Geeks for Geeks - Build your First Agent

In this hands-on exercise, you'll work with the Coral Cloud Resorts sample app. Coral Cloud Resort is a fictional beach resort that leverages data and artificial intelligence to provide highly personalized experiences to its guests. Your task is to create a customer service agent to assist guests in learning more about their available experiences. You will explore their pre-built capabilities powered by standard actions, and you can choose to extend them with custom actions built with Flow, Apex, and prompts. Get started by enabling Generative AI in your Org.

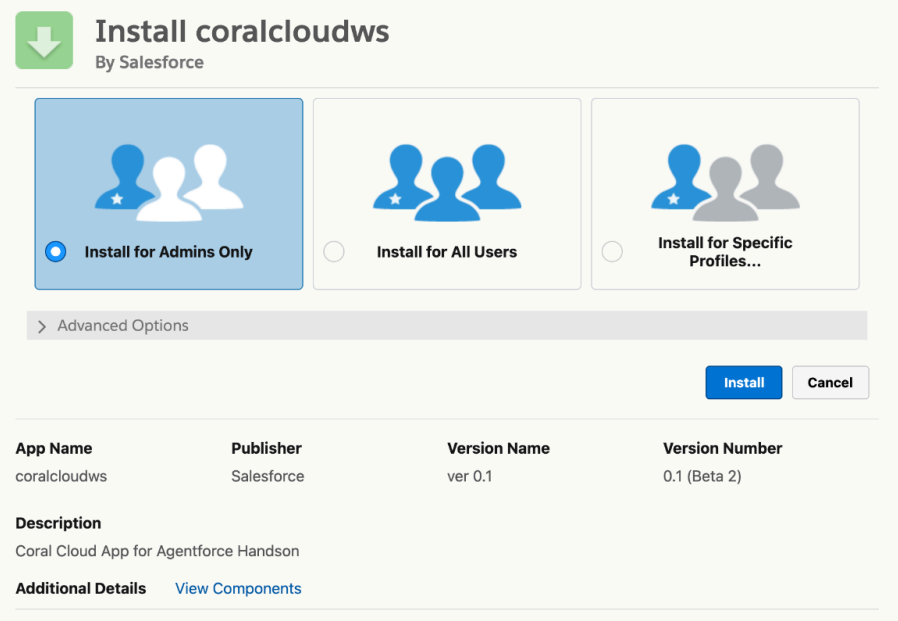
Before You Begin

Sign up for an org and install Coral Cloud App

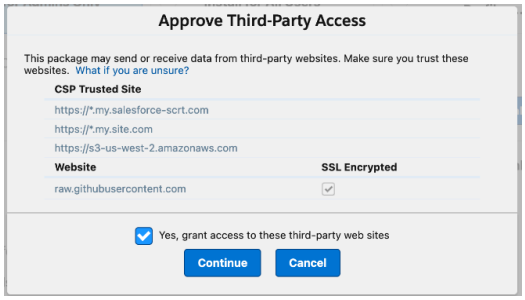
1. Sign up for an Agentforce enabled Developer Edition. Data Cloud is also enabled in this org ;)
2. Install the Coral Cloud App in your org.
    a. Once logged in into your org, replace the URL segment that comes after **lightning.force.com** with:

    `/packaging/installPackage.apexp?p0=04tgL00000006DVQAY`

    b. Press **Enter**.
    c. Select **Install for Admins Only** and click **Install**.



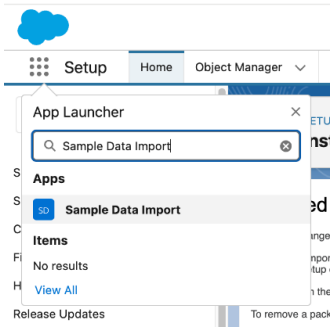
3. Approve Third Party access



4. When the package is installed (it should take less than a minute), click **Done**.

Import Sample Data

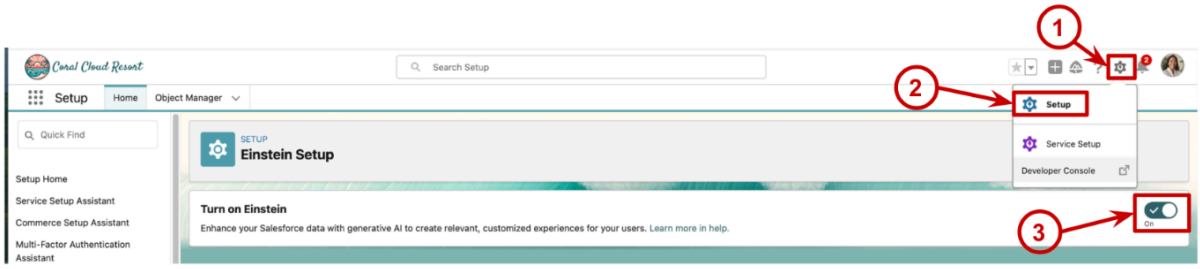
1. Click **App Launcher**, type `Sample Data Import` and select **Sample Data Import**.



