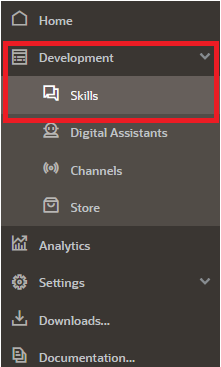
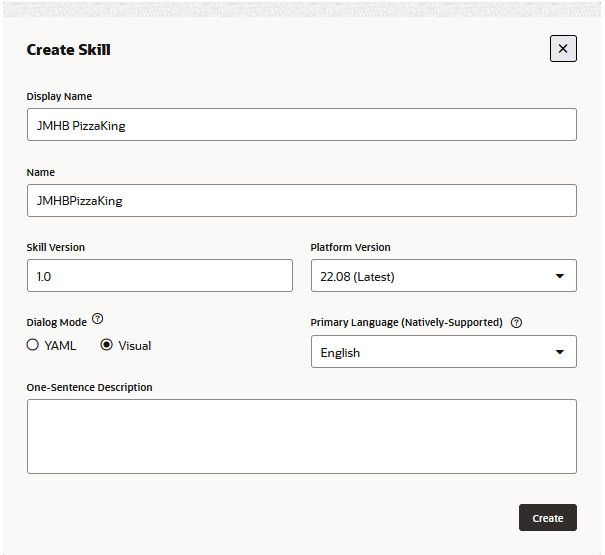
**Oracle Digital Assistant**

**Create a Skill**

1. With the Oracle Digital Assistant UI open in your browser, click to open the side menu.
2. Click **Development** and then select **Skills**.



1. Click the **+ New Skill** button.
2. For **Display Name** enter Pizza King. If you are working in an environment where others may also be creating the same tutorial, prefix Pizza King with your unique initials.
3. Optionally, fill in a one-sentence description (e.g., Skill for ordering from Pizza King)
4. For the other fields, leave the default values. Note that the Dialog Mode is **Visual**. Later on in this tutorial, you're going to create a dialog flow, the definition of the skill-user conversation, using the declarative approach of the Visual Flow Designer.



1. Click **Create**. The designer will then open on the Intents page.

**Create Intents**

**Create the Order Pizza Intent**

1. Click the **+ Intent** button.
2. Next to the **Conversation Name** field, click edit icon, and enter Order Pizza.
3. In the **Name** field, type OrderPizza.
4. Select and copy all of the example sentences below to your clipboard:
   * Would you happen to have thin crust options on your Pizzas?
   * Let's order a cheese pizza
   * Would love a large Pepperoni please!
   * I feel like eating some pizza
   * I would like to order a pizza
   * Can I order a Pizza?
   * What's on the menu today?
   * I want pizza
   * Do you server gluten-free pizza?
   * I want to order pizza for lunch
   * Do you have deep dish pizzas available?
   * Order Pizza!

(You'll notice that it's fine for utterances to have inconsistent punctuation and capitalization.)

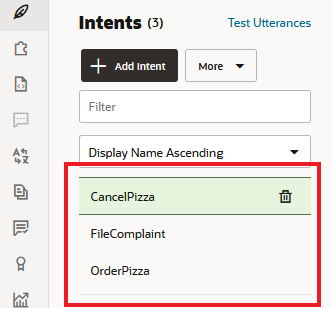
1. In the Examples section, click **Advanced input mode**.
2. In the **Utterances to Add** field, paste the examples. Then click **Create**.

**Create the Cancel Pizza Intent**

1. Click the **+ Add Intent** button (located at the top left of the page).
2. Next to the **Conversation Name** field, click edit icon, and enter Cancel Pizza.
3. In the **Name** field, type CancelPizza.
4. Click edit icon in the Answer field. Enter I am sorry to hear this. Let me take your cancellation request.
5. Select and copy all of the example sentences below to your clipboard:
   * Can i cancel my order?
   * Cancel my order
   * Cancel my Pizza please
   * How do I cancel my order?
   * I don't want my Pizza anymore
   * I really don't want the Pizza anymore
   * I'd like to cancel my order please
   * It’s been more than 20 mts. Please cancel my order and issue a refund to my card.
   * Need to cancel my order
   * Please cancel my pizza order
   * Please don't deliver my Pizza
6. In the **Utterances to Add** field, select **Advanced input mode** (if needed), then paste the examples.
7. Click **Create**.

