

ORACLE®

Oracle Digital Assistant

The Complete Training

Application Initiated Conversation

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

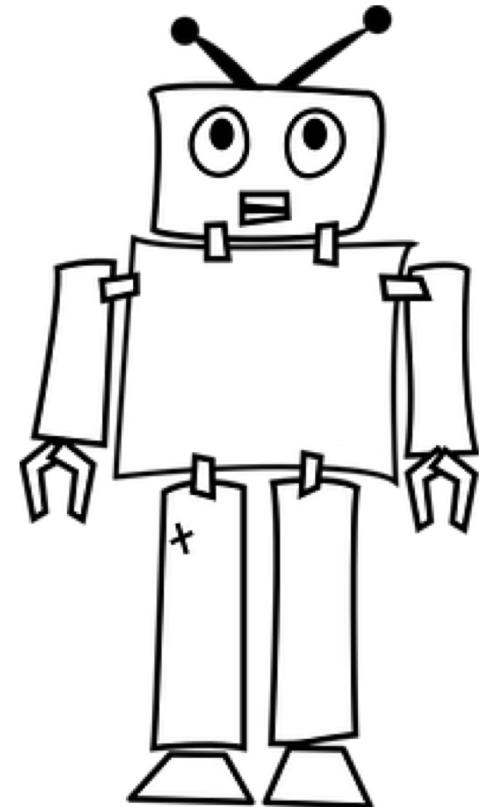
Topic agenda

- 1 ➤ Application events
- 2 ➤ Configure skill
- 3 ➤ Configure application and user channel
- 4 ➤ Configure external application
- 5 ➤ How to use embedded tester to test AIC

Topic agenda

- 1 ➤ Application events
- 2 ➤ Configure skill
- 3 ➤ Configure application and user channel
- 4 ➤ Configure external application
- 5 ➤ How to use embedded tester to test AIC

Digital Assistant can now **initiate the conversation** with a user, based on external application events!



AIC: Use case scenarios

Approval notification

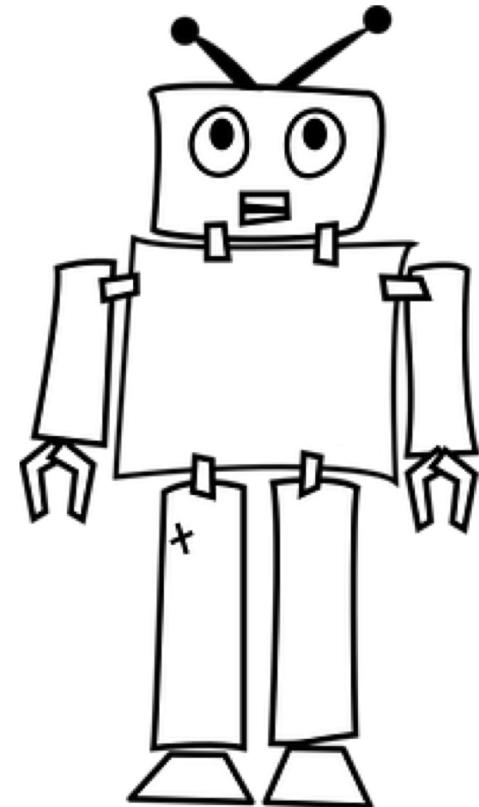
Grant, you have a new expense report to approve, reply to this SMS from below options