2. Click **Import sample data**.

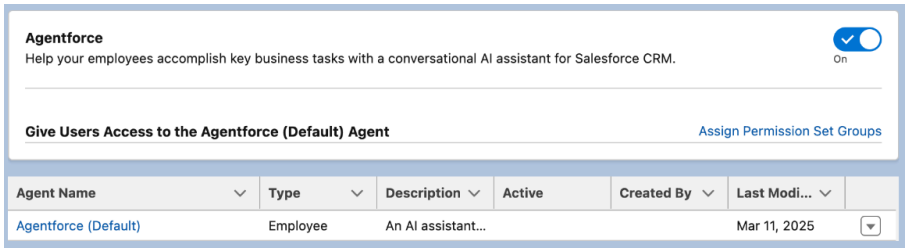
Step 1: Verify Generative AI is Enabled

1. Open **Setup** by clicking the **Gear** icon in the top-right corner.
2. In the **Quick Find** search bar, type **Generative AI**, then select **Einstein Setup**.
3. Ensure that the **Turn on Einstein** toggle is set to **On**.



Step 2: Verify Agentforce Agents is Enabled

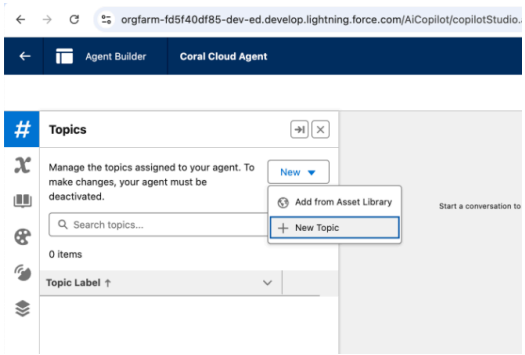
1. Open **Setup** by clicking the **Gear** icon in the top-right corner.
2. In the quick find, search for **Agents** and click **Agents** (under Agent Studio).
3. Ensure that the **Einstein Copilot for Salesforce** toggle is set to On.
4. You can see the default Agentforce Agent is available in your Org.



Step 3: Configure the Coral Cloud Agent

Configure the agent with topics and actions that it can use to support your customers.

1. In the quick find, search for **Agents** and click **Agents** (under Agent Studio).
2. On the Agents page in setup, click on the **Agentforce (Default)**.
3. Remove the default topics **General CRM** and **Single Record Summary** by clicking the dropdown next to each and selecting **Remove from Agent**. (Before removing the topics, make sure to click the Deactivate button on the top right of the page.
4. In Agent Builder, click on the **New** drop down and click on **New Topic**.



4. Configure the Topic as follows:

	Field	Value
1	Topic Label	`Customer Experience Support`
2	Classification Description	`This topic addresses customer inquiries and issues related to booking experiences at Coral Cloud Resort, including making reservations, modifying bookings, and answering queries about experience details.`
3	Scope	`The agent's job is to assist users in navigating and managing bookings for different experiences offered by Coral Cloud Resort, ensuring a seamless customer service experience by providing accurate information and resolving issues promptly.`
4	Instruction	`If a customer would like more information on Activities or Experiences, you should search for the related Experience__c records and summarize the output. Do not share information about available slots, Ids or Record Numbers.`

5. Click on **Next**
6. Do **not** select any actions.
7. Click **Finish**.

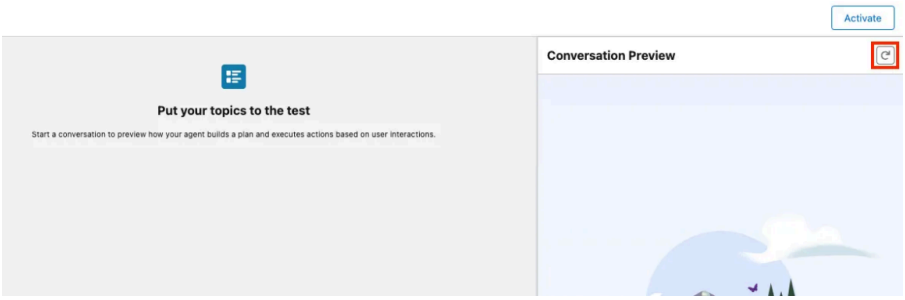
Step 4: Create custom service agent action

You can create custom actions for your agent using Flow, Apex, or prompts to access data in Salesforce. In this exercise, you will create a custom action to retrieve experience details using an existing flow called *Get Experience Details*.

1. In Agent Builder, select the **Customer Experience Support** topic.
2. Click on the **This Topic's Actions** tab in the topic details.
3. Click on the **New** drop down and click on **Create New Action**.
4. Configure the action as follows:

	Field	Value
1	Reference Action Type	`Flow`
2	Reference Action	`Get Experience Details`
3	Agent Action Label	Keep default
4	Agent Action API Name	Keep default

5. Click **Next**.
6. Uncheck **Show loading text for this action**.
7. Leave the default instructions in.
8. Check **Require input** for the `experienceName` input.
9. Check **Show in Conversation** for the `experienceDetails` output.
10. Click **Finish**.
11. Test out the instructions in the Conversation Preview. If prompted that you are about to use Einstein, click on **Got It**.
12. Click on the **refresh** button to reset the conversation.