**Create the File Complaint Intent**

1. Click the **+ Add Intent** button.
2. Next to the **Conversation Name** field, click edit icon, and enter File Complaint.
3. In the **Name** field, enter FileComplaint.
4. Click edit icon in the Answer field. Enter I am sorry to hear this. Let me take your complaint details.
5. Select and copy all of the example phrases below to your clipboard:
   * I am upset
   * You charged me wrong
   * I want to file a complaint
   * I am not happy with my recent order
   * I have some grief to share
   * I want to speak with a manager
   * Can I raise a complaint
6. In the **Utterances to Add** field, paste the examples and then click **Create**.



**Train Your Intents**

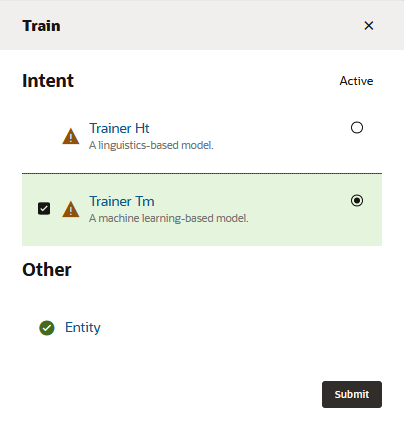
You've now provided the basic ingredients that allow the skill to recognize user input for ordering a pizza, but the skill has no cognition. It can't understand any user input.

To enable the skill to interpret user input based on the utterances that you just added, you need to train to build the intent model.

1. On the right side of the page, locate and click the **Train**.

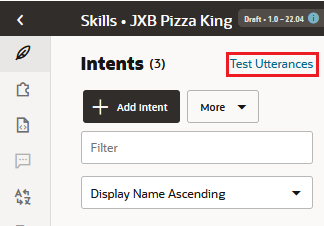


1. Select **Trainer Tm** and then click **Submit**.

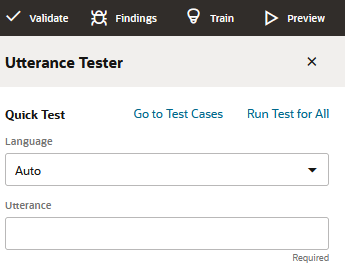


**Test Your Model**

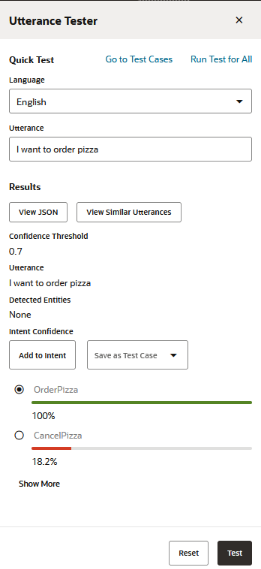
1. Click the **Test Utterances** link that appears near the top of the Intents page.



1. The Utterance Tester slides out from the right side of the page with its Quick Test page open.



1. In the **Language** field of the dialog, select **English**.
2. In the **Utterance** field, type I want to order pizza. Then click the **Test** button (located at the bottom right of the tester) and look at the test results.



1. Click **Reset** (located at the bottom right of the tester).
2. Now scroll back to the top of the dialog, enter I feel like eating some pizza in the **Utterance** field, and then click **Test**.

This should also resolve to the OrderPizza intent. Click **Reset**.

1. Now try Cancel my order.

This should resolve to the CancelPizza intent. Click **Reset**.

1. And now try Dude, bring me pizza and see what that resolves to.
2. Type You are expensive and you still don't deliver on time in the **Utterance** field and click **Test**.
3. To remedy this, you're going to add this utterance to the FileComplaint intent:
   1. Click **Show More** to access the FileComplaint result.
   2. Select the **FileComplaint** radio button.
   3. Click **Add to Intent**.

Retrain the skill with **Trainer Tm**.