- 1. View
- 2. Approve
- 3. Deny

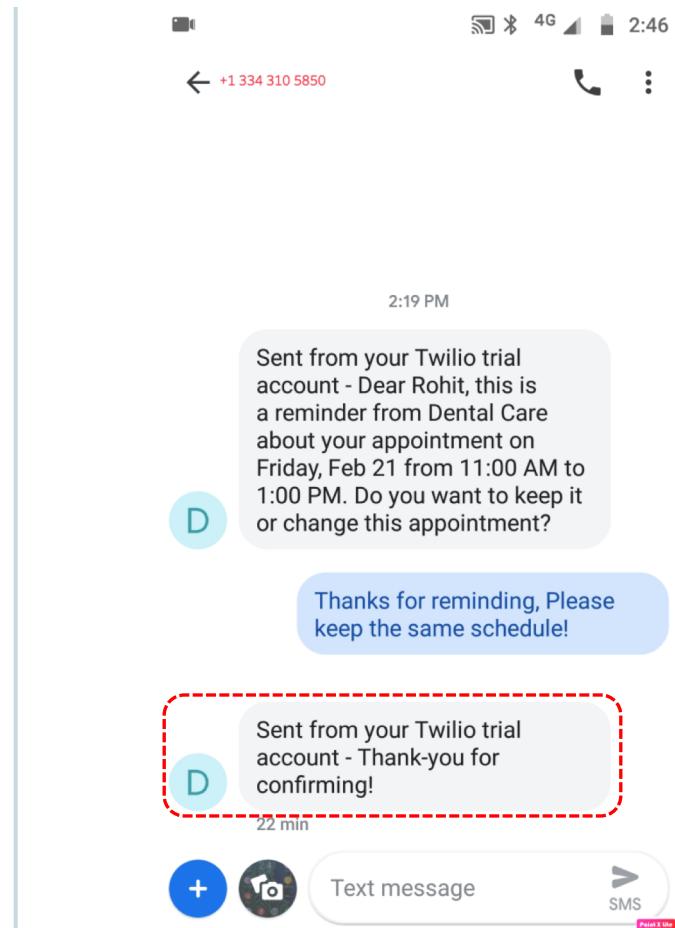
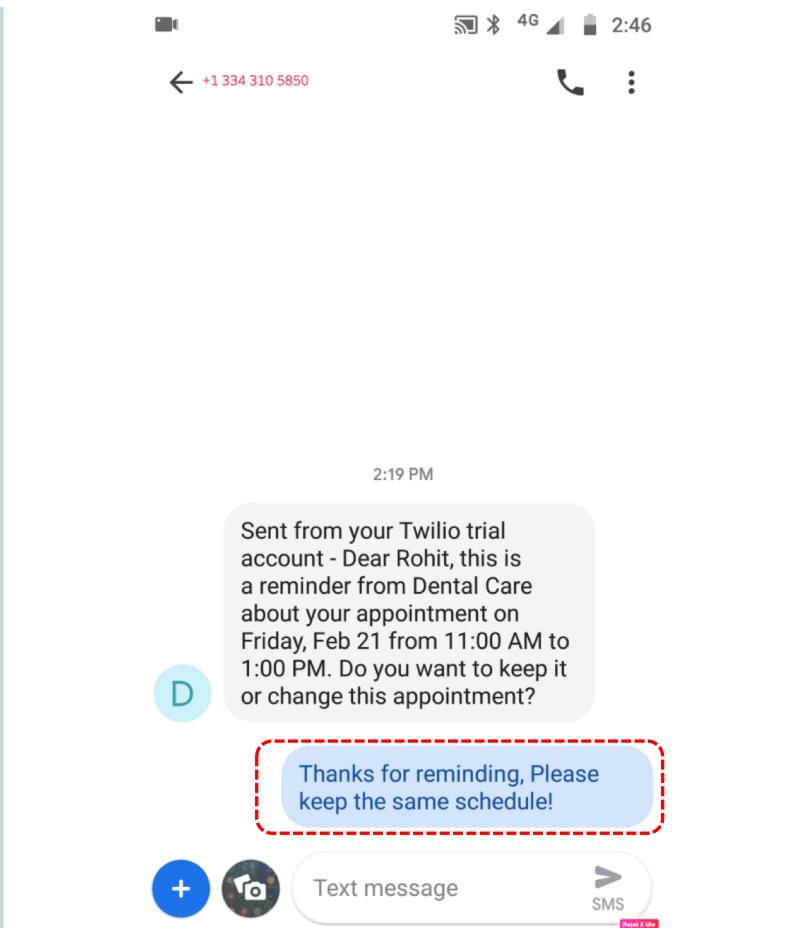
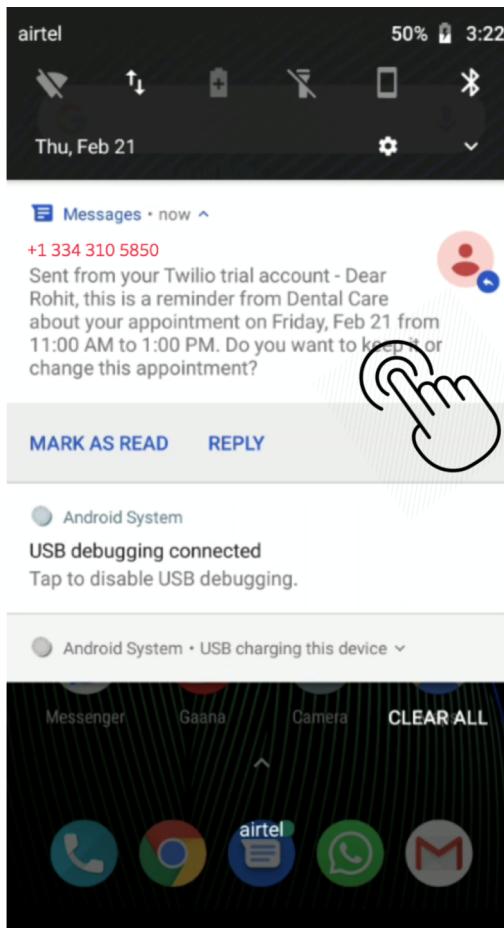
Upgrade notification

