



salesforce  
Agentforce  
World Tour  
*Copenhagen*

# Salesforce Hands-on Training

## Unlock AI Capabilities in Your Custom Apps

Exercise Guide



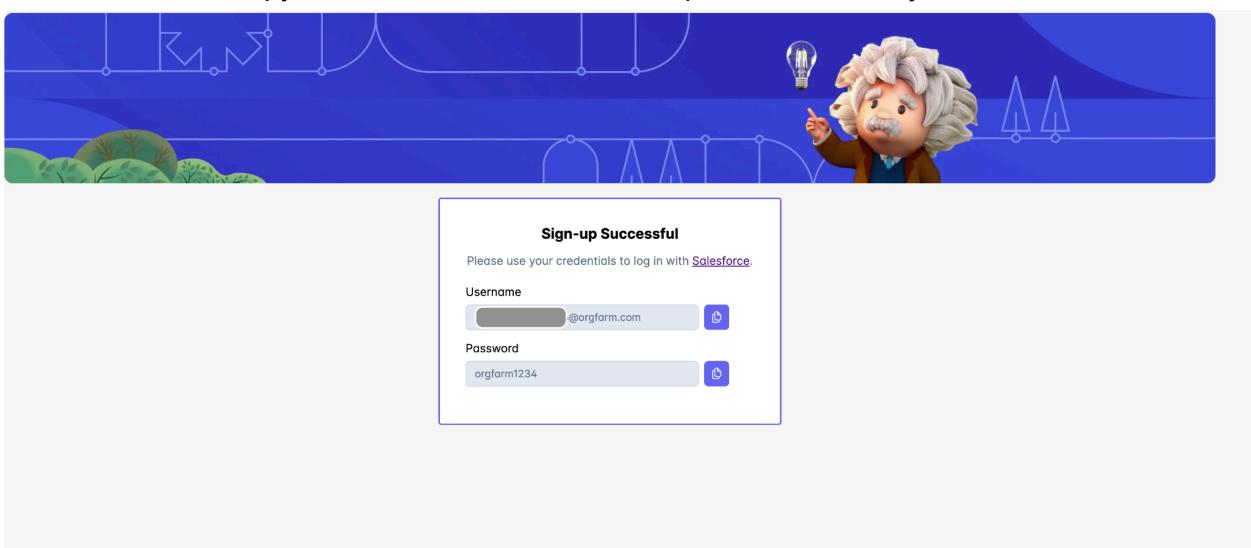
Learn more at: <http://sfdc.co/successplanresources>

Astro intentionally left this page blank....

# Obtain your Salesforce org and Log in

## Task 1: Obtain an org using orgfarm and sign into the org

1. Open <http://sfdc.co/OrgFarmSignup> in your browser
2. Use the event code **provided on screen**.  
**Attention: The code is case sensitive**
3. Fill in the rest of the form and submit
4. Once you've submitted the form you should see a confirmation page like the one shown below. Be sure to copy and save the username - the password is always the same



5. Click the **Go to Salesforce Login** button
6. Fill in your username and password and click **Log In**

## Exercise Guide

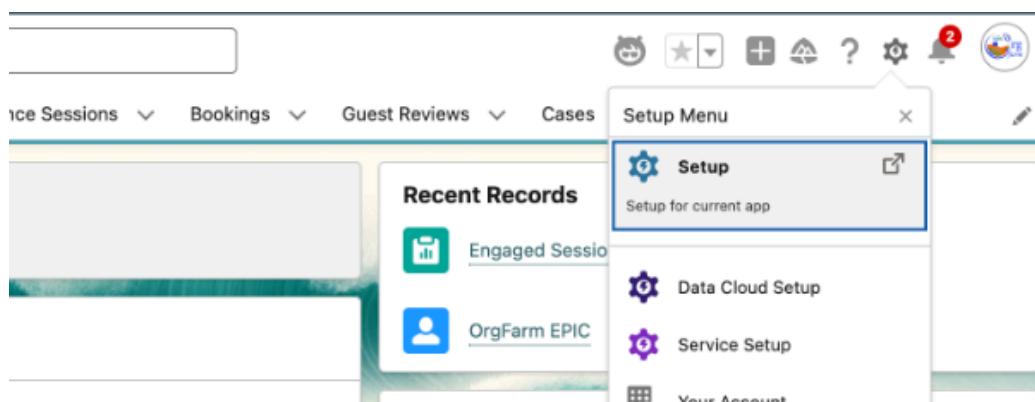
### Unlock AI Capabilities in Your Custom Apps Workshop

## Initial setup

Before we start with the exercises we will do some quick initial setup. This is to allow certain processes to run during the first exercises.

### Instructions:

1. Download these 4 files to your local machine:
  - [experiences.pdf](#)
  - [hotel-policies.pdf](#)
  - [valet-services-policies.pdf](#)
  - [wedding-planning-guide.pdf](#)
2. In your org go to **Setup**



3. In the quick find box type **data library** (Agentforce Data Library)
4. Click on **New library**
5. Configure as follows:

Parameter	Value
Name	Coral Cloud Documents
API Name	[Keep default]
Description	Documents containing unstructured information from Coral Cloud Resorts

