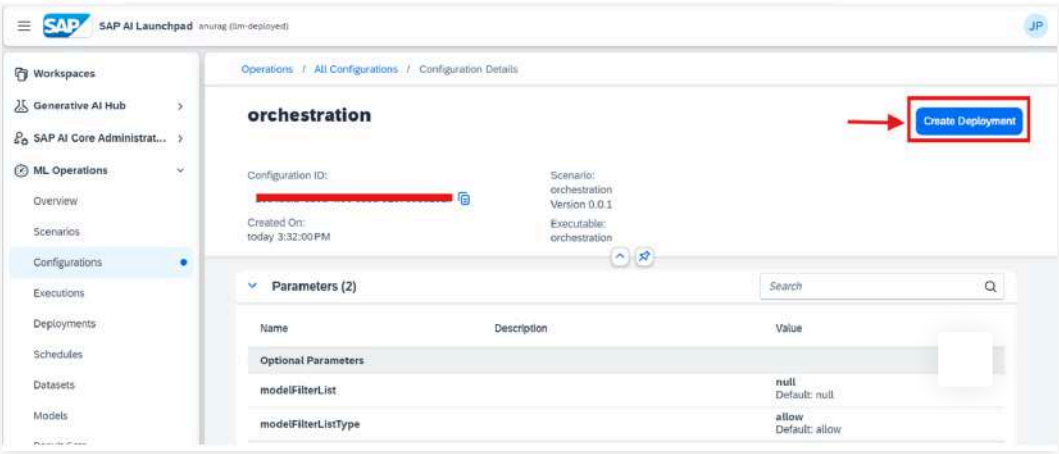
Scenario: "orchestration"

Version: "0.0.1"

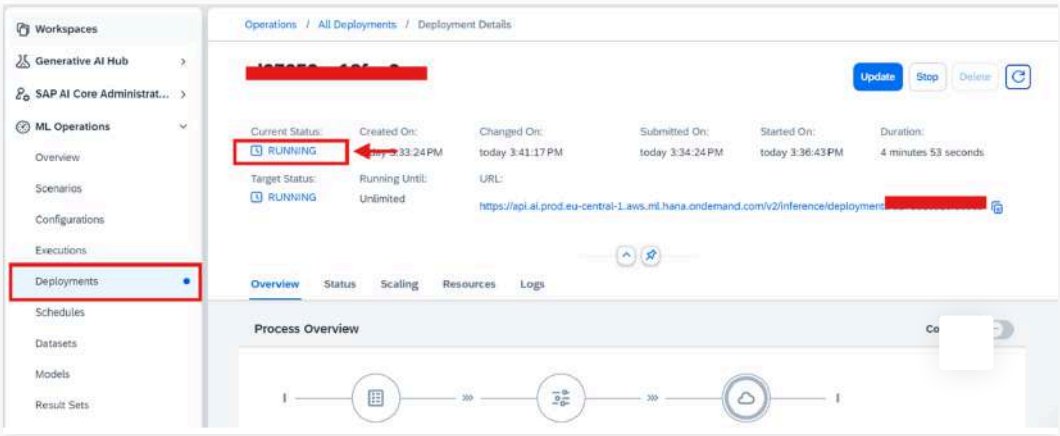
Click **Next.**



When prompted, click **Create Deployment.** Continue through the setup by clicking **Next** until you receive the deployment confirmation.



Once the deployment begins, continue to the status page. Verify that the Deployment Status changes to **Running** (refer to following screenshot for reference).