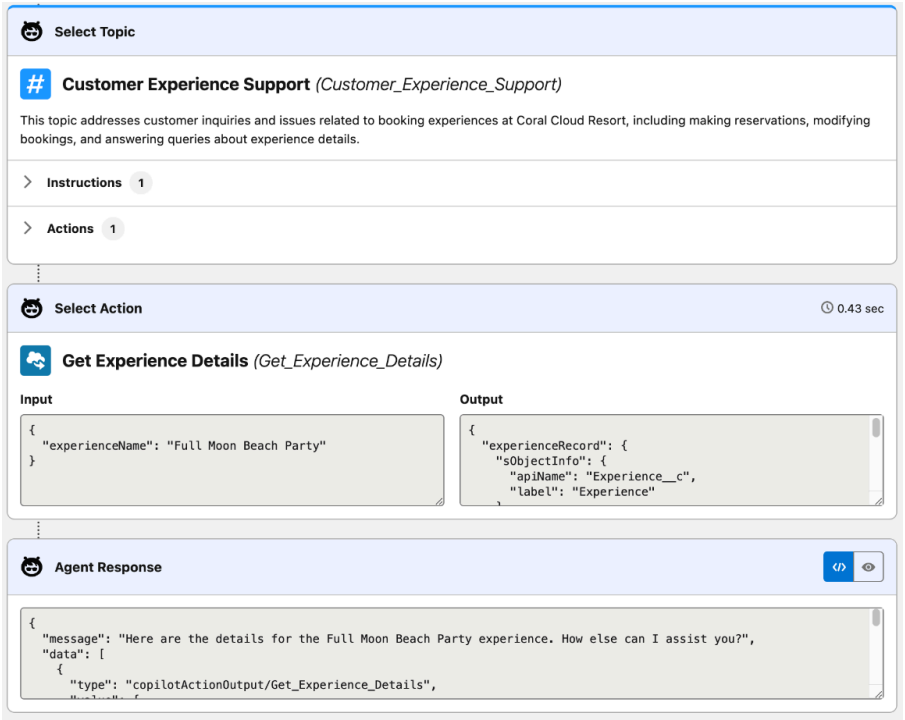


13. Enter this prompt in the dialog box:

```
Can you let me know more about the full moon beach party experience?
```

14. Press your Return/Enter key and notice the response, which gives information about that party.

In the planner, notice that the reasoning engine first selected the **Customer Experience Support** topic, then the **Get Experience Details** action you just created.



15. Click on **Activate** to activate the Agent
16. Click the Back button on the top left corner to exit the Agent Builder.

You have just configured a Service Agent.