## Exercise Guide

### Unlock AI Capabilities in Your Custom Apps Workshop

Add a Data Library

---

\* Select a Data Space  
default

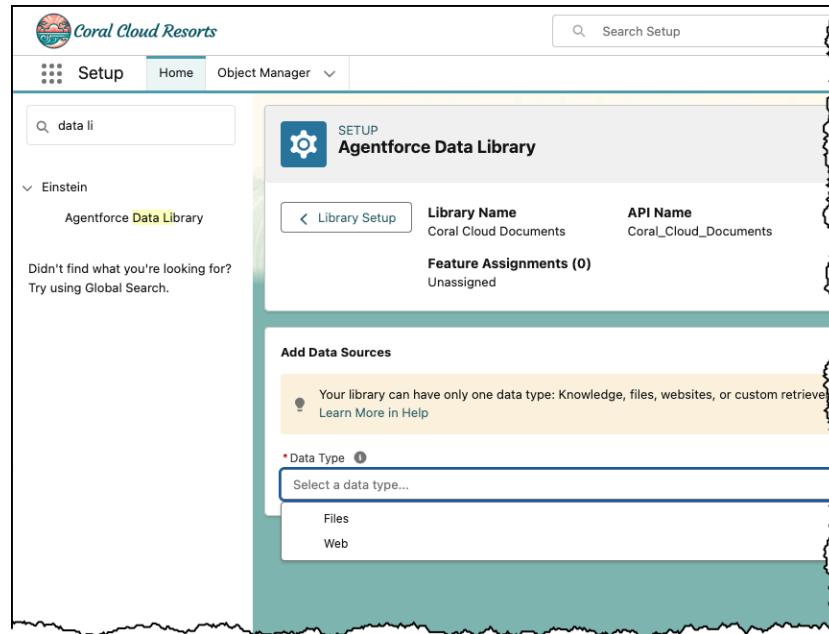
\* Name      \* API Name ⓘ  
Coral Cloud Documents      Coral\_Cloud\_Documents

Description  
Documents containing unstructured information from Coral Cloud Resorts

---

Cancel      Save

6. Click **Save**
7. For Data Type select “Files”



8. Add the 4 files you downloaded previously (click **Upload Files** or drag and drop). Once the file upload is done click **Done** to close the upload dialog.
9. This process will automatically build a DLO (Data Lake Object), a DMO (Data Model Object), chunk the data (break it into smaller parts), initialize the vector database and the search retriever and start the indexing of the documents. In total, this process will take around 15 minutes. We will use this unstructured data in [exercise 3](#).

## Exercise Guide

Unlock AI Capabilities in Your Custom Apps Workshop

# Exercise 1: Using a Basic Prompt - Field Generation Template

## Task 1: Enable Einstein Features

1. Ensure you are still in **Setup** (otherwise reopen **Setup**).
2. Enable Einstein:
  - a. Use **Quick Find** to search for and open **Einstein Setup**.
  - b. Verify that the toggle next to “Turn on Einstein” is turned on. If not, toggle the **Turn on Einstein** slider to **On**.
  - c. Refresh your browser page if you had to turn it on.

### Turn on Einstein

Enhance your Salesforce data with generative AI to create relevant, customized experiences for your users. [Learn more in help.](#)



## Task 2: Create a Field Generation Template

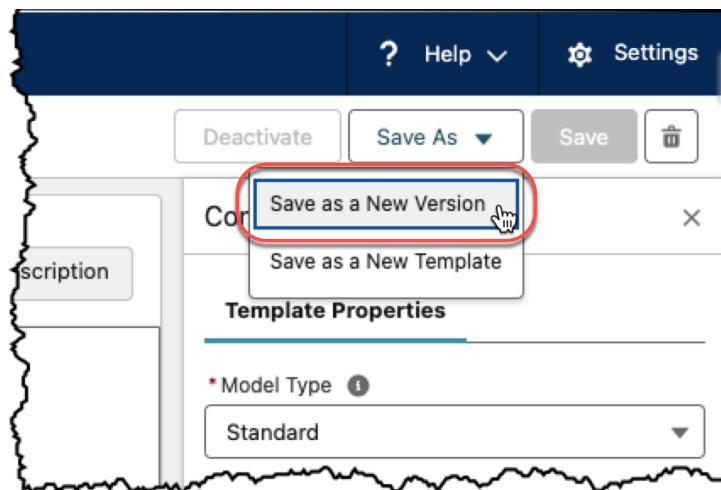
1. Use Quick Find to search for and open **Prompt Builder**. If you cannot find **Prompt Builder** you most likely had to turn on the Einstein feature so ensure you reload the page.
2. Click the **Generate Experience Description** template in the list. The semi-complete prompt will open. As you can see some of the fields cannot be found so we need to fix it.
3. Replace the field placeholders (where it says “missing field”) with the corresponding merge fields (use the **Resource** field to access merge fields):

Placeholder	Merge Field	API Name
ORGANIZATION_NAME	<b>Current Organization &gt; Name</b>	Organization.Name
EXPERIENCE_NAME	<b>Experience &gt; Experience Name</b>	Input:Experience__c.Name
EXPERIENCE_TYPE	<b>Experience &gt; Type</b>	Input:Experience__c.Type__c
EXPERIENCE_LOCATION	<b>Experience &gt; Location</b>	Input:Experience__c.Location__c
EXPERIENCE_DURATION	<b>Experience &gt; Duration (Hours)</b>	Input:Experience__c.Duration_Hours__c
EXPERIENCE_LEVEL	<b>Experience &gt; Activity Level</b>	Input:Experience__c.Activity_Level__c
EXPERIENCE_CAPACITY	<b>Experience &gt; Capacity</b>	Input:Experience__c.Capacity__c