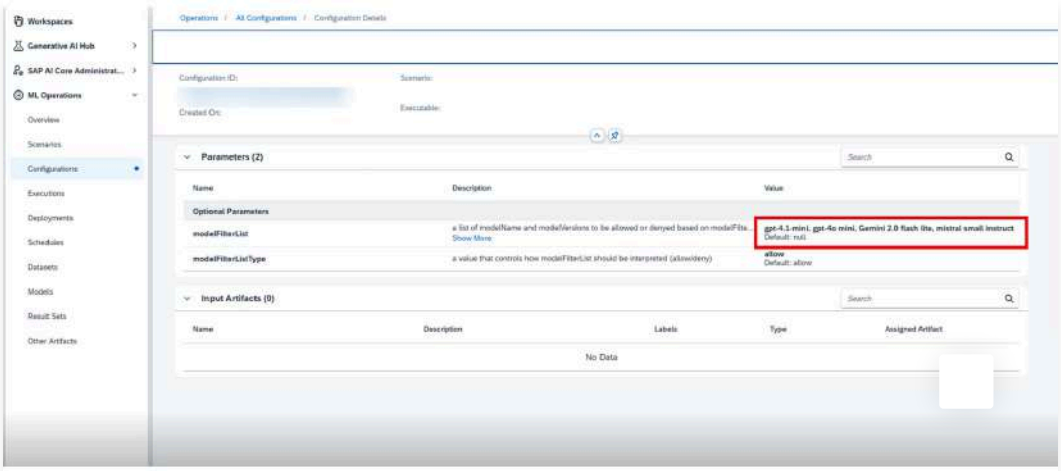
Model Restriction

You can explicitly control which LLMs are used for orchestration deployment. This is crucial for enforcing internal standards, compliance,

- **modelFilterList:** This is a list of specific modelNames and, optionally, modelVersions that you want to either allow or deny. If modelVersions is not defined for a modelName, all versions of that model are considered.
- **modelFilterListType:** This parameter controls how the modelFilterList should be interpreted:
- - **deny:** Excludes the models and defined versions from use within this orchestration deployment. Any LLM in the modelFilterList will not be available.
  - **allow:** Only allows the models and defined versions in the modelFilterList to be used within this orchestration deployment. Any LLM not in the modelFilterList will be unavailable.

This mechanism enables you to implement internal policies, such as only allowing certain LLMs that have been pre-approved for specific types of data or performance characteristics.

An example is shown here:



It ues the following json list in **modelFilterList**.

JSON

This instance is restricted to the specified models. Since no version is specified, the latest available version from the generative AI hub will be used.

This is a powerful way to enforce internal usage policies and temporarily allow or restrict models that are not yet approved for certain tasks.

Deploy LLMs in the Generative AI Hub

Specific models can also be deployed according to your requirements. For example, you can deploy models like cohere--command-a-reasoning , which is not available in the orchestration deployment.

1. **Create a Deployment:** This step involves creating a deployment, which can be done either programmatically through APIs or via the user-friendly interface of the SAP AI Launchpad. When creating a deployment, you will reference a model provider-specific executable – for example, models provided through models from OpenAI (via Azure), GCP Vertex AI, Amazon, Anthropic (via AWS Bedrock), and Meta. You will also configure essential parameters such as the model name (e.g., gpt-4o, Gemini 2.2 Pro, or Claude 3.5 Sonnet) and its version.
2. **Access the Deployed LLM:** Upon successful deployment, SAP AI Core provides a unique, secure URL for each deployment. This URL acts as the endpoint your applications will call to interact with the deployed LLM. This abstraction simplifies how your applications connect to and utilize these powerful models, regardless of the underlying model provider.

lifecycle for your applications. You typically choose one of two main strategies when configuring your deployment: auto upgrade or manual upgrades. This decision impacts control, effort, and application stability.

Automatic Model Upgrades (modelVersion: latest)

This strategy tells your deployment to always use the newest supported version of a specified model.

- **How it Works:** When setting up your LLM configuration (whether for an orchestration workflow or a direct model deployment), you set the modelVersion parameter to latest. This will automatically use the most recent version of the specified model that is supported.
- **Benefits:**
  - **Less Manual Effort:** Your deployment automatically upgrades to a newer version, potentially giving you access to model improvements without needing your intervention.
  - **Continuous Updates:** You automatically receive the latest bug fixes, performance enhancements, and new capabilities for that model.
- **Risks and Limitations:**
  - **Less Control:** Your application might suddenly start using a new model version you haven't tested. This could introduce subtle changes in output behavior or performance.
  - **Potential for Instability:** If your application relies on highly consistent LLM output, unexpected behavioral shifts from a new version can cause issues.
  - **Same Model Only:** This strategy only updates versions of the same model (for example, from an older gpt-4o to a newer gpt-4o version). It does not automatically switch to a completely different model (for example, from gpt-4o to gpt-5).

