

DUMPSBOSS.

Oracle Cloud Platform Digital Assistant 2022 Professional

Oracle 1z0-1071-22

Version Demo

Total Demo Questions: 10

Total Premium Questions: 79

Buy Premium PDF

<https://dumpsboss.co>

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

QUESTION NO: 1

You have a skill for ordering pizzas. In your experience, you have found that 95% of your customers want a regular crust and just 5% prefer gluten-free crust. Because so few people are likely to want a gluten-free crust, you don't want to prompt for that option, but you do want to make it available.

Assuming you already have a composite bag entity that contains a PizzaCrust entity that has the values "regular" and "gluten free", what would be the simplest way to have your skill enable a user to order a pizza with a gluten-free crust without the skill prompting the user for that option?

- A. Don't set a prompt for the PizzaCrust entity item.
- B. For the PizzaCrust entity item, set the Extract With property to reference an entity with the single value "gluten free".
- C. In the composite bag, set the PizzaCrust entity item's Prompt for Value property to False. Then, once the the composite bag is resolved, check if the PizzaCrust entity has a value. If it doesn't, set its value to "regular".
- D. Don't set a prompt for the PizzaCrust entity item and set the Out of Order Extraction property to True.

ANSWER: A

QUESTION NO: 2

In your conversation flow, you want to make sure that users always see a message, even when there is no data to display. To implement this, you decide to use a system.setvariable component that verifies that the variable mydata contains a value and, if it does, sets the value of the

displayVar variable to the value of mydata. If no value is specified for mydata, then displayVar is set to the string 'No Data 1.

Which two BotML with Apache FreeMarker examples implement this requirement?

```
A) checkHasContent:
  component: "System.SetVariable"
  properties:
    variable: "displayVar"
    value: "${mydata.value!'No Data'}"

B) checkHasContent:
  component: "System.SetVariable"
  properties:
    variable: "displayVar"
    value: "${mydata.value?not_null?then(mydata.value,'No Data')}}"

C) checkHasContent:
  component: "System.SetVariable"
  properties:
    variable: "displayVar"
    value: "${!mydata.value 'No Data'}"

D) checkHasContent:
  component: "System.SetVariable"
  properties:
    variable: "displayVar"
    value: "<#if mydata.value != null>${mydata.value}<#else>'No Data'</#if>"

E) checkHasContent:
  component: "System.SetVariable"
  properties:
    variable: "displayVar"
    value: "${mydata.value?has_content?then(mydata.value,'No Data')}}"
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

ANSWER: B E

QUESTION NO: 3

Assuming that conversation is a reference to the Bots Node SDK, which two sets of code statements to access the custom component's orderid and userAccount input parameters are valid?

- A. `const order = conversation.properties().orderid;`
`const account = conversation.properties().userAccount;`
- B. `const order = conversation.request().variables[*orderid];`
`const account = conversation.request().variables['accountName 1];`
- C. `const order = conversation.variable(1 'orderid');`
`const account = conversation.variable('accountName1);`

D. `const { orderId } = conversation.properties();
const { accountName } = conversation.properties();`

E. `const { orderId } = conversation.variable();
const { accountName } = conversation.variable();`

ANSWER: A E

QUESTION NO: 4

With the conversation variable being the reference to the Bots Node SDK, which statement correctly describes what happens when the custom component executes the following code?

```
conversation.reply( 1 HelloWorld 1 );
```

```
conversation.keepTurn(true);
```

```
done();
```

- A. The code prints the "HelloWorld" message in response to the next user message.
- B. The code triggers dialog flow navigation to a state, which has its name mapped to the current dialog flow state's HelloWorld action transition.
- C. The code prints "HelloWorld" multiple times until an infinite loop gets detected by the dialog flow engine.
- D. The code prints "HelloWorld" as a message and triggers dialog flow navigation to the next state.
- E. The code prints "HelloWorld" as a message and waits for user input.

ANSWER: A

QUESTION NO: 5

What happens after the skill output "Welcome to our store's chat bot!" in this dialog flow?

```
output1:
  component: "System.Output"
  properties:
    text: "Welcome to our store's chat bot!"
    keepTurn: true
  transitions:
    next: "output2"

output2:
  component: "System.Output"
  properties:
    text: "You can ask me about what products we have in the store."
  transitions:
    next: "intent"
```

- A. The skill displays "Welcome to our store's chat bot!" again.
- B. The skill goes to output2, outputs "You can ask me about what products we have in the store", and then returns control to the user.
- C. The skill returns control to the user. After the user enters text, it goes to the intent state.
- D. The skill returns control to the user. After the user enters text, it goes to output2 and outputs "You can ask me about what products we have in the store."

ANSWER: A

QUESTION NO: 6

Which three statements are true about composite bag entities?

- A. They define a business domain object as a collection of related system entities and custom entities.
- B. When you add entities to the composite bag, you can control how they get extracted in related to other entities and when they are prompted for.
- C. The composite bag will always enforce that every entity has a valid value before allowing the conversation to move on to the next state in the dialog flow.
- D. You need to create a separate composite bag to handle nonentity types such as strings. Locations, and attachments.
- E. The composite bag can resolve all entity values using only a single state in the dialog flow.

ANSWER: B C E

QUESTION NO: 7

How do you declare a context variable for an entity?

- A. Set the variable type to "nlresult".

- B. Set the variable type to the same name as the entity.
- C. Set the variable type to "entity".
- D. Set the variable type to "map" and reference the value by the entity name.

ANSWER: C

QUESTION NO: 8

Which two statements are true the purpose of the compose bag entity item Out of Order Extraction property?

- A. It should not be used for String entity items.
- B. When set to True, an entity item value can be updated when prompting for a different entity item value.
- C. Only set to True, an entity item value can be updated when prompting for a different entity item value.
- D. Only when set to true can entity values be specific in any order in the initial user input.
- E. When set to True, an entity item cannot be resolve until a dependent entity item is first resolved.

ANSWER: C E

QUESTION NO: 9

Which three options are true for the system, entityToResolve variable?

- A. System.entityToResolve can reference the resolve value of all entity values defined within the skill.
- B. The system.entityToResolve variable tracks an entity value, that is, as you resolve entities in the composite bag, it references the current entity resolved.
- C. `#{system.entityToResolve.value.userinput}` returns the text entered by the user.
- D. `#{system.entityToResolve.value.resolvingField}` returns the text entered by the user.
- E. The system.entityToResolve variable can be referenced from within the system. ResolveEntities and system.commonResponse components to display, for example, information about the entity that has been resolved.

ANSWER: B C D

QUESTION NO: 10

You are designing a skill for a railway company. You created a value list entity (StationEntity), which is the list of all possible train stations To resolve an intent (RouteIntent), you need to determine whether the user is asking for a route which is either to a station or from a station.

Which statement describes the most robust and efficient approach for extracting this information from the user input?

- A.** Create a value list entity called ToFromEntlty with values of "to" and "from" and with appropriate synonyms for each value. Create a value list called DirectionStation and add ToFromEntity and StationEntity to this. Then, add DirectionStation to the RouteInfoIntent.
- B.** Create two derived entities based on StationEntity. In one entity, set the preceding phrase to "to" (along with any required synonyms). In the other entity, do the same but with the preceding phrase "from". Add both entities to the RouteInfoIntent intent.
- C.** Duplicate StationEntity. In one version, prefix all of the train station names with "to" and in the other prefix with "from". Then add both entities to the RouteInfoIntent intent.
- D.** Add StationEntity to the RouteInfoIntent intent and then update the training data with phrases beginning with "from".

ANSWER: B