## Exercise Guide

### Unlock AI Capabilities in Your Custom Apps Workshop

4. Click **Save As** and then **Save as a new Version**



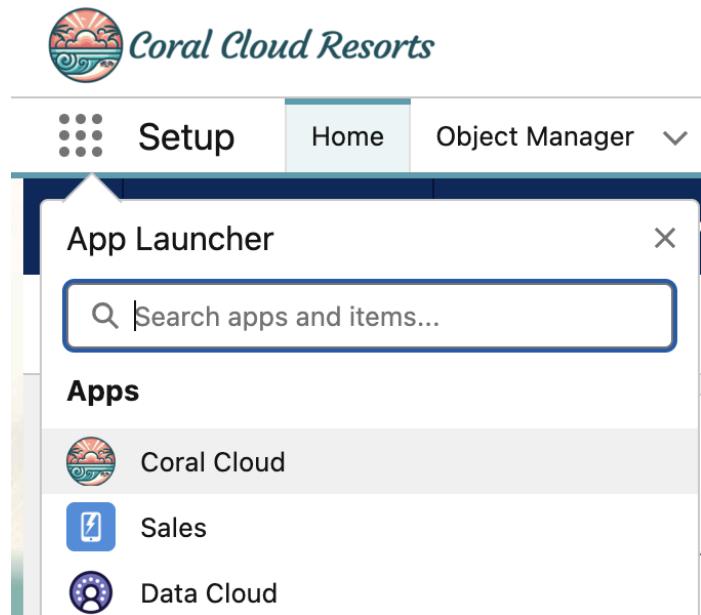
5. "Aqua Fitness in Paradise" is one of the experiences offered at Coral Cloud. To preview the template, find **Aqua Fitness in Paradise** in the Related Record field.
6. Click **Preview**.
7. In the **Resolution** panel, examine the prompt that was generated. Notice that the merge fields have been replaced with the actual values from the "Aqua Fitness in Paradise" record.
8. In the **Response** panel, examine the experience description that was generated by the LLM (Large Language Model).
9. Try adjusting the template instructions. For example, change the number of words you want the generated description to have. Select **Save & Preview** again to review the results.
10. Optional: You could also ask the LLM to generate the response as a rhyme. To do this add "Generate the response as a rhyme." at the end of the template.
11. Optional: Try out another out of the box LLM Model in the Template Properties.
12. Click **Activate** when you're satisfied with your template.

## Exercise Guide

### Unlock AI Capabilities in Your Custom Apps Workshop

#### Task 3: Test the Template to the Experience Description Field

1. In the **App Launcher**, open the **Coral Cloud** app



2. Go to the **Experiences** tab. Open the **Aqua Fitness in Paradise** record.
3. Click **Edit** (the pencil icon ) next to the **Description** field.
4. Click the **Sparkle-icon** next to the **Description** input field to get Einstein's help in creating this field value. It will take a few seconds for Einstein to create a description. Any changes you made in the Prompt Builder will be reflected here.

**Please note:** If you cannot see the Sparkle-icon you most likely had to turn on the Einstein feature so ensure you reload the page.

**Please note:** If you cannot see the updated template being used you most likely didn't Activate the template.

5. Click **Use** to add the generated description to the Description field.
6. Click **Save** to save the record.

## Exercise 2: Using a Flex Template

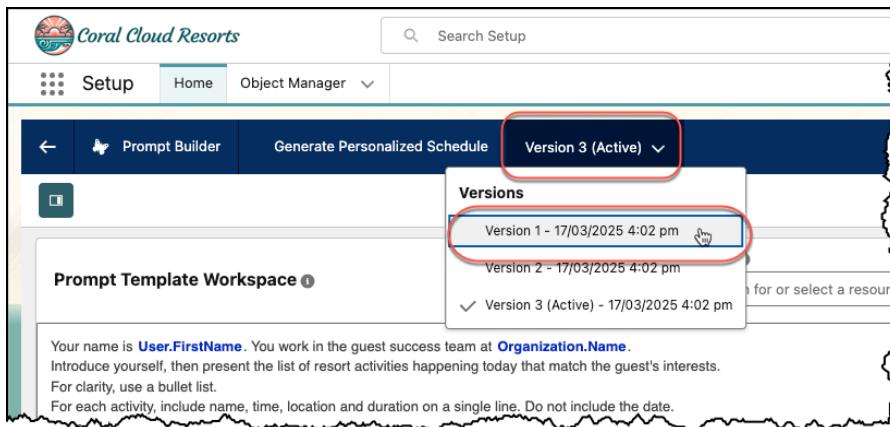
#### Task 1: Choose the right version of the Flex Template

1. Return to **Setup**, use **Quick Find** to search for and select **Prompt Builder**.
2. Click the **Generate Personalized Schedule** template in the list

## Exercise Guide

### Unlock AI Capabilities in Your Custom Apps Workshop

3. The template opens to version 3 of this template. Prompt Builder allows you to switch between different versions of the template in the Prompt Builder using the dropdown menu at the top of the builder (next to the name of the template).
4. Switch to **Version 1** of this prompt. The builder will switch to the below prompt template



Your name is `User.FirstName`. You work in the guest success team at `Organization.Name`.

Introduce yourself, then present the list of resort activities happening today that match the guest's interests.

For clarity, use a bullet list.

For each activity, include name, time, location and duration on a single line. Do not include the date.

Tell the guest to reach out if they'd like to book any of these activities.