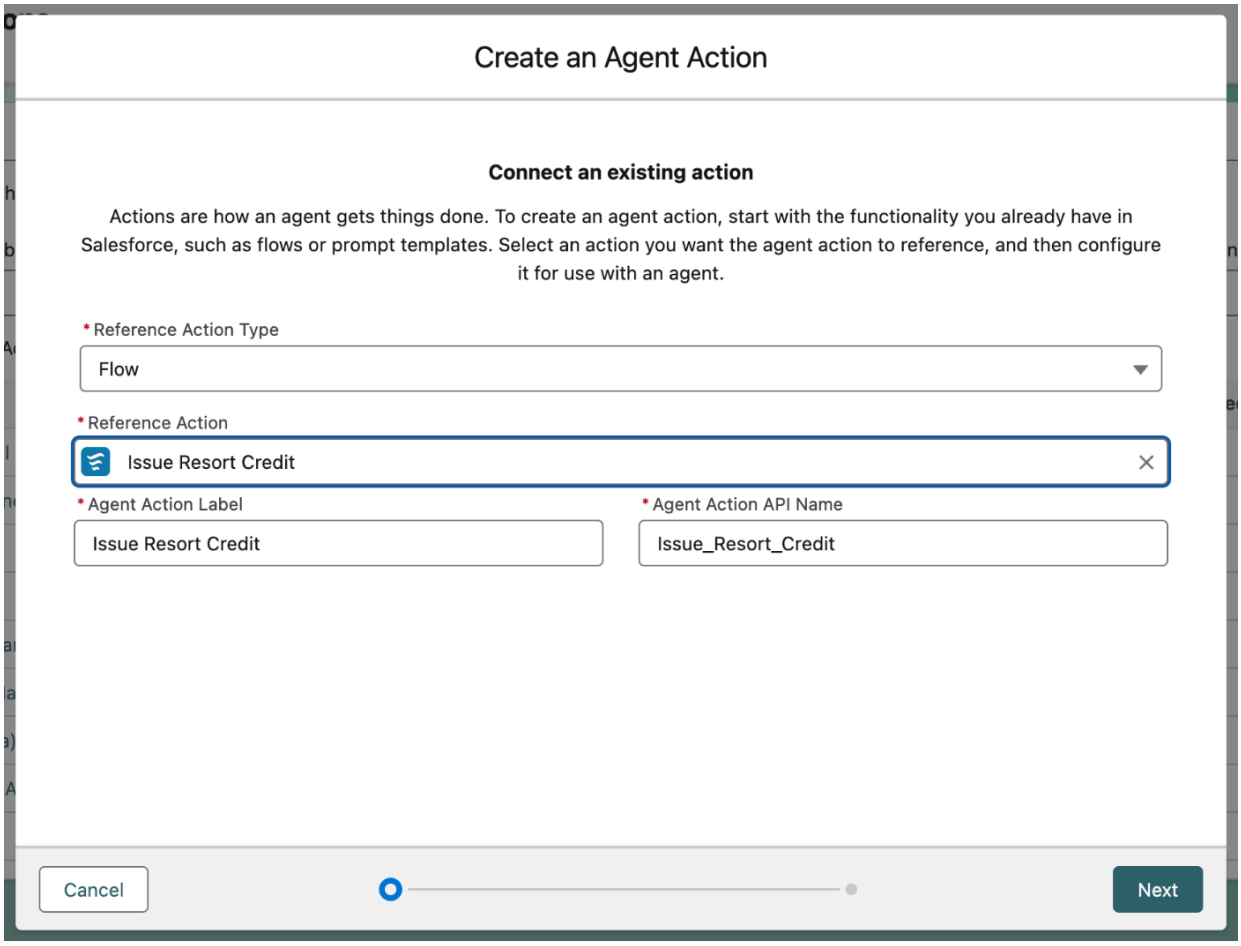
Extend Agent with Flow Action

Step 1: Create the Agent Custom Action

You can extend Agent with custom actions built with Flow, Apex, or prompts. In this exercise, you extend your agent with a custom action powered by another Flow. This custom action allows customer service representatives to issue resort credits to guests. You can use the flow, **Issue Resort Credit** which is installed in your org.

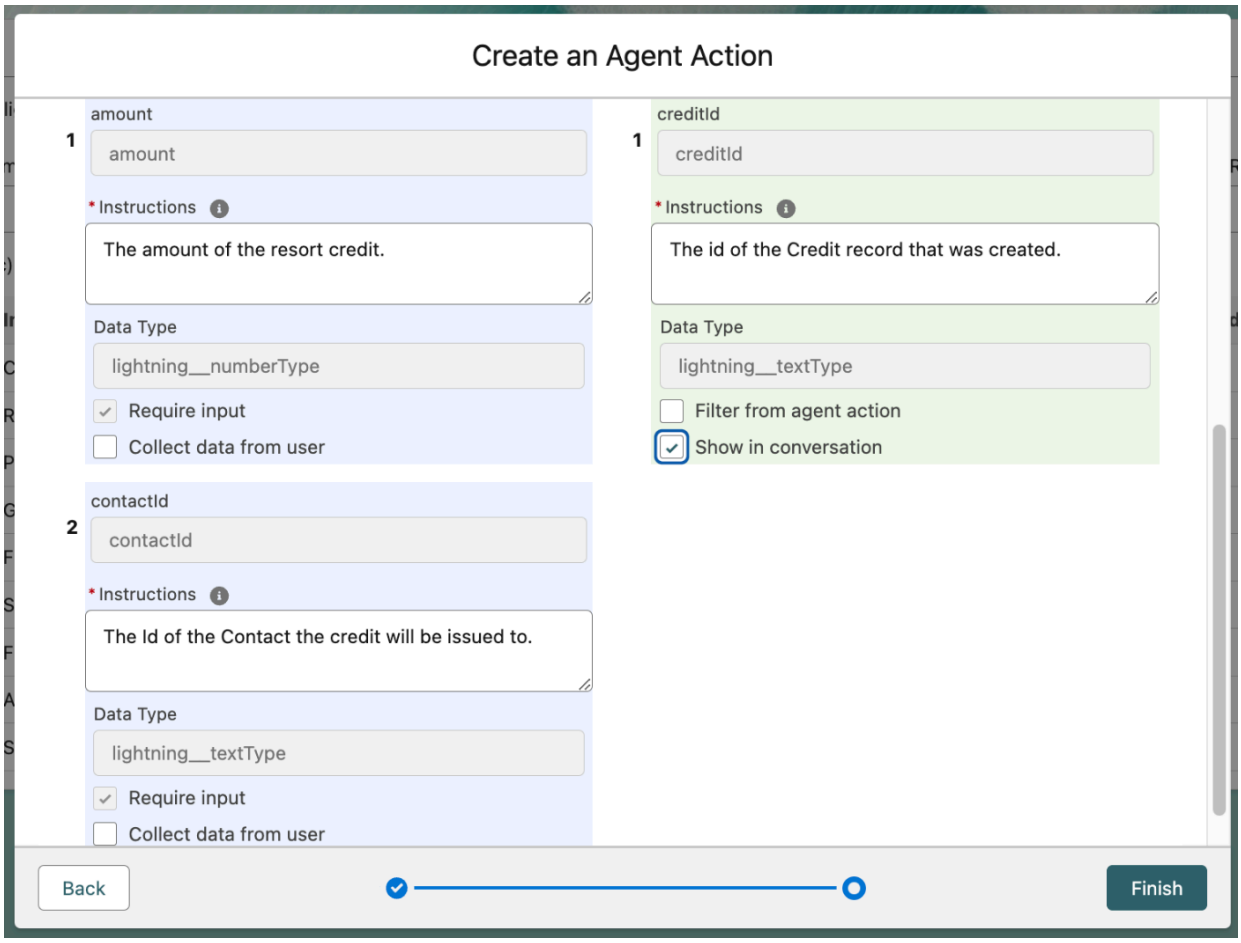
1. In the Setup Quick Find, search for and select **Agent Actions**(under Agent Studio).
2. Click **New Agent Action**.
3. Configure the action as follows:

	Field	Value
1	Reference Action Type	**Flow**
2	Reference Action	**Issue Resort Credit**
3	Agent Action Label	Keep default
4	Agent Action API Name	Keep default



5. Click **Next**.
6. Uncheck **Show loading text for this action**.
7. Leave the instructions with the default values.

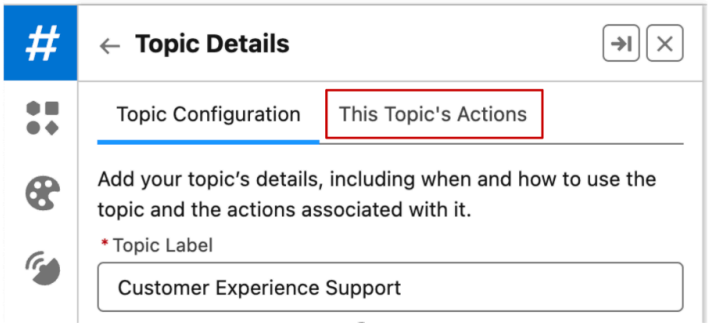
8. Check **Require Input** for both inputs (`amount` and `contactId`).
9. Check **Show in conversation** for the `creditId` output.



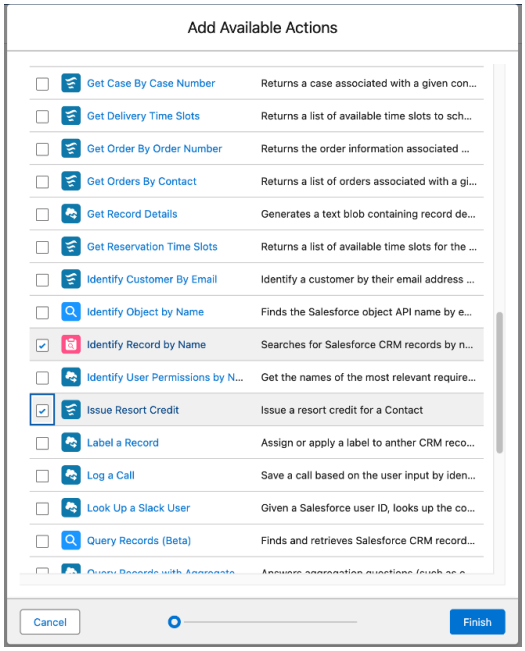
9. Click **Finish**.

Step 2: Add the Action to Your Agent

1. In the Setup Quick Find, search for and select **Agents**(under Agent Studio).
2. Click on the **Agentforce (Default)** in the list of agents at the bottom.
3. Click **Open in Builder**.
4. Click **Deactivate** to deactivate the agent so that you can add your new custom action.
5. In the **Topics** sidebar, click the **Customer Experience Support** topic.
6. Click the **This Topic's Actions** tab.



7. Click **New** button and click **Add from Asset Library**.
8. Check the **Identify Record by Name** and **Issue Resort Credit** actions, then click **Finish**.