- All Registered Digital Bank users "the magnetic stripe cards" will have to be replaced with the EMV chip, PIN-based ones by December 31, 2018. Please update if there is a change in your correspondence address.

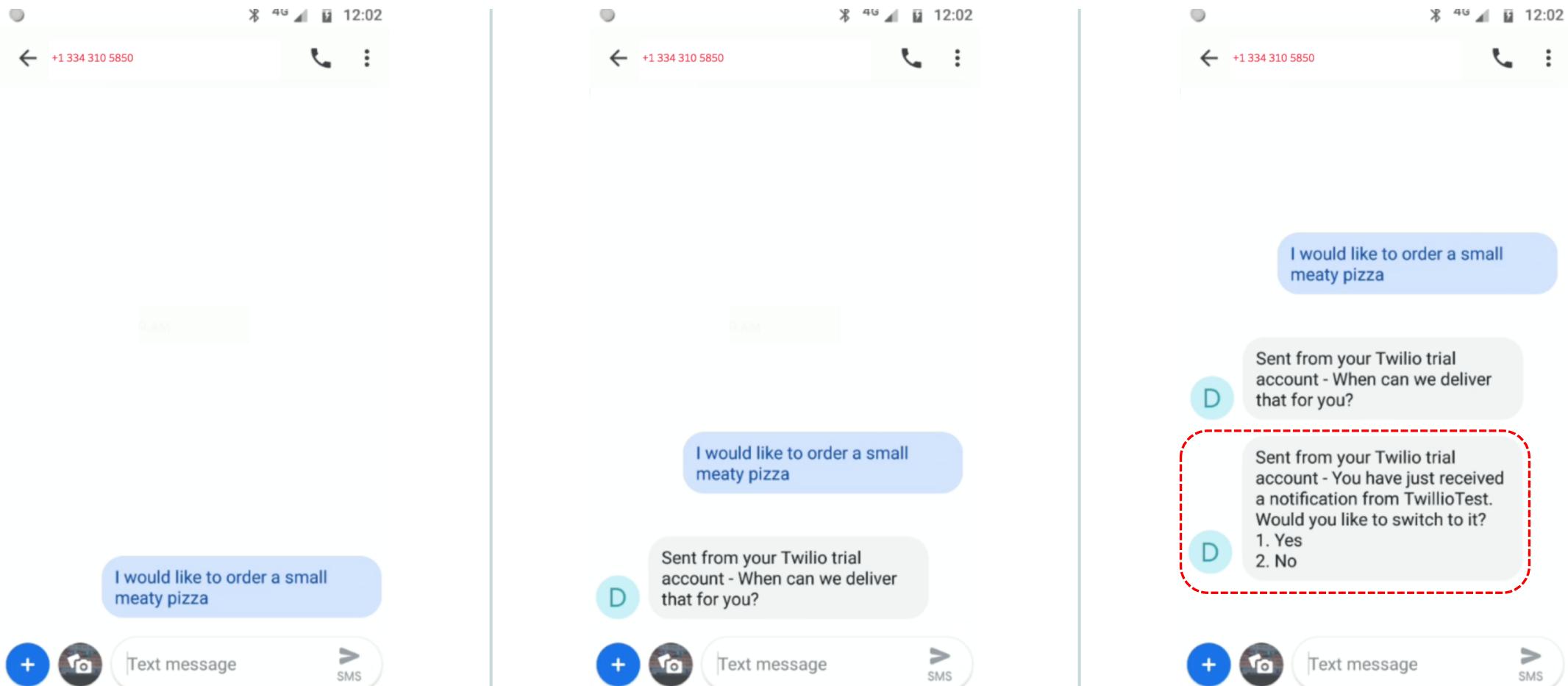
The user may OR may not have **started a conversation** while receiving the notification – DA handles both cases gracefully!