List of activities happening today that match the guest's interests:

5. In the Preview panel, choose **Sofia Rodriguez** as the Contact.
6. Click **Preview**.
7. In the **Resolution** panel, examine the prompt that was generated.
8. In the **Response** panel, examine the personalized list that the LLM generated.

### Task 2: Switch to Version 3 of the Prompt Template using Secure Data Grounding

1. Switch back to Version 3 of the **Generate Personalized Schedule** prompt template
2. Notice the prompt is now **grounded** with source data coming from an Apex class. Think out grounding like merging actual data into the prompt before sending it to the LLM.
3. In the Preview panel, select **Sofia Rodriguez** as the Contact.
4. Select **Preview**.
5. Optional: Review the Apex class used for the dynamic grounding below - as you can see it's normal Apex code using the `@InvocableMethod` annotation:

## Exercise Guide

### Unlock AI Capabilities in Your Custom Apps Workshop

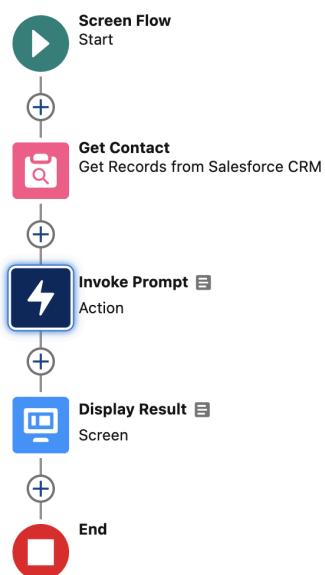
```
public with sharing class PersonalizedGuestExperiences {  
    // Make this method available for grounding  
    // the Generate_Personalized_Schedule prompt template  
    @InvocableMethod(label='Generate_Personalized_Schedule')  
    public static List<Response> getSessions(List<Request> reqs) {  
        Request input = reqs[0];  
        Contact contact = input.myContact;  
        List<Session__c> sessions =  
            ExperienceSessionController.getSessions(contact);  
  
        // Create expected response  
        List<Response> responses = new List<Response>();  
        Response res = new Response();  
        res.prompt = JSON.serialize(sessions);  
        responses.add(res);  
        return responses;  
    }  
  
    // the variables in this class used for the result  
    // must match the names used in the prompt template  
    public class Request {  
        @InvocableVariable(required=true)  
        public Contact myContact;  
    }  
    public class Response {  
        @InvocableVariable  
        public String Prompt;  
    }  
}
```

## Exercise Guide

### Unlock AI Capabilities in Your Custom Apps Workshop

#### Task 3: Use the Prompt in a Screen Flow

1. In Setup, search for and select **Flows**.
2. Open the **Personalized Schedule** flow.
3. Review the Flow



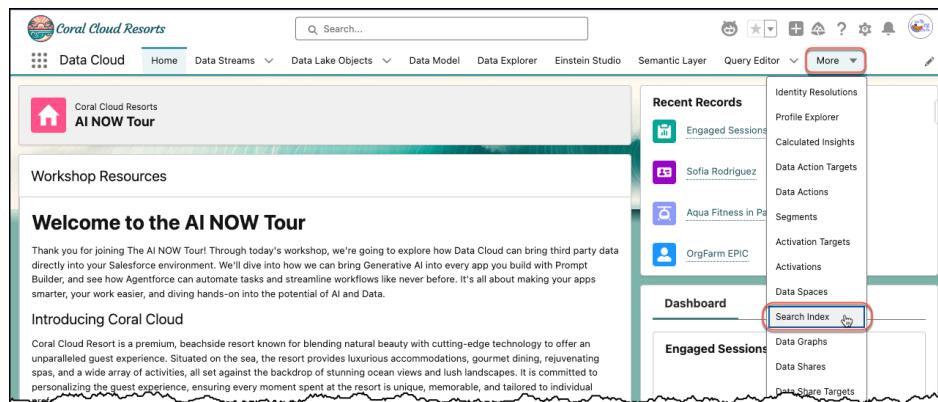
4. Here's a breakdown of what the Flow does:
  - a. In the "Get Contact" element the Contact record open in Salesforce gets loaded
  - b. It then uses out-of-the-box `Invoke Prompt` Action to run the "Generate Personalized Schedule" Prompt Template we were just looking at
  - c. It then displays the result using a Screen element
5. Back in Salesforce use the **App Launcher**, to open the Coral Cloud Resort app if not already open. Use global search to search for and open the Contact record for "**Sofia Rodriguez**"
6. Click the **Personalized Schedule** Quick Action and observe the pop-up screenflow with a personalized schedule.
7. Click **Finish** to close the dialog

## Exercise Guide

### Unlock AI Capabilities in Your Custom Apps Workshop

## Exercise 3: Building the Prompt for RAG

By now your Search Index should have been created and updated. You can check this by going to the App Launcher and opening the **Data Cloud** app. Now open the **Search Index** tab. You might need to click the **More** dropdown to the right of the tabs to find it - Data Cloud comes with many features...



The **Search Index Last Run Status** (the last column) should show **Ready**. If not there is no reason to continue. Yet. Have a bit of patience and then raise your hand and call for an expert to come take a look if too much time passes...

The screenshots show the Data Cloud app's Search Index list. Both screens have a header with columns: Configuration, Data, Search, API Name, Source Data, Index Data, Chunk Data, File, Last Modified, Search Index, and Search Index Status. A search bar and various filter and sort icons are at the top.