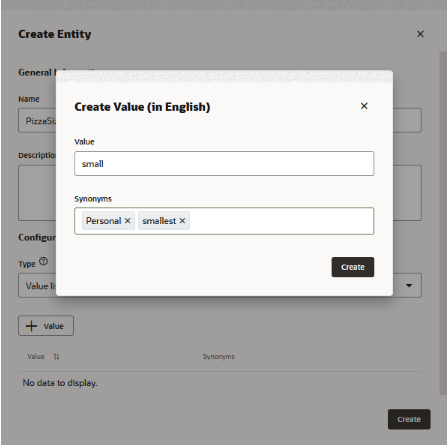
1. If you find one that doesn’t match well, select the intent that it should be resolved to and click **Add to Intent**.
2. Train the model again and then re-test.

## Create Entities

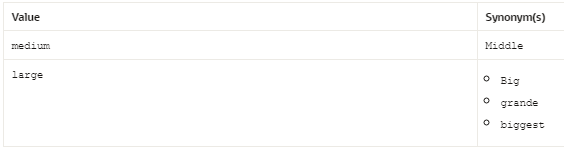
Now it's time to add entities, which detect information in the user input that can help the intent fulfill a user request. For the Pizza King business, such information could be pizza size, pizza toppings, and delivery time.

**Create Entities for Pizza Size and Pizza Topping**

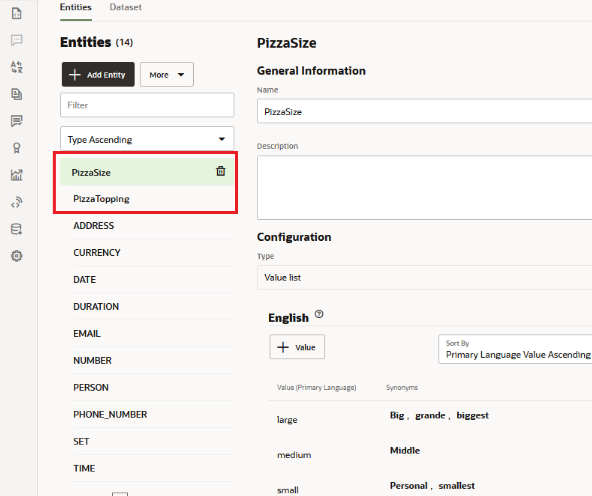
1. In the left navigation for the designer, select puzzle sign.
2. Click **+ Add Entity** to create a new entity.
3. In the **Name** field, change the value to PizzaSize.
4. In the Configuration section, select **Value list** from the **Type** menu.
5. Click **+ Value**.
6. For **Value**, type small.
7. For **Synonyms**, type Personal, then click Enter. Type smallest, and then click Enter again.
8. Click **Create**.



1. Following the procedure described in the previous four steps, you are going to add two more values to the PizzaSize entity: one for medium-sized pizza orders, and one for large-sized pizza orders.



1. Click **Create** to complete the entity.
2. Click **+ Add Entity** to create another entity.
3. In the **Name** field, change the value to PizzaTopping.
4. In the Configuration section, select **Value List** from the **Type** menu
5. Add separate values for Meaty, Veggie, Hot and Spicy, and American Hot.
6. Click **Create**.



### Edit Prompts for the Entities

Now let's enter some text to prompt users to enter values that can be matched by these entities.

1. In the list of entities, select **PizzaSize**.
2. Scroll down to the Prompts section.
3. In the row for the default prompt ("Please enter PizzaSize"), click edit icon, replace the text with Which size would you like for your pizza? Then click Enter.
4. In the list of entities, select **PizzaTopping**.
5. Scroll down to the Prompts section.
6. In the row for the default prompt ("Please enter PizzaTopping"), click edit icon, replace the text with What type of pizza would you like?

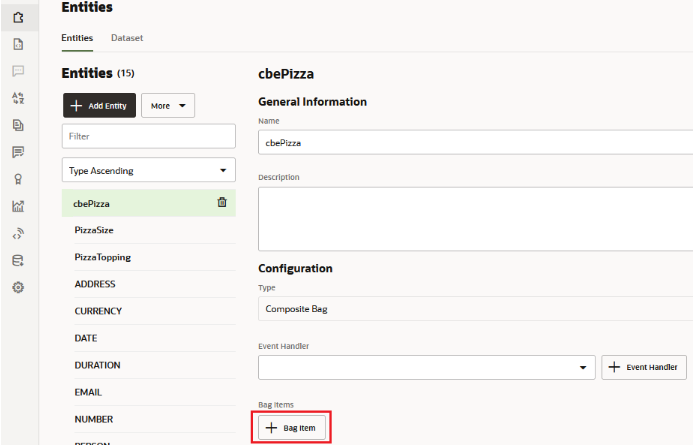
**Create a Composite Bag Entity**

In this step, you're going to simplify your development efforts using a composite bag entity, which enables you to manage the three entities that you just created as a consolidated entity.