Manual Model Upgrades (Specific model/Version)

This strategy gives you precise control over which model version your deployment uses.

- **How it Works:**: When creating your orchestration or model configuration, you specify an exact modelVersion, such as "2024-05-13". As you prepare to move to a newer model version, whether because the current version is retiring or you choose to upgrade, follow these steps:

a. **Update Configuration**: Easily change your configuration to the new, specific model version using the harmonized API in orchestration.

b. **Apply (if direct deployment)**: If you are managing your own custom endpoint for a direct LLM deployment, you might then patch your existing deployment to reference this new configuration ID.

- **Benefits:**
  - **Full Control**: You decide precisely which model version your application uses and when to make changes.
  - **Comprehensive Testing**: Allows you to test new model versions comprehensively in a controlled environment before deploying them to production.
  - **Predictable Behavior**: Essential for production systems where consistent LLM behavior is critical, mitigating risks from unexpected changes.
- **Risks:**
  - **More Manual Effort**: You must actively monitor model retirement dates and perform the upgrade steps yourself.

Choosing Your Strategy: Best Practices

The ideal upgrade strategy depends on your application's priorities:

- **For most production applications, especially those sensitive to potential behavioral changes,** specifying a fixed modelVersion (Manual Upgrade) is the **recommended best practice.** This provides the necessary control and predictability. Most application owners prefer this approach to avoid unexpected behavior shifts that can occur with new LLM versions. Fixing the version allows you to plan and test updates proactively.
- **For non-critical or exploratory applications** where minimizing operational overhead is preferred, using Auto Upgrade (modelVersion: latest) can be suitable. However, always be aware that behavior might change with new versions, and you'll still need to manually manage transitions if you decide to switch to a different model entirely.

Lesson Summary

model access within deployments and the considerations for managing model upgrades, balancing the need for control with the desire for automation.

This comprehensive setup is a prerequisite for developing powerful and secure AI-driven applications. It allows you to focus on crafting intelligent prompts and solving business problems without worrying about the

Knowledge quiz

It's time to put what you've learned to the test, get 2 right to pass this unit.

1. What is the primary purpose of the Orchestration Service within the SAP generative AI hub?

Choose the correct answer.

- ○ To manage SAP database connections for large datasets
- ○ To coordinate and execute multiple user interface elements for AI-driven dashboards
- ◉ To coordinate and execute multiple AI dashboards, services, and data flows within a unified workflow
- ○ To train foundation models from scratch for custom enterprise needs

Correct

Orchestration Service within the SAP generative AI hub coordinates and executes multiple AI models, services, and data flows within a unified workflow.

sequence of modules?

Choose the correct answer.

- [ ] To make workflows rigid and prevent customization
- [x] To ensure data is securely processed, grounded in facts, and compliant with enterprise standards
- [ ] To simplify LLM prompt wording for developers
- [ ] To minimize token usage by skipping unnecessary modules

Correct

While the order of orchestration modules is predetermined to ensure robust security, data integrity, and compliance, the configuration and activation of each module are highly flexible and tailored to your specific use case.

YOUR SCORE

2/2

You passed Discovering the Orchestration Service

2 Lessons 35min

Next up: Exploring the Generative AI Hub

Learning

inks
nload Catalog (CSV, JSON, XLSX, XML)
Learning Hub
Training Shop
Developer Center
Community
sletter
ng Support
Support
e Feedback
ase Notes
SAP
pany Information
right
emark
dwide Directory
ers
s and Press
ormation
cy
s of Use