	Configuration	Data	Search	API Name	Source Data	Index Data	Chunk Data	File	Last Modified	Search Index	Status
1	FileUDMO...	defa...	Hybr...	FileUDMO...	RagFileUD...	FileUDMO...	FileUDMO...	FileUDMO...	3/26/2025, ...	3/26/2025, ...	In Prog...
2	EAC - Docu...	defa...	Hybr...	EAC_Docu...	EAC_Docu...	EAC_Docu...	EAC_Docu...	EAC_Docu...	3/17/2025, ...	3/17/2025, ...	Ready

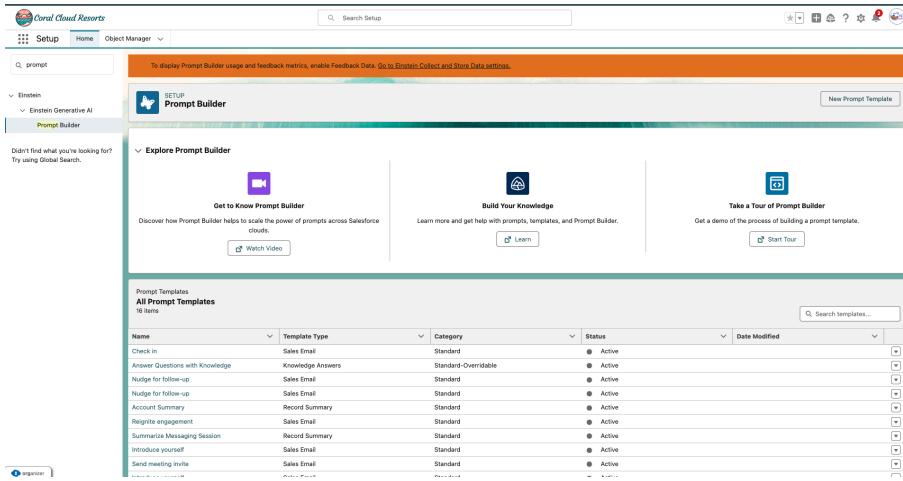
## Exercise Guide

### Unlock AI Capabilities in Your Custom Apps Workshop

#### Task 1: Creating a Prompt Template

In order to use the unstructured data we need to connect the retriever (the element that does the searching for you) to a prompt template. Let's create a new Prompt Template from scratch.

1. Go to Setup
2. Using quick search find and open **Prompt Builder**
3. Click **New Prompt Template**.



4. Select **Flex** as the Prompt Template Type
5. Use the details below to complete the rest of the form

Parameter	Value
Prompt Template Type	Flex
Prompt Template Name	Coral Cloud Documents
API Name	[Keep the default] (Coral_Cloud_Documents)
Template Description	Prompt Flex Template to semantically search for answers in unstructured documents from Coral Cloud Resorts
<b>Define Sources</b>	
Name	Query
API Name	myQuery
Source Type	Free Text

## Exercise Guide

### Unlock AI Capabilities in Your Custom Apps Workshop

#### New Prompt Template

Select the type of prompt template to build. Based on the template type, define the resources available for the template. [Learn more in Salesforce Help.](#)

\* Prompt Template Type

\* Prompt Template Name  \* API Name

Template Description

Define Sources  
Define up to 5 sources for your flex template. Assign a unique label, API name, source type, and object, as applicable. Keep in mind that you can't associate an object with String source types.

1. \* Name  \* API Name  \* Source Type

git

6. Click **Next** to create the empty prompt template
7. Copy/paste the below text into your prompt template - the text can be found in a Gist on Github at [http://sfdc.co/agentforce\\_wt\\_workshop](http://sfdc.co/agentforce_wt_workshop):

Unset

Clearly answer the user's Query directly and logically based only on well-reasoned deductions drawn from the Context below.

Then respond to the user's Query logically, methodologically, thoughtfully and thoroughly from multiple perspectives, emphasising different viewpoints based on Context with details and careful reasoning.

Provide details with organized structure in your response. Consider alternative perspectives or approaches that could challenge your current line of reasoning. If you don't know how to answer the query, or if there is not sufficient context, please respond with 'Sorry, I couldn't find sufficient information to answer your question.'

Evaluate the evidence or data supporting your reasoning, and identify any gaps or inconsistencies.

Clearly articulate with details what are facts versus what are opinions or beliefs.

If you don't know the answer, ask questions to clarify the user's intent.

Pay attention to the entities mentioned in the user's Query and make sure the context contains information about those entities.

Context:

{!\$EinsteinSearch:<add your own>}

## Exercise Guide

### Unlock AI Capabilities in Your Custom Apps Workshop

Query:

```
{!$Input:myQuery}
```

Format instructions:

Format your response with Markdown structures as follows:

Start with an overview of the topic.

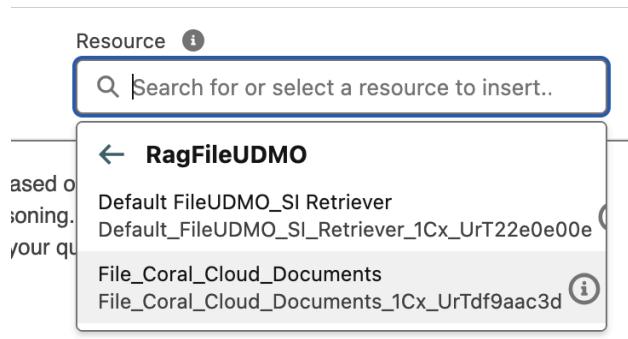
List the key points in a list and emphasise any critical terms using bold.

