



## Randomizing Responses from your Bot with Oracle Digital Assistant

Grant Ronald (v. 2.0) April 2019

One element to the success of your bot is the tone and voice your bots uses. Informal and “chatty” response may be a more engaging and better fit with your brand. In line with this approach, you would probably want to ensure your bot replies are not formulaic and predictable. In this article I demonstrate how you can easily randomize the responses from your bot.

### Randomizing the responses from your bot

There are many different ways you can achieve this functionality. To keep the implementation as simple as possible I’ve done so by defining a map with message types within the BotML. Each message type contains an array with the possible messages for that message type. A random message is retrieved using a simple pseudo random number generator (based on date/time) which picks one of your messages defined within one of the arrays.

Furthermore, it’s quite simple to extend the map to include different message types for different points in the conversation: welcome, goodbye, not understanding a response etc.

The sample is simple and self-documenting. Import the bot and run it you will get three random messages for welcome, unresolved and goodbye.

**Note:** When resolving composite bag entities, you can define random prompts on the entity or composite bag item itself (by specifying the same sequence number). More information on this feature can be found in the Oracle Digital Assistant documentation and on [ODA Enablement](#) - Real world entity extraction with composite bag.

```

setMap:
  component: "System.SetVariable"
  properties:
    variable: "messages"
    value:
      welcome:
        - "Hey, what's up"
        - "Hi there, ready to help!"
        - "How can I help you"
        - "Barry the bot ready to help you!"
        - "Great to see you, how can I help"

      goodbye:
        - "See you later"
        - "Thanks for trying me out"
        - "Thanks, see you soon"
        - "Bye!"

      confused:
        - "I'm sorry I didn't understand"
        - "Oops, I'm only a bot and I didn't quite get that"
        - "Sorry I'm not doing a great job today"

```

Figure 1 - a map with message types and corresponding array of possible messages

```

#Say a Random Welcome Message
sayAWelcomeMessage:
  component: "System.Output"
  properties:
    text: "${messages.value.welcome[.now?long % messages.value.welcome?size]}"
    keepTurn: true

#Say a Random Don't Understand Message
sayAUnresolvedMessage:
  component: "System.Output"
  properties:
    text: "${messages.value.confused[.now?long % messages.value.confused?size]}"
    keepTurn: true

#Say a Random Bye Message
sayAByeMessage:
  component: "System.Output"
  properties:
    text: "${messages.value.goodbye[.now?long % messages.value.goodbye?size]}"
    keepTurn: true
  transitions:
    return: done

```

Figure 2 - retrieving a random message for a message type