AIC scenario: When user has not started conversation



AIC scenario: When user is in conversation



How event-driven conversation works

Let's see the actors involved in this process!

1. External Application

2. Oracle Digital Assistant

3. Twilio

4. SMS client on user phone

How event-driven conversation works

1. External Application

2. Oracle Digital Assistant

3. Twilio

4. SMS client on user phone

How event-driven conversation works

1. External Application

2. Oracle Digital Assistant

3. Twilio

4. SMS client on user phone

- Creates an event for a user and sends it to ODA
- E.g: Dental care backend system generating appointment reminders for user

How event-driven conversation works

1. External Application

2. Oracle Digital Assistant

3. Twilio

4. SMS client on user phone

- Reacts to the event sent by external application
- Depending on the event, this app invokes one of the ODA's skills at a specific state
- Sends message to the user and wait at this state

How event-driven conversation works

-
1. External Application
 2. Oracle Digital Assistant
 3. Twilio
 4. SMS client on user phone

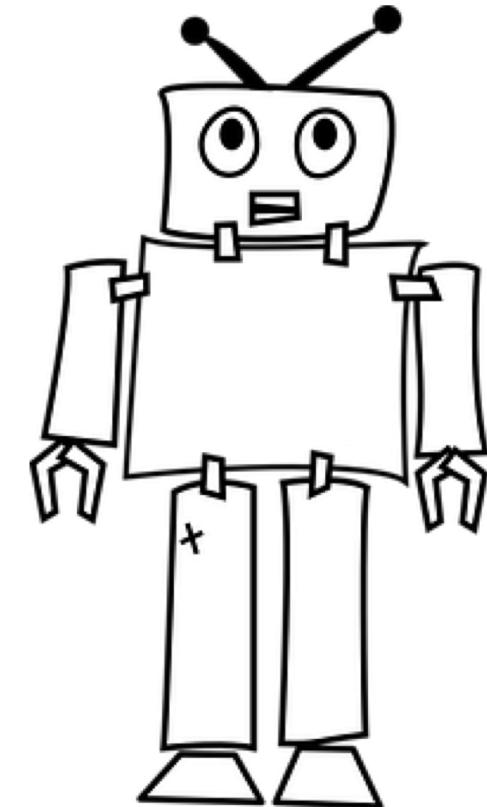
- Send and receive text messages over the network

How event-driven conversation works?