For subsequent sections, create headings and subheadings that incorporate the subqueries implicitly.

If there are any steps or sequential data, present them in an ordered list.

End with a conclusion.

8. Next, update the prompt (after “Context:”) with your org specific Retriever API Name using the Resources.
  - a. First delete `{!$EinsteinSearch:<add your own>}`
  - b. In its place add the “Einstein Search” → RagFileUDMO → “File\_Coral\_Cloud\_Documents” retriever from your org using the Resources.



9. Next, click on that added retriever. This displays additional configuration for the retriever in a panel on the right.
  - a. Select the **Search Text** as your defined **Free Text** → **Query**
  - b. Select **Chunk** and **SourceRecordId** under Output Fields

## Exercise Guide

### Unlock AI Capabilities in Your Custom Apps Workshop

- c. Compare your Configuration with the screenshot below to ensure everything is configured correctly.

The screenshot shows the 'Configuration' interface for the 'File\_Coral\_Cloud\_Documents' object. It includes sections for 'Description' (File\_Coral\_Cloud\_Documents), 'Data Model Object' (RagFileUDMO) and 'Version' (1). The 'Search Parameters' section contains a search bar with 'Input:myQuery' and a character limit of 17/255. The 'Output Fields' section lists 'Chunk' and 'SourceRecordId'. The 'Number of Results' field is set to 10.

10. Only if you did NOT use the exact names specified when you create the prompt template - back in your prompt replace the `{!$Input:<myQuery>}` with your Free Text Query Field from the Resources

The screenshot shows the 'Prompt Template Workspace' with a 'Resource' search bar containing 'Free Text' and a result 'A\_0 Query'.

11. You can now test the prompt template by clicking the **Test Inputs** button in the **Preview** pane. This will open a pane on the right where you can test with your own queries. Below are some examples for you:

- What is the cancellation policy for Coral Cloud Resorts?
- List the options for hosting a Wedding at Coral Cloud Resorts
- What is the cost of the Family Adventure Quest?

## Exercise Guide

### ***Unlock AI Capabilities in Your Custom Apps Workshop***

12. Test the query by clicking **Preview** (or Save & Preview)

**Please note:** If you see "searchResults" : [] in your Resolution this means you either have selected the wrong retriever, or your Search Index has not completed (no results are returned).

13. Click **Activate** to activate the prompt template. This will be important later.

## Exercise Guide

*Unlock AI Capabilities in Your Custom Apps Workshop*

# Exercise 4: Agentforce

### Task 1: Enable Agentforce

1. Open **Setup**
2. Enable Einstein Copilot:
  - a. Use **Quick Find** to search for **Agents** and open **Agentforce Agents**.
  - b. Verify that Agentforce is turned on. If not, toggle the **Agentforce** slider to **On**.
  - c. Also enable the **Agentforce (Default)** agent
3. If you had to enable Agentforce the page refreshes to display a list view
4. You should see **Einstein Copilot** and **Coral Cloud Agent** in the list (if this does not appear automatically, refresh the page).

### Task 2: Explore and Activate the “Coral Cloud Agent” Agent

1. Click on the **Coral Cloud Agent** agent to open it.
2. Notice the five tabs on the **Agent Details** page: Details, Topics, System Messages, Language Settings, and Connections. Take a minute to click each tab to explore the agent settings.
3. Click on the **Open in Builder** button to launch **Agent Builder**.
4. Let's take a quick tour. There are three main UI elements on the page:
  - a. On the left - Agent Details, such as Topics, Context, etc.
  - b. In the middle - Interaction preview of how the planner service reasons including the actions chosen
  - c. On the right - Conversation preview of the agent for testing the agent

### Task 3: Test-drive the “Coral Cloud Agent” Agent

1. In the Conversation Preview pane on the right type the below piece of text in the box at the bottom. This is the text you send to the agent. Press Enter to submit the text or click the arrow:  
*Draft an email concept to promote the Aqua Fitness in Paradise experience. Include some information about the experience.*
2. Observe how the agent responds to you in the pane on the right and show you how it reasoned about the text you sent it in the middle pane.

## Exercise Guide

### Unlock AI Capabilities in Your Custom Apps Workshop

3. See how it shows you the topic it chose and which action (or actions...) it invoked to perform the action and / or generate input for the response. This is some of the power of Agentforce!

The screenshot illustrates the Agentforce workflow:

- Select Action:** Shows the action "Get Experience Details (Get\_Experience\_Det...)" with a duration of 0.39 sec. The input is a JSON object with "experienceName": "Aqua Fitness in Paradise". The output is another JSON object with "experienceRecord": {"sObjectInfo": {"apiName": "Experience\_\_c", "id": "000000000000000000"}, ...}.
- Reasoning:** A box contains the text: "GROUNDED: The response is grounded in the details provided by the 'Get Experience Details' function, which includes the description and features of the 'Aqua Fitness in Paradise' experience."
- Agent Response:** The response message is: {"message": "Subject: Dive into Fitness with Aqua Fitness in Paradise!\n\nDear [Recipient's Name],\n\nAre you ready to make a splash and get fit in the most refreshing way possible? Join us for Aqua Fitness in Paradise! We offer a variety of water-based activities designed to challenge your body and mind. From swimming lessons to aqua aerobics, there's something for everyone.\n\nContact us today to learn more about our programs and how you can become part of our community. We look forward to welcoming you soon!\n\nBest regards,\n[Aqua Fitness in Paradise] Team"}.