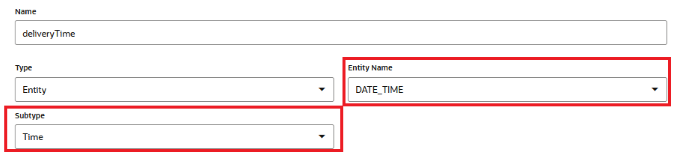
1. Click **+ Add Entity**.
2. In the **Name** field, enter cbePizza.
3. In the Configuration section, select **Composite Bag** from the **Type** menu.

**Add Entities to the Composite Bag**

1. Click **cbePizza** in the entity list to access its editing page.
2. To add the entities that are managed by the composite bag, you need to create references for them called bag items. Click **+ Bag Item**.



1. In the Add Bag Item dialog, replace *BagItem1* in the Name field with pizzaSize.
2. Select **Entity** as the type.
3. Choose **PizzaSize** from the Entity Name list.
4. Click **Close** (located at the upper right).
5. Repeat these steps to create a bag item for the PizzaTopping entity:
   1. Name the item pizzaTopping.
   2. Choose **PizzaTopping** from the Entity Name list.
6. In addition to enabling users to select the pizza type and size, the skill will also enable them to enter a delivery time. For this, you're going add another entity item to the bag. Unlike the PizzaSize and PizzaType entities that you created from scratch, you're instead going to use the built-in TIME entity (or if your skill is Version 22.08 or higher, the DATE\_TIME entity), which recognizes various time formats in the user input. To create this item:
   1. Click **+Bag Item**.
   2. Enter deliveryTime in the Name field.
   3. Select **Entity** from the Type list.
   4. Choose **TIME** from the Entity Name List. If your skill is Version 22.08 or higher, choose **DATE\_TIME** from the Entity Name List. Then choose **TIME** from the Subtype menu.



* 1. Scroll down the page, then click **+ Prompt**.
  2. Type when can we deliver that for you (e.g., 4pm)? and then click Enter.

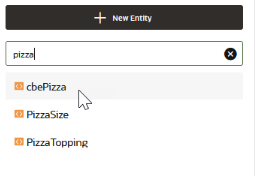


1. Click **Train**.
2. Select **Entity**, and then click **Submit**.

**Associate the Entities with OrderPizza Intent**

For an entity to be recognized when parsing the user input message, it needs to be associated with the appropriate intents. In this case, we need to associate the composite bag entity with the OrderPizza intent.

1. In the left navigation for the designer, select leaf sign.
2. Select the **OrderPizza** intent.
3. Click the **Add Entity** dropdown (in the upper right side of the page), type pizza in the **Filter Field**, and select **cbePizza**.

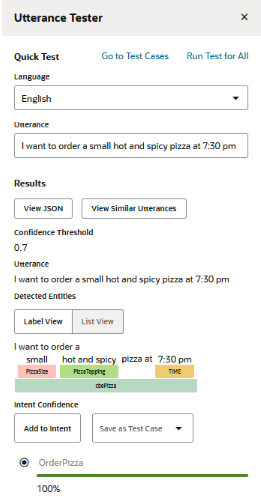


1. Retrain the model with Trainer Tm.

**Test the Entities**

The Utterance Tester feature enables you to test whether the skill identifies entity values in user input.

1. In the left navigation for the designer, select leaf sign.
2. Click the **Test Utterances** link. Click **Reset** to clear the input field.
3. In the **Utterance** field of the dialog, type I want to order a small hot and spicy pizza at 7:30 pm and then click **Test**.



1. Now let's try another utterance. In the **Utterance** field, now type I want to order the biggest meaty pizza at noon and click **Test**.