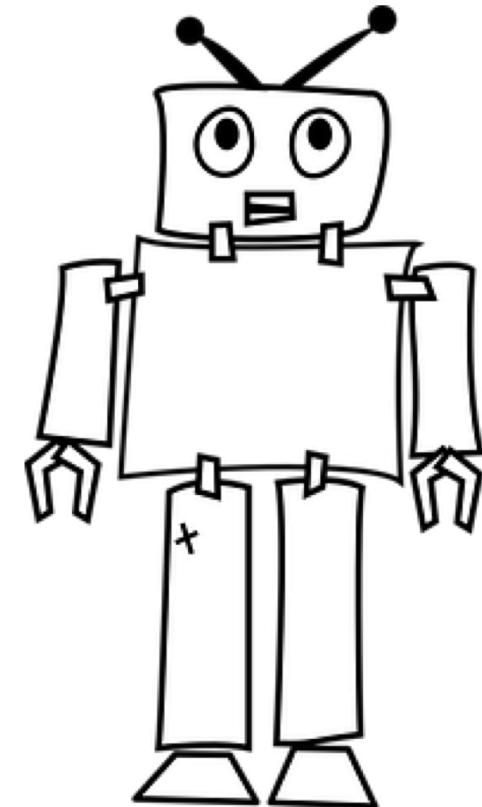
-
1. External Application
 2. Oracle Digital Assistant
 3. Twilio
 4. SMS client on user phone

- Sends/receives SMS messages to/from Twilio

You enable AIC by configuring the **skill**,
the **digital assistant** it belongs to, and
by adding specific **parameters** to the
payload of the external app's event
message.



The external application sends a payload to ODA, which has a **key** that identifies the start of the state in the dialog flow. This is referred as **payload-to-state mapping**



Application initiated conversation set up

1. Oracle Digital Assistant

2. Twilio

3. External Application

4. SMS client on user phone

Application initiated conversation set up

-
1. Oracle Digital Assistant
 2. Twilio
 3. External Application
 4. SMS client on user phone

- Configure application channel
 - Allows the application to talk to ODA
- Configure Twilio SMS channel
 - Allows ODA to communicate on an SMS channel
- Add payload-to-state mapping

Application initiated conversation set up

-
1. Oracle Digital Assistant
 2. Twilio
 3. External Application
 4. SMS client on user phone

- Configure SMS enabled Twilio number
- Link skill to the Twilio number

Application initiated conversation set up

1. Oracle Digital Assistant

2. Twilio

3. External Application

4. SMS client on user phone

- Configure external application
 - External application sends specific events with parameters to DA/skill
 - Structure of payload in documentation
 - <https://docs.oracle.com/en/cloud/paas/digital-assistant/use-chatbot/application-initiated-conversations.html#GUID-5DF067AA-E30B-4711-9C95-CEEE91F5F99B>

Include these properties in the message payload:

- **userId**—The actual phone number of the user. This is one of the numbers that are associated with the phone number that's both assigned to the Twilio account and used by the Twilio channel configuration.
- **payloadType**—The name of the payload that's mapped to the initialization state in the dialog flow.
- **skillName**—The name (identifier) of the digital assistant or the name of the skill that's registered to the digital assistant and the recipient of the application event message payload.
- **channelName**—The name of the Twilio Channel that's configured for the digital assistant. The channel configuration uses the number assigned to the Twilio account. For the System test channel, you need to define **userId** with the system-generated ID and **channelName** with the name of the System test channel.
- **variables**—The values that get passed to the dialog flow's context variables. If the corresponding context variables have been defined in the dialog flow, then they will be populated with the corresponding values passed from the application event message payload.