4. In the Conversation Preview pane on the right try asking something it shouldn't try to answer. You could use the following text:

*I'm thinking about building my own resort - what would be good experiences to provide?*

5. If you chose the above text observe how the agent switched topic and classified the question as "Off Topic" i.e. something it should not try and respond to. This is the guardrails in action. The agent should know what to respond to and what not to respond to.

The screenshot shows the following steps:

- User Prompt:** The user asks: "I'm thinking about building my own resort - what would be good experiences to provide?"
- Select Topic:** The topic chosen is "# Off Topic (Off\_Topic)". It is described as "Used when a user asks about off topic subjects".
  - Instructions:** 2
  - Actions:** 0
- Reasoning:** A box contains the text: "SMALL\_TALK: The response is small talk, redirecting the conversation to support-related topics as instructed. Nothing in the response indicates a prompt injection."
- Agent Response:** The response is: "I'm sorry, but I'm not able to assist with that request. Please let me know if there's anything else I can help you with."

## Exercise Guide

### Unlock AI Capabilities in Your Custom Apps Workshop

## Customize the Einstein Employee Facing Agent

With Agentforce you can create both customer and employee facing agents. So far the agent you have been working with has been a customer facing agent. In this section you will customize the employee facing Einstein Copilot agent to help handle the management of guest experiences and questions from guests.

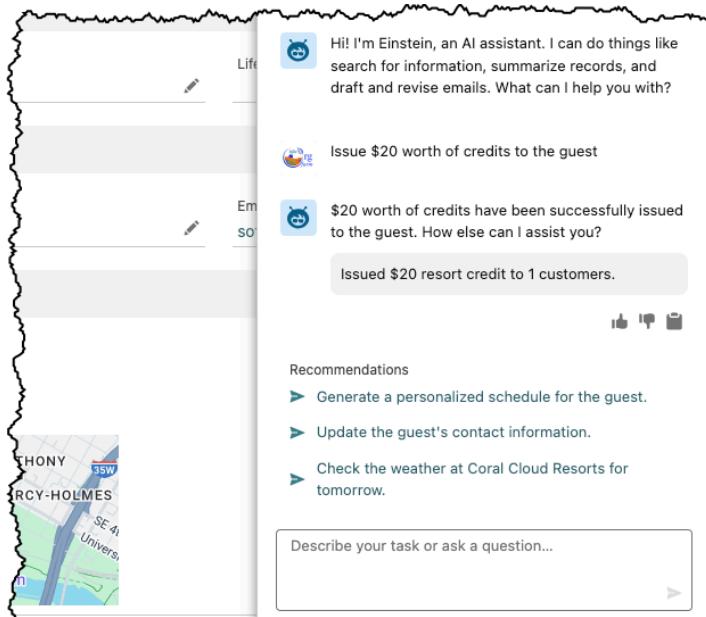
You will extend the agent to use the prompt template you created in the previous exercise. This will help Coral employees, especially those working the reception desk, to suggest a personalized daily schedule for a guest.

### Task 1: Activate the employee facing agent

1. Open **Setup**
2. Using Quick Search find and open **Agents**
3. As you can see the Employee agent called “Einstein Copilot” is not active. Click the dropdown arrow and click “Open in Builder”. This opens the agent in Agent Builder.
4. Click the **Activate** button
5. Exit Agent Builder confirming you would like to leave.

### Task 2: Try out the customer facing agent

1. The agent understands the context it's in. Back in Salesforce, open the Contact record for Sofia Rodriguez. If you cannot see the Contacts tab switch to the **Coral Cloud** app using the App Launcher.
2. Open Agentforce by clicking the Astro Agentforce icon near the top.
3. Ask the agent to issue resort credits for Sofia:  
**Issue \$20 worth of credits to the guest**
4. Observe that Agentforce answers that credits have been issued. On the Related tab in the Credits related list you can see the credits have been issued. This is Agentforce being able to perform actions.



## Exercise Guide

### Unlock AI Capabilities in Your Custom Apps Workshop

#### Task 3: Asking the employee agent policy questions

The employee facing agent tries its best to answer the question asked but unless proper guardrails are in place it may hallucinate. Let us try it out.

1. In the Agentforce pane in Salesforce ask about the policy for guest room occupancy:  
**What is the policy regarding extra guests in the same room?**
2. Observe how Agentforce tries to answer the question - there is a problem though. The policies at Coral Cloud Resort (the ones we uploaded earlier) states that there are rules regarding extra occupancy in rooms. And the answer doesn't follow those policies.
3. Close the Agentforce pane.

Let us fix this by

1. Ensuring proper guardrails for the agent, and
2. Ensure the agent has access to the uploaded documents by creating an agent action using the prompt template we created earlier

#### Task 4: Add guardrails

1. Open **Setup**
2. Using Quick Search find and open **Agents**
3. To the right of the “Einstein Copilot” agent, click the dropdown arrow and click “Open in Builder”. This opens the agent in Agent Builder.
4. You cannot edit an active agent so click the **Deactivate** button to deactivate it (click OK when asked to confirm)
5. Click the **CustomerServiceAssistant** topic
6. Under **Instructions** click the **Add Instructions** button. In the field that appears add the following instruction:  
**Never answer questions about policy matters (valet, tours, bookings etc) unless you find specific information on that topic from an agent action**
7. Click the **Save** button at the bottom of the pane.
8. Now ask the same question in the **Conversation Preview** pane on the right (if the change is not picked up reload the agent in the upper right corner):  
**What is the policy regarding extra guests in the same room?**
9. Observe how the agent now refuses to answer the question as it doesn't have the information. But let's now give it access to the information it needs.