Exploring Use Cases for the Orchestration Service

Objective

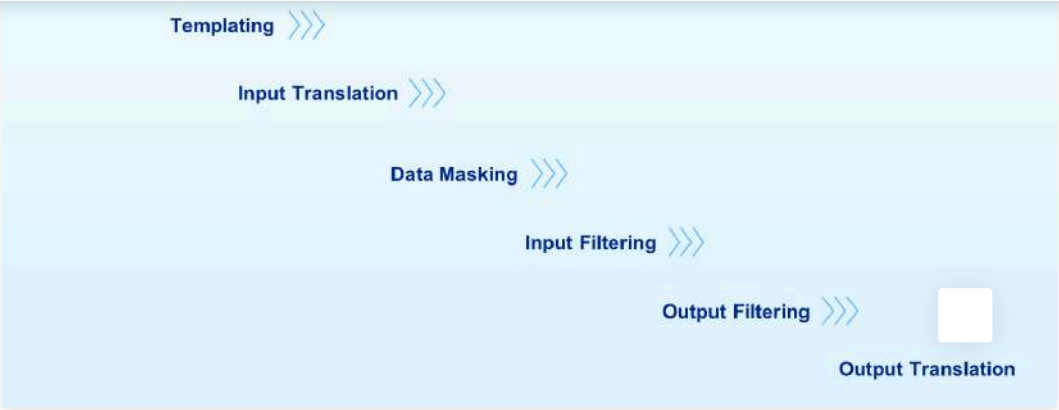
After completing this lesson, you will be able to demonstrate the value of orchestration workflows through practical use cases.

Exploring Use Cases for the Orchestration Service

This lesson moves from theory to practice. We will explore a few orchestration workflows, demonstrating how its specific sequence of steps transforms a powerful LLM into a truly reliable, compliant, and integrated enterprise asset. You will explore how these services work together in a structured yet highly adaptable pipeline to deliver robust AI solutions.

Orchestration Workflow

The Orchestration Service is most effective when its features work together in a specific order. A single API call can trigger a sophisticated, multi-step process that prepares data for the LLM and secures the output. The standard workflow is structured as follows:



1. **User Input:** The initial query or data, typically originating from an end-user or an upstream application.
2. **Grounding:** Based on the user input, the system searches and retrieves relevant, factual, and up-to-date information from designated enterprise data repositories. Examples of these repositories can be SAP Help Portal, internal documentation, master data etc. This ensures the LLM's response is fact-based.
3. **Templating:** The retrieved grounded data, the original user input, and predefined system instructions, which can include LLM persona and rules for response generation, are combined into a comprehensive prompt using a pre-configured template.
4. **Input Masking:** This crucial step scans the templated prompt for Personally Identifiable Information (PII) or other sensitive data (e.g., emails, names, addresses, phone numbers) and pseudonymizes it. This ensures sensitive data is not directly exposed to the LLM.
5. **Input Filtering:** The masked prompt is then scanned by a content safety service (for example, Azure Content Safety) for any harmful, toxic, or inappropriate content before it is sent to the LLM.
6. **Input Translation:** If the primary LLM processes in a language different from the user's input, the entire filtered and masked prompt is translated into the LLM's operating language, for example from German to English.
7. **LLM Processing:** The thoroughly prepared, safe, masked, filtered, and translated prompt is sent to the selected LLM, which processes this input and generates its response (typically in its operating language, such as English).

the final, safe LLM response is translated back into the user's native language for example, English to German.

While the order of orchestration modules is predetermined to ensure robust security, data integrity, and compliance, the configuration and activation of each module are highly flexible and tailored to your specific use case. **This means you control whether a module is active, how it behaves, and with what parameters, allowing the workflow to be both reliable and adaptable.**

For instance, you might configure strict input filtering but looser output filtering or specify different PII entities for masking based on the data being processed. This balance between a fixed structure and dynamic configuration is key to the service's power.

Use Case 1: Automated Technical Support Response

In this case, you will observe an elaborate example of a workflow customized for a specific need.

- **The Business Problem:** A customer submits a technical support ticket in German, asking for help configuring notifications in SAP Signavio Process Manager and including their email address. The support system needs to provide an accurate, policy-aligned response, protect sensitive customer data, and seamlessly handle the language barrier. If a full answer can't be given automatically, it should summarize the issue for a human agent.
- **The Orchestration Workflow in Action:** The support system receives an email, which triggers this workflow with the customer's German message.
- 1. **User Input:** The German support issue: "Betreff: Unterstützung benötigt. Nachricht: Hallo, ich benötige Unterstützung mit SAP Signavio. Insbesondere möchte ich Benachrichtigungen im SAP Signavio Process Manager konfigurieren. Bitte kontaktieren Sie mich mit unter Jane.Janeson@gmx.net."
  2. **Grounding:** The system uses keywords from the German user input like "SAP Signavio Process Manager", "Benachrichtigungen", to query official help.sap.com documentation. Relevant articles on configuring notifications are retrieved as context for the issue or issue-context.

template also incorporates the original German support issue and the retrieved English (or mixed language) grounding context.