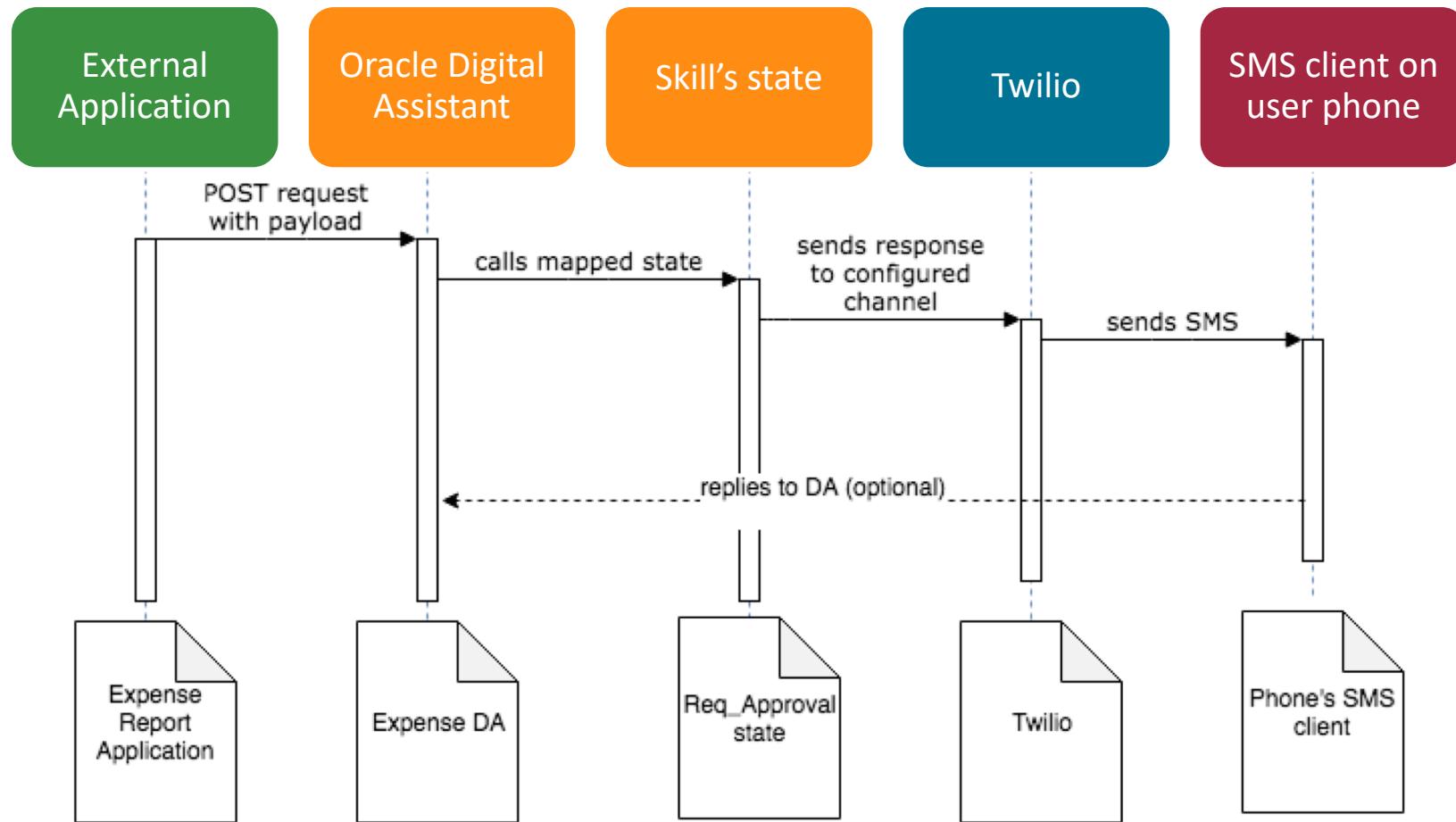
```
{  
  "userId": "+14255555000",  
  "messagePayload": {  
    "type": "application",  
    "payloadType": "accountType",  
    "skillName": "FinancialBot",  
    "channelName": "MyTwilioChannel",  
    "variables": {  
      "accountType": "checking",  
      "txntype": "credits"  
    }  
  }  
}
```

Application initiated conversation set up

-
1. Oracle Digital Assistant
 2. Twilio
 3. External Application
 4. SMS client on user phone

- User communicates with DA using SMS channel

Communication flow: How AIC works