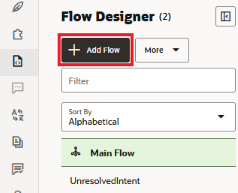
The result should look like what is shown below. The PizzaSize label notes that the "biggest" synonym is matched for the PizzaSize entity. The Time label notes that "noon" is a recognized value.

**Design the Dialog Flow**

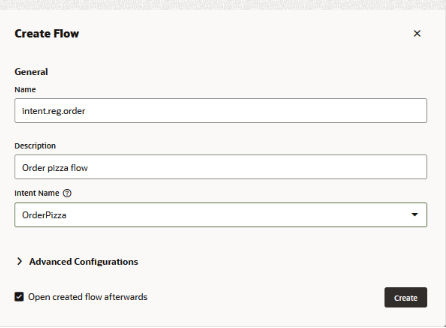
**Create the Order Pizza Flow**

Now that you have the OrderPizza intent and the pizza variable, you're ready to start building the dialog flow.

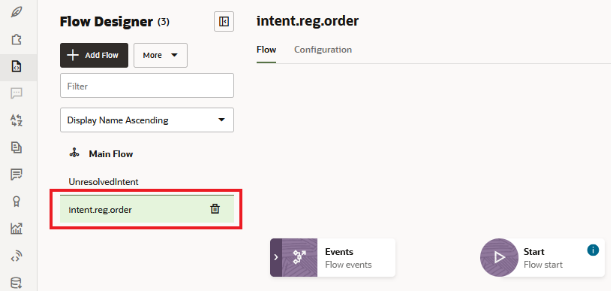
1. Click **Flows** in the left navbar.
2. Click **Add Flow**.



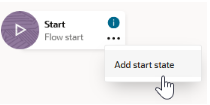
1. In the Create Flow dialog, enter intent.reg.order as the name.
2. For the description, enter Order pizza flow.
3. Select **OrderPizza** from the Intent Name list to create a mapping between the OrderPizza intent and this flow. This mapping instructs the Main Flow to execute this flow when user input resolves to the OrderPizza intent. You can also set intent event mappings like this one within the Main Flow.



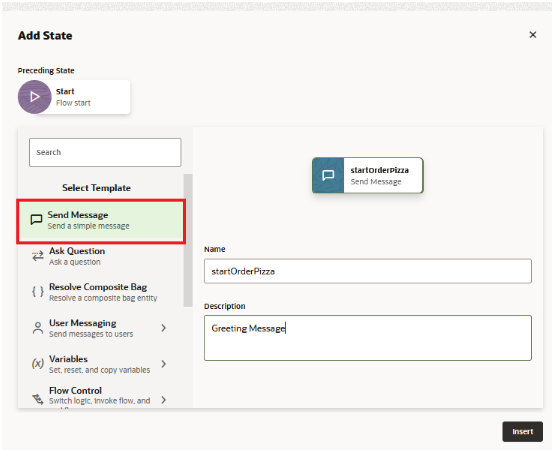
1. Click **Create**.
2. Click **intent.reg.order** in the flow list.



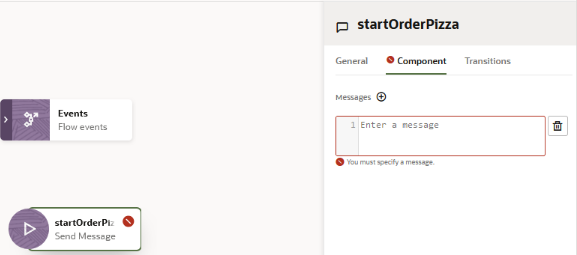
1. In the dialog flow editor, hover over the Start node to display its menu . Click the menu then click **Add start state**.



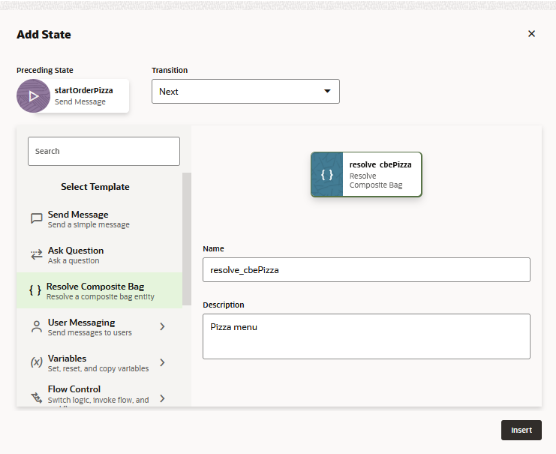
1. In the Add State dialog, choose **Send Message**, then enter startOrderPizza as the name and Greeting Message as the description. Then click **Insert**.