4. **Input Masking:** The masking module identifies "Jane.Janeson@gmx.net" within the templated prompt and pseudonymizes it (for example, replaces it with [profile-email]).
5. **Input Filtering:** The filtering module scans this masked prompt for any harmful or inappropriate content that might be present in the raw feedback; this filtering can be configured as 'relaxed' to reduce its stringency and allow a broader range of input. When relaxed, the system has a higher tolerance for content that might otherwise be flagged, often to prevent blocking legitimate business-specific terms.
6. **Input Translation:** This module translates German text to English for better processing by LLM.
7. **LLM Processing:** The LLM (for example, gpt-5) receives the fully prepared, safe, and masked English prompt. It analyzes the feedback against the grounded HR policies and generates a summary and thematic analysis in English.
8. **Output Filtering:** The filtering.output module scans the LLM's generated English summary to ensure it contains no toxic language, bias, or inappropriate suggestions.
9. **Output Translation:** The English output is translated back to German for consumption by the user.

- **Business Value Delivered:**
  - **Ensured Anonymity & Compliance:** Sensitive PII is masked, and content is filtered, aligning with HR data privacy policies.
  - **Rapid Insight Generation:** Quickly identifies trends and issues from large volumes of feedback, enabling faster response from HR.
  - **Grounded Analysis:** LLM's analysis is informed by actual company policies, making it relevant and actionable.
  - **Translation:**The input for LLM and output from LLM is translated for best user experience.
  - **Efficiency:** Automates a labor-intensive review process, allowing HR professionals to focus on strategic initiatives.
  - **Cost Optimization:** By omitting unnecessary translation steps, the workflow reduces processing time and associated costs.

- **The Business Problem:** An internal Human Resources department periodically collects employee feedback via survey forms. These forms are typically submitted in English. The HR team needs to analyze the feedback for common themes, sentiment, and potential policy violations, but it must strictly ensure employee anonymity and filter out any inappropriate content. The final summary and analysis need to be in English for internal reporting.
- **The Orchestration Workflow in Action:** An HR analyst uploads a batch of English feedback forms, triggering the workflow.
- 1. **User Input:** Employee feedback forms.
  2. **Grounding:** The system uses keywords from the feedback (for example, "benefits," "work-life balance") to search and retrieve relevant company HR policies and internal best practices documents from the HR knowledge base. This provides crucial context for the LLM's analysis.
  3. **Templating:** A predefined prompt template is populated. It includes a system instruction: "You are an HR analyst. Summarize the key themes, sentiment, and actionable insights from the employee feedback. Ensure anonymity and highlight any potential policy violations (based on the provided HR policies). Make sure the output is concise and objective." The template integrates the employee feedback and the retrieved HR policies.
  4. **Input Masking:** The masking module scans the templated prompt for any identifying information (for example, employee names, IDs, specific department names if considered sensitive for this context) that might have been inadvertently included in the feedback and pseudonymizes it.
  5. **Input Filtering:** The filtering module scans this masked prompt for any harmful or inappropriate content that might be present in the raw feedback; this filtering can be configured as 'relaxed' to reduce its stringency and allow a broader range of input. When relaxed, the system has a higher tolerance for content that might otherwise be flagged, often to prevent blocking legitimate business-specific terms.
  6. **Input Translation:** (This module is not configured or is effectively skipped for this use case). Since the user input is already in English and the LLM processes in English, translation is not required.

8. **Output Filtering:** The filtering.output module scans the LLM's generated English summary to ensure it contains no toxic language, bias, or inappropriate suggestions.
9. **Output Translation:** (This module is not configured or is effectively skipped for this use case). Since the desired output is English, no translation back to another language is needed.