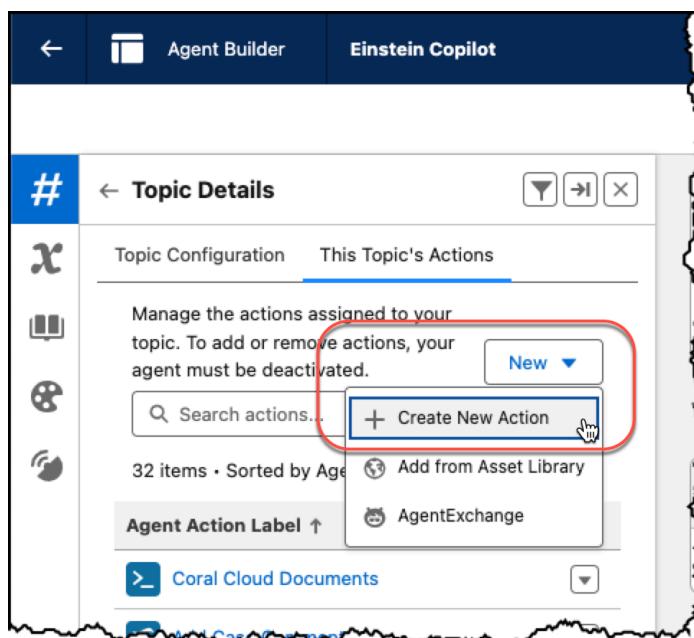
#### Task 5: Create an agent action

1. Click back into the **CustomerServiceAssistant** topic.
2. Click on the **This Topic's Actions** tab.

## Exercise Guide

### Unlock AI Capabilities in Your Custom Apps Workshop

3. Click on **New** and select **Create New Action**



4. Configure the action as follows:

Field	Value
Reference Action Type	Select <b>Prompt Template</b>
Reference Action	Select <b>Coral Cloud Documents</b>
Agent Action Label	[Keep default]
Agent Action API Name	[Keep default]

5. Click **Next**.
6. Uncheck the checkbox next to **Show loading text for this action**
7. Configure the **Inputs** as follows:

Field	Value
<b>Query</b>	
Instructions	This is the query to search for within the documents
Collect data from user	[Keep default - unchecked]
<b>Citation Mode</b>	
[Keep all defaults]	

## Exercise Guide

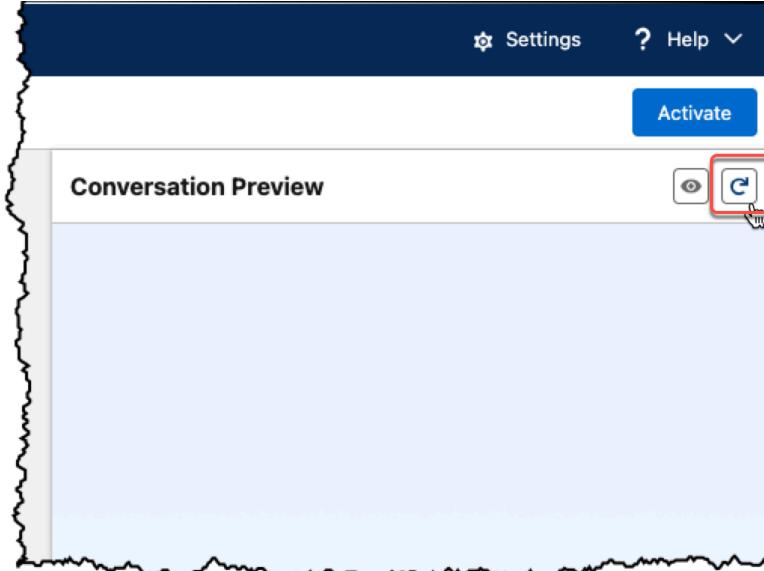
### Unlock AI Capabilities in Your Custom Apps Workshop

- Configure the **Outputs** as follows:

Field	Value
<b>Prompt Response</b>	
Instructions	[Keep default]
Filter from agent action	[Keep default]
Show in conversation	Checked
Output Rendering	[Keep default]
<b>Citations</b>	
[Keep all defaults]	

- Click **Finish**

- If you scroll through the list of actions you can see the action we just created marked with **New**
- Refresh the agent by clicking the **Refresh** action near the top right of Agent Builder



- Now ask the same question in the **Conversation Preview** pane on the right:  
**What is the policy regarding extra guests in the same room?**
- Observe how the agent now is able to answer.  
If the agent isn't able to answer, exit out of Agent Builder and reopen the agent in Agent Builder for the cache to be properly updated.
- Optional: Activate the agent and switch back to Salesforce and see how this now works for employees in Salesforce as well.

# **THANK YOU!**

# Additional Material

Additional training is available.

To learn more about Agentforce at your own pace go to

<https://developer.salesforce.com/agentforce-workshop/agents/overview>

To learn more about Prompt Builder at your own pace go to

<https://developer.salesforce.com/agentforce-workshop/prompt-builder/overview>

To learn more about Data Cloud at your own pace go to

<https://developer.salesforce.com/agentforce-workshop/data-cloud/overview>



# Introducing the new Developer Edition with Agentforce & Data Cloud

Get yours today at  
<https://developer.salesforce.com>