1. When you're done, the startOrderPizza state should appear in the canvas with its property editor open to the Component page.



1. Paste the following into the Messages field: “Ok, let's get that order sorted.”
2. Hover over the startOrderPizza state to access the menu. Then click **Add State**.
3. In the Add State dialog, choose **Resolve Composite Bag**.
4. Name the state resolve\_cbePizza.
5. Enter Pizza menu as then description. Then click **Insert**.



1. Like the startOrderPizza state, the resove\_cbePizza state is flagged with an error because of an undefined property. For this state, the Component page tells you that you can fix this error by adding a variable reference a composite bag entity, which in this case, is the cbePizza entity. To create this variable, click **Create**.
2. Because only the intent.reg.order flow uses this variable, choose **Flow Scope Variable**. The value held by this variable will persist only until the flow executes. If the other flows in this skill needed this variable, then you would create a skill scope variable instead.
3. For the name, enter pizza.
4. For the description, enter Resolves the pizza order.
5. Select **Entity** as the variable type, then select **cbePizza** as the entity name.
6. Click **Apply**.
7. Hover over the resolvePizza state to access the menu, then click **Add state**.
8. Select **Send Message**.
9. Name the state displayOrderConfirmation.
10. Enter Confirmation message as the description. Then click **Insert**

In the Component page of the property inspector, enter the following confirmation message. It uses Apache FreeMarker expressions to access the values set for the pizza variable:

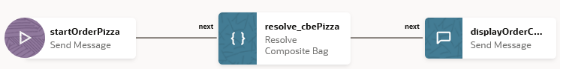
* For Versions 22.04 and 22.06:

Your ${pizza.value.pizzaSize.value} ${pizza.value.pizzaTopping.value} pizza will be delivered at ${pizza.value.deliveryTime.date?long?number\_to\_time?string('HH:mm')}

* For Version 22.08 and higher:

Your ${pizza.value.pizzaSize.value} ${pizza.value.pizzaTopping.value} pizza will be delivered at ${pizza.value.deliveryTime.value?time.xs?string['hh:mm a']}.

When you're done, the flow should look like this:



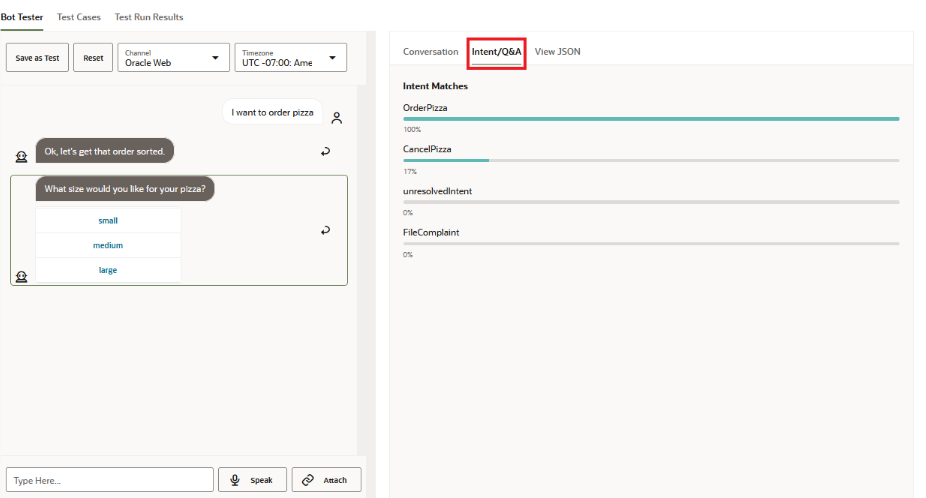
## Test Your Skill

In this section, you'll test out the intent.reg.order flow, but also test out the conversation flows that are handled automatically: the FileComplaint and CancelPizza answer intents and unresolvedIntent.

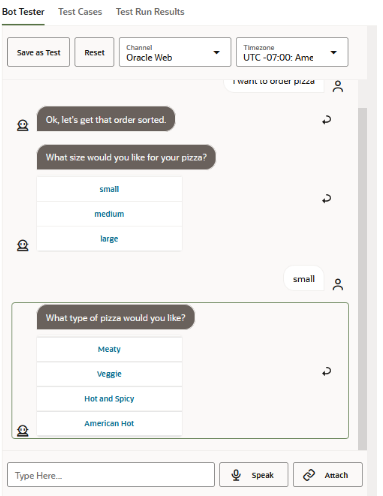
**Test the Order Pizza Flow**

1. Open the skill tester by clicking Preview at the top of the page.
2. In the **Utterance** field, type I want to order pizza and then press Enter.

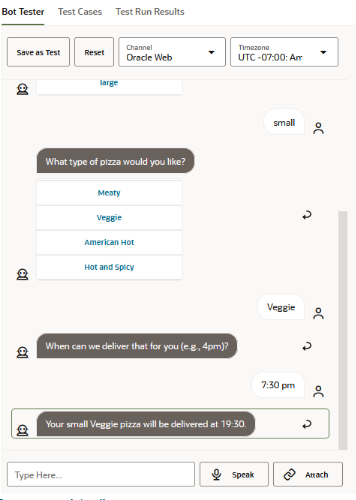
You should see a menu of pizza sizes:



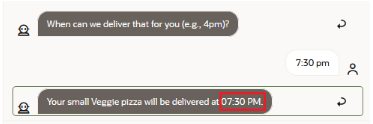
1. Click the **Intent/Q&A** view to see how the utterance resolves to the OrderPizza intent.
2. Back in the tester, select an option (e.g., **small**) from the menu.



1. A prompt for the pizza topping should appear. Continue the conversation by selecting a topping (e.g., **Veggie**).
2. A prompt for the delivery time should appear. Enter a delivery time, such as 7:30 p.m.



You should receive an order confirmation that summarizes the order. In the Conversation Tester, note the dialog flow traversal that routes the request from the Main Flow to the intent.reg.order flow.



1. Click **Reset**.
2. Finally, enter I want to order a veggie pizza at 8:00pm.

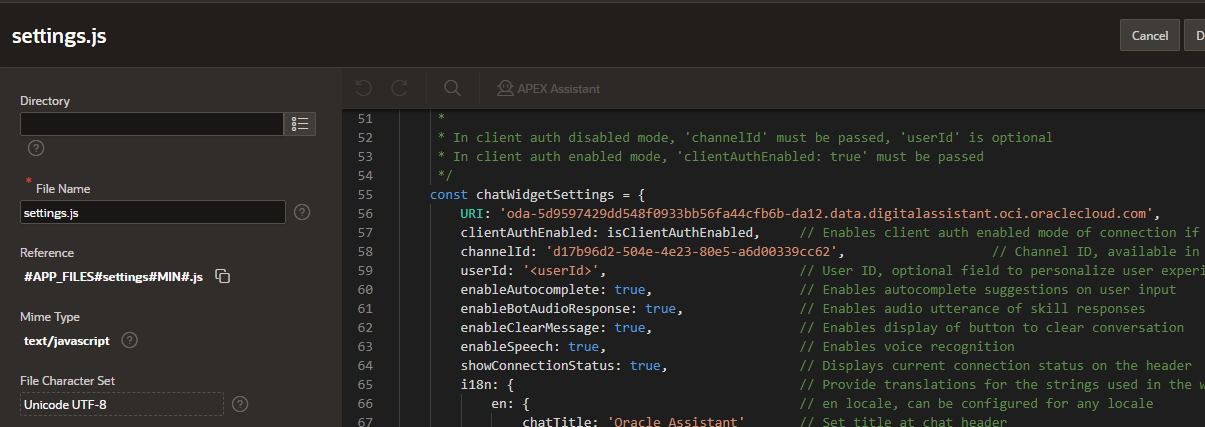
This time the topping menu and the delivery time should be skipped, but the pizza size list should be displayed.

1. Select a pizza size. Your order should then be completed.

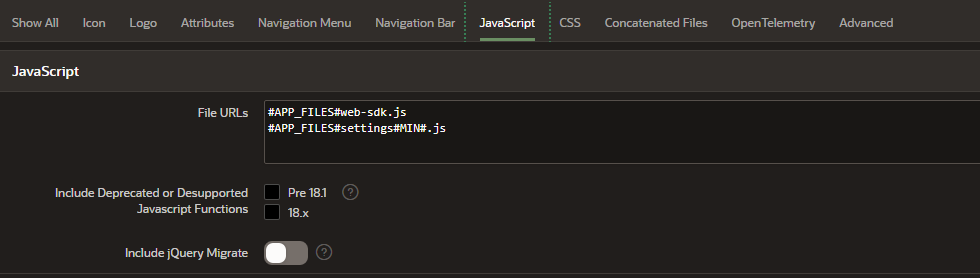
# Integrating an Oracle Digital Assistant Chatbot in to Oracle APEX

**Configure settings.js:**

1. Before we begin, ensure that you've successfully downloaded the Oracle Web SDK. If you haven't done so yet, you can find the download link on this [**page**](https://www.oracle.com/downloads/cloud/amce-downloads.html).
2. Open the settings.js file and change the URI and Channel ID under initSdk(name) -> chatWidgetSettings.
3. Replace the <URI> with the hostname of the Oracle Digital Assistant instance, and replace <Channel ID> with the channel ID of the Web Channel to which you want to connect.

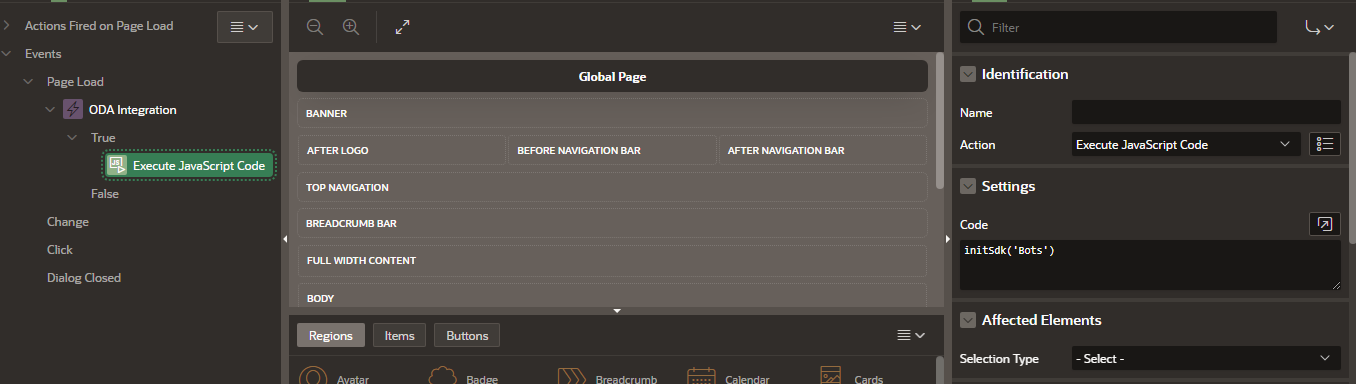


1. Navigate to the**Shared Components** of your APEX application. From there, proceed to User Interface Attributes. Within this section, focus on the "JavaScript" subsection. Here, paste the Reference URLs for both settings.js and web-sdk.js to enable file access.



**Configuring Oracle Digital Assistant Integration:**

1. In your Oracle APEX application, head to "Page Zero" to kick off the customization process. Then, navigate to the "Dynamic Action" tab within the rendering section.
2. Create a dynamic action to trigger on page load, and within the "True" section, choose the action type "Execute JavaScript Code."
3. Under settings, In the code section, enter initSdk('Bots') to initiate the necessary setup for your Oracle Digital Assistant integration.



**Reference Links:**

* <https://docs.oracle.com/en/cloud/paas/digital-assistant/tutorial-skill/index.html#before_you_begin>
* <https://blogs.oracle.com/apex/post/integrating-an-oracle-digital-assistant-chatbot-in-to-your-oracle-apex-app>