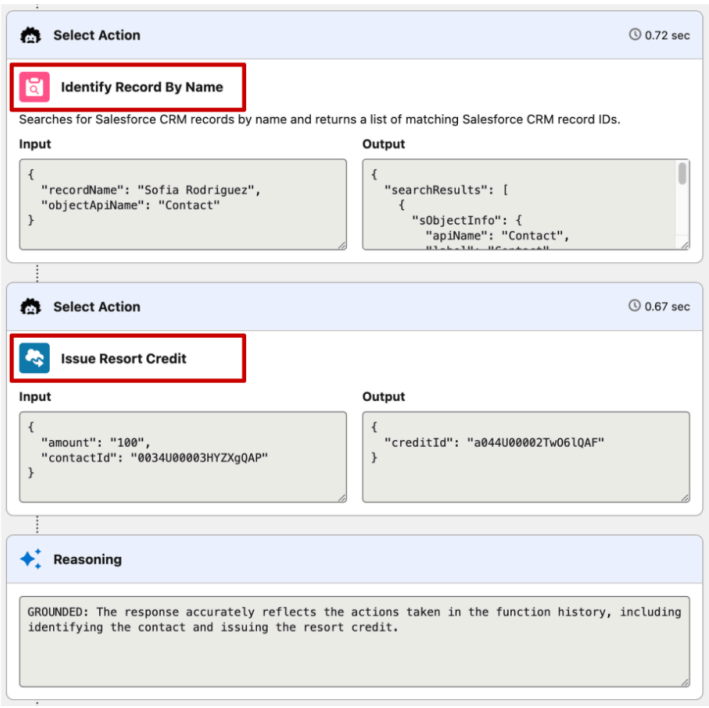


Step 3: Try it out

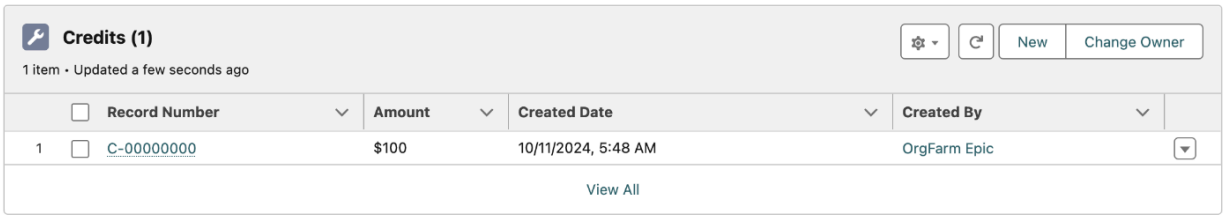
1. In the Conversation Preview panel, enter the following prompt:

```
Issue $100 resort credit to contact named Sofia Rodriguez
```

In the planner, notice that the reasoning engine first selected the **Identify Record By Name** action, then the **Issue Resort Credit** action you just created.



2. Launch Coral Cloud Resorts app using the App Launcher, navigate to the contact record for **Sofia Rodriguez**.
3. Click the **Related** tab.
4. Scroll down and verify that you see the resort credit you just issued using your agent. (**Note**: If Credits is not found in the Related tab, Add it to the Contact Page Layout from the Credit__c Object setup)



Extend Agent with Apex Action

Coral Cloud Resorts front-desk employees need an easy way to check the weather, so that they can recommend experiences based on the weather forecast. In this exercise, you'll create a custom action using an Invocable Apex class that allows your agent to invoke a third-party weather API.

Step 1: Create the Apex class

1. From Setup, search for **Apex** and click **Apex Classes**.

2. Click **New** button and copy-paste the following code to create **CheckWeather** Apex class.

```
public with sharing class CheckWeather {
    @InvocableMethod(
        label='Check Weather'
        description='Check weather at Coral Cloud Resorts at a specific date. The date
    )
    public static List<WeatherResponse> getWeather(
        List<WeatherRequest> requests
    ) {
        // Retrieve the date for which we want to check the weather
        Datetime dateToCheck = (Datetime) requests[0].dateToCheck;

        // Call a weather service to retrieve the weather through an API call
        WeatherService.Weather weather = WeatherService.getResortWeather(
            dateToCheck
        );

        // Create the response for the agent
        WeatherResponse response = new WeatherResponse();
        response.minTemperature = weather.minTemperatureC;
        response.maxTemperature = weather.maxTemperatureC;
        response.temperatureDescription =
            'Temperatures will be between ' +
            weather.minTemperatureC +
            '°C (' +
            weather.minTemperatureF +
            '°F) and ' +
            weather.maxTemperatureC +
            '°C (' +
            weather.maxTemperatureF +
            '°F) at Coral Cloud Resorts.';
        return new List<WeatherResponse>{ response };
    }

    public class WeatherRequest {
        @InvocableVariable(
            required=true
            description='Date for which we want to check the temperature. The variable
        )
        public Date dateToCheck;
    }

    public class WeatherResponse {
        @InvocableVariable(
            description='Minimum temperature in Celsius at Coral Cloud Resorts locatio
        )
        public Decimal minTemperature;
        @InvocableVariable(
            description='Maximum temperature in Celsius at Coral Cloud Resorts locatio
        )
        public Decimal maxTemperature;
        @InvocableVariable(
            description='Description of temperatures at Coral Cloud Resorts location f
        )
```

```
        public String temperatureDescription;
    }
}
```

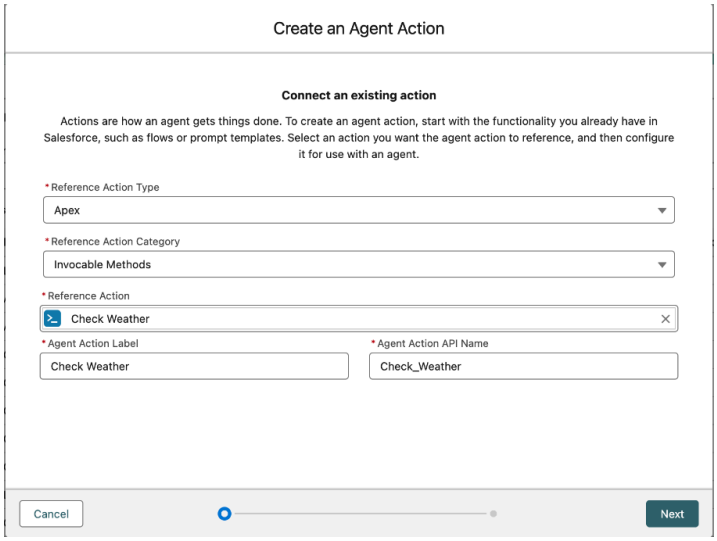
The `getWeather()` method is defined as an `@InvocableMethod` so it can be invoked by your agent. The method returns a `WeatherResponse` object. The attributes of the `WeatherResponse` class are defined as `@InvocableVariable`. The descriptions in both `@InvocableMethod` and `@InvocableVariable` are important because they allow your agent to understand how to use the action.

3. Click **Save**.

Step 2: Create the agent custom action

1. From Setup, open **Agents Actions**.
2. Click **New Agent Action**.
3. Configure the action as follows:

	Field	Value
1	Reference Action Type	`Apex`
2	Reference Action Category	`Invocable Methods`
3	Reference Action	`Check Weather`
4	Agent Action Label	Keep default
5	Agent Action API Name	Keep default

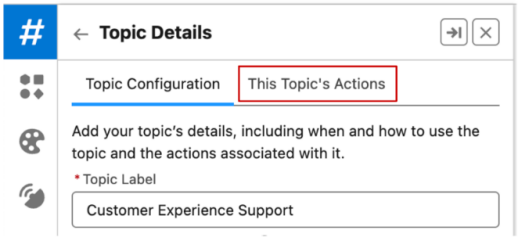


4. Click **Next**.
5. Uncheck **Show loading text for this action**.
6. Notice that the instructions fields are pre-filled based on the descriptions provided in the Apex code.
7. Check **Collect data from user** for the input (`dateToCheck`).

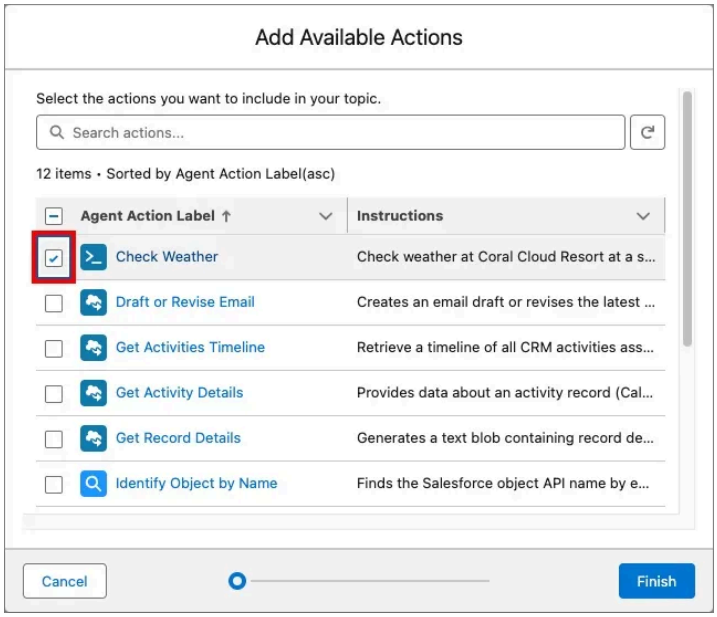
8. Check **Show in conversation** for the `temperatureDescription` output.
9. Click **Finish**

Step 3: Add the Action to Your Agent

1. In the Setup Quick Find, search for and select **Agents**(under Agent Studio).
2. Click on the **Agentforce (Default)** in the list of agents at the bottom.
3. Click **Open in Builder**.
4. Click **Deactivate** to deactivate the agent so that you can add your new custom action.
5. In the **Topics** sidebar, click the **Customer Experience Support** topic.
6. Click the **This Topic's Actions** tab.



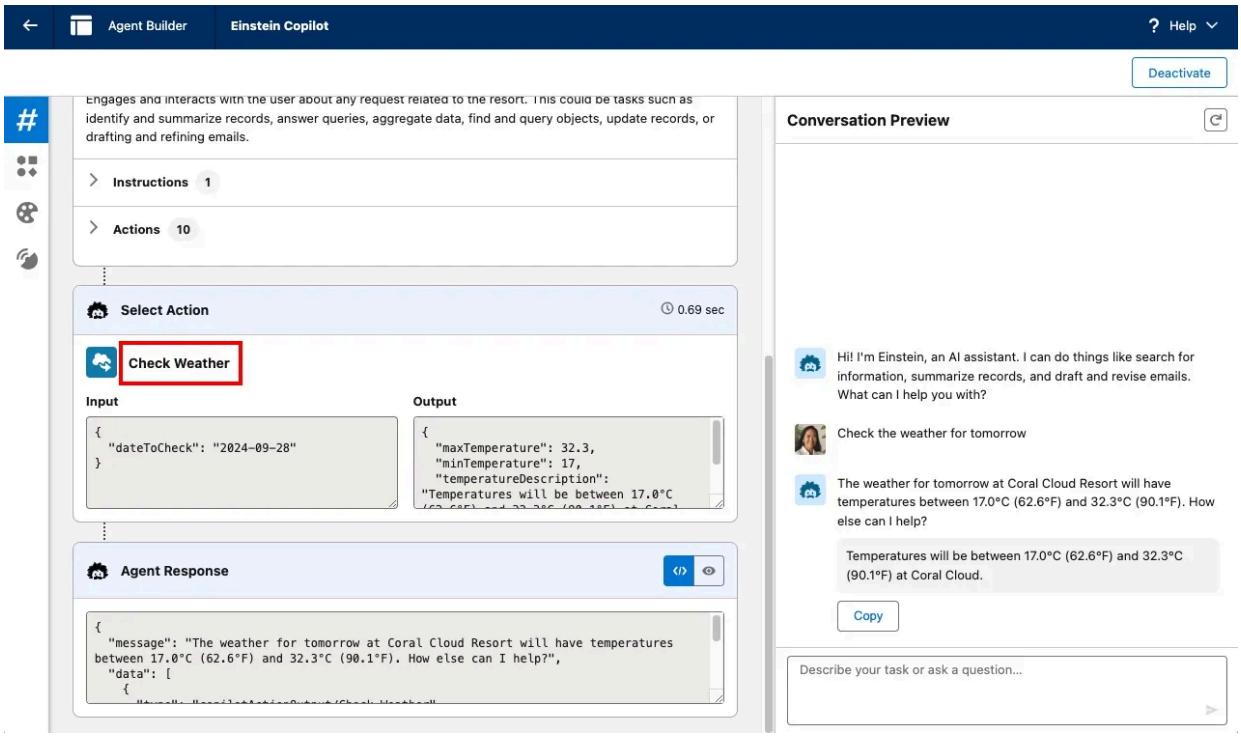
7. Click **New** button and click **Add from Asset Library**.
8. Check the **Check Weather** action and click **Finish**.



Step 4: Try it out

1. In the Conversation Preview panel, enter the following prompt:

```
Check the weather for tomorrow
```



   Examine the planner and note that the reasoning engine selected the **Check Weather** action to fullfill the request.

2. Click the **back arrow button** to go back to setup.

That's how easy it is to extend agents with custom actions powered by Apex.

Create Your Own Agent

Create a new Agent

The default Agentforce Agent is available for you to distribute to your internal Salesforce users. You have just configured and extended the Agent with flow and apex actions. You can also create custom Agents that you can use just about anywhere else you want - Slack, Website, etc.. Adding a new agent is as simple as clicking a button and adding details.

1. In the Setup Quick Find, search for and select **Agents**.
2. Click **+New Agent**.
3. Click **Agentforce Service Agent** to select it and click **Next**.
4. Deselect all of the pre-created topics and click **Next**.
5. Set field values as follows:

	Field	Value
1	Topic Label	`Coral Cloud Service Agent`
2	Description	`This is the Coral Cloud Agent that helps customers learn more about Experiences as well as book sessions.`
3	Role	`The agent's job is to assist users in navigating and managing bookings for different experiences offered by Coral Cloud Resorts, ensuring a seamless customer service experience by providing accurate information and resolving issues promptly.`
4	Company	`Coral Cloud Resorts is a fictitious seaside resort that manages guests and their reservations. It offers a rich set of experiences.`
5	Agent user	New Agent User
6	Enrich event logs with conversation data	TRUE

6. Click **Next**.
7. Click **Create**.

`TIP`
`Each agent has a designated running user and will have access to all of the data and metadata that that user can see. Ensure that you start with a minimum access profile and only give the agent access to the data that it needs.`

Agent Name	Type	Description	Active	Created By	Last Modified	
Agentforce (Default)	Employee	An AI assistant f...			Mar 11, 2025	
Coral Cloud Service Agent	Service Agent	This is the Coral ...			Mar 11, 2025	

Optional: Add Get `Experience Details action` to your new Agent.

You have just created a custom Service Agent that can now be deployed anywhere you want - Slack, Website, etc. You can extend this agent by creating topics and actions. You can create actions by using flows, apex, or prompt templates. You can learn more by exploring the below resources.

Resources

Trailhead: Build an AI Agent with Agentforce