- **Business Value Delivered:**
  - **Ensured Anonymity & Compliance:** Sensitive PII is masked, and content is filtered, aligning with HR data privacy policies.
  - **Rapid Insight Generation:** This process quickly identifies trends and issues from large volumes of feedback, enabling faster HR response.
  - **Grounded Analysis:** LLM's analysis is informed by actual company policies, making it relevant and actionable.
  - **Efficiency:** Automates a labor-intensive review process, allowing HR professionals to focus on strategic initiatives.
  - **Cost Optimization:** By omitting unnecessary translation steps, the workflow reduces processing time and associated costs.

Designing and Deploying Orchestration Workflows

For developers, it's important to understand that these complex workflows are defined and executed efficiently.

- **Unified API and SDK:** Developers use the SAP Cloud SDK for AI to define an orchestration template, which is a JSON file that specifies the sequence of services. This templated approach is key to balancing the workflow's fixed logical sequence with the dynamic, use-case-specific configurations, enabling developers to precisely define how each step operates for their unique requirements.
- **Single API Call:** Once the template is deployed, the entire multi-step workflow can be executed with a single, unified API call. The application simply sends the initial query to the Orchestration Service endpoint, which manages the entire chain of events internally. This abstracts away the complexity of calling multiple different services and handling data between them.

This approach dramatically simplifies development, ensures consistency, and allows for rapid deployment and modification of sophisticated AI-

practical, real-world examples that follow the official SAP workflow. You understand that starting with **Grounding** is key to enterprise-grade accuracy, and that post-processing steps like **Content Filtering** and **Data Masking** are essential for security and compliance.

This inherent structure provides enterprise-grade reliability and predictability, while the configuration within each module ensures the workflow can be precisely tailored to diverse business problems. By sequencing these services, you can transform general-purpose LLMs into specialized, data-driven, and highly secure business tools, enabling you to build sophisticated and trustworthy AI solutions.

Discovering the Orchestration Service

Objective

After completing this lesson, you will be able to identify the orchestration service.

Discovering the Orchestration Service

The generative AI hub provides access to a wide range of LLMs, offering features such as model selection and prompt management through the SAP AI Launchpad. Now, as you prepare to build more sophisticated and robust AI applications, you will have workflows that need integration for multiple tasks such as data filtering or data anonymization etc.

This is where the **Orchestration Service** in the generative AI hub becomes indispensable. It elevates your generative AI capabilities beyond individual LLM interactions, enabling you to design, manage, and execute complex AI workflows. This lesson will introduce you to this powerful service, clarifying its purpose, essential features, and how it ensures your AI solutions are not only intelligent but also compliant, efficient, and scalable within the enterprise.

The Orchestration Service is a managed service within **SAP AI Core** that provides unified access, control, and execution of generative AI models. Orchestration here refers to the systematic management and coordination of multiple AI models, services, and data flows to achieve a unified business objective. It functions as a central coordinator, managing multiple AI models and services to complete complex tasks.

The Orchestration Service streamlines the integration and management of various AI models, enabling businesses to efficiently utilize advanced AI features without changing their application code whenever models or versions are updated.

Orchestration is Essential for Enterprise AI

The Orchestration Service addresses several critical challenges inherent in building production-ready AI applications:

- **Seamless Integration and Provider Agnosticism:** In a rapidly evolving AI landscape, you might need to use different LLMs from various providers for specialized tasks, performance, or cost reasons. The Orchestration Service provides a harmonized API that allows your applications to interact with different foundation models without being tightly coupled to a specific provider. This means you can switch or compare models easily without altering your core application logic.
- **Enhanced Control and Compliance:** Enterprise solutions demand strict adherence to standards, privacy regulations, and ethical guidelines. The Orchestration Service offers built-in mechanisms to ensure compliance with SAP standards and provides centralized control over your AI workflows. For example, it provides features like data

workflow where the output of one module can automatically serve as the input for the next, streamlining processes and automating multi-step tasks. Furthermore, it supports the deployment of these orchestration workflows, providing the flexibility and scalability needed for high-volume enterprise use cases.

- **Expandability and Adaptability:** The dynamic nature of the AI market means new capabilities constantly emerge. Orchestration offers inherent expandability, allowing you to easily add new features like content filtering, data masking, grounding, and translation as needed, ensuring your AI solutions can adapt to changing technical and commercial requirements efficiently.

Key Features of the Orchestration Service

The Orchestration Service provides a suite of powerful modules that can be chained together. You will learn that while the sequence of these modules is structured to enforce logical processing and security, each module's configuration offers extensive flexibility for diverse use cases. These include:

- **Grounding:** The Orchestration Service facilitates the integration of external, domain-specific, or real-time data sources (like your SAP systems) to enhance the contextual relevance and factual accuracy of AI model outputs. This is a critical mechanism for combating hallucinations and ensuring reliability.

- **Content Filtering:** To maintain compliance and safety, the service can restrict the type of content passed to and received from generative AI models. This helps prevent the generation or processing of inappropriate or sensitive information.
- **Data Masking:** Supports the anonymization or pseudonymization of sensitive data before it is processed by generative AI models. Crucially, it also offers the ability to unmask data in responses when pseudonymization is used, ensuring data privacy while maintaining utility.
- **Translation:** For global enterprises, the service enables the translation of input and output data directly within the orchestration workflow, supporting multilingual use cases and breaking down language barriers.

Prerequisites for Using the Orchestration Service

To begin leveraging the Orchestration Service, certain foundational elements must be in place:

- **SAP BTP Account:** You need an active SAP BTP account as the underlying platform.
- **SAP AI Core Instance:** An instance of SAP AI Core must be set up within your BTP account, as the Orchestration Service runs on this infrastructure.
- **Extended SAP AI Core Service Plan:** An extended service plan for SAP AI Core is typically required, as the generative AI hub capabilities, including orchestration, are not usually available in free or standard tiers.
- **Orchestration Deployment:** Ensure that at least one orchestration deployment is created or used within a resource group, providing the executable definition of your AI workflow. A resource group will have a default deployment. You can create and use your own deployment to customize available models.

Using SDK

The SAP Cloud SDK for AI offers a straightforward way to begin using the Orchestration Service for AI-powered tasks. By setting up a template and connecting it to an LLM like GPT-5 or Gemini 2.5 Pro, users can automate processes such as translation or content generation with minimal effort. The SDK handles the orchestration logic, allowing users to define inputs,

run tasks using predefined templates, and get consistent outputs tailored to their needs. It's a practical starting point for integrating generative AI into everyday operations.

Lesson Summary

You've now been introduced to the **Orchestration Service** in SAP AI Core, recognizing its pivotal role in building complex, enterprise-grade Generative AI applications. You understand its purpose in unifying access and control over diverse LLMs, and its importance in ensuring compliance, efficiency, and scalability. You've also learned about its key features like templating, content filtering, data masking, grounding, and translation, which empower you to create robust and reliable AI workflows. This

Knowledge quiz

It's time to put what you've learned to the test, get 3 right to pass this unit.

1. Which type of interaction exemplifies "Software-to-Software via AI" in the provided examples?

Choose the correct answer.

- [x] An automated recruitment workflow initiated by an AI-driven service in a connected applicant tracking system.
- [ ] A developer describes a desired feature and the AI produces the code structure for the service.
- [ ] A business analyst gets a textual summary of customer support tickets.
- [ ] A backend system feeds sales order data to the AI, generating a natural language report.

a recruitment workflow in a connected applicant tracking system.

2. What is the primary role of SAP's AI Foundation within SAP Business Technology Platform (SAP BTP)?

Choose the correct answer.

- ○ To provide analytical and visualization tools for non-business users
- ● To serve as SAP's AI operating system, enabling secure, scalable, and responsible AI development and operations
- ○ To serve as SAP's AI database, enabling secure, scalable, and responsible AI development and operations
- ○ To augment SAP HANA as the main database for AI workloads

responsible AI development and operations,

3. Which of the following components are available within the generative AI hub interface in SAP AI Launchpad? (Select all that apply)

There are three correct answers.

- [x] Model Library for model selection and comparison
- [x] Chat interface for interactive model testing
- [x] Grounding Management for data pipeline lifecycle
- [ ] Prompt Editor and Prompt Management for code management and versioning

Correct

The generative AI hub interface in SAP AI Launchpad has Model Library for model selection and comparison, Chat interface for interactive model testing, Grounding Management for data pipeline lifecycle, and Prompt Editor and Prompt Management for prompt creation and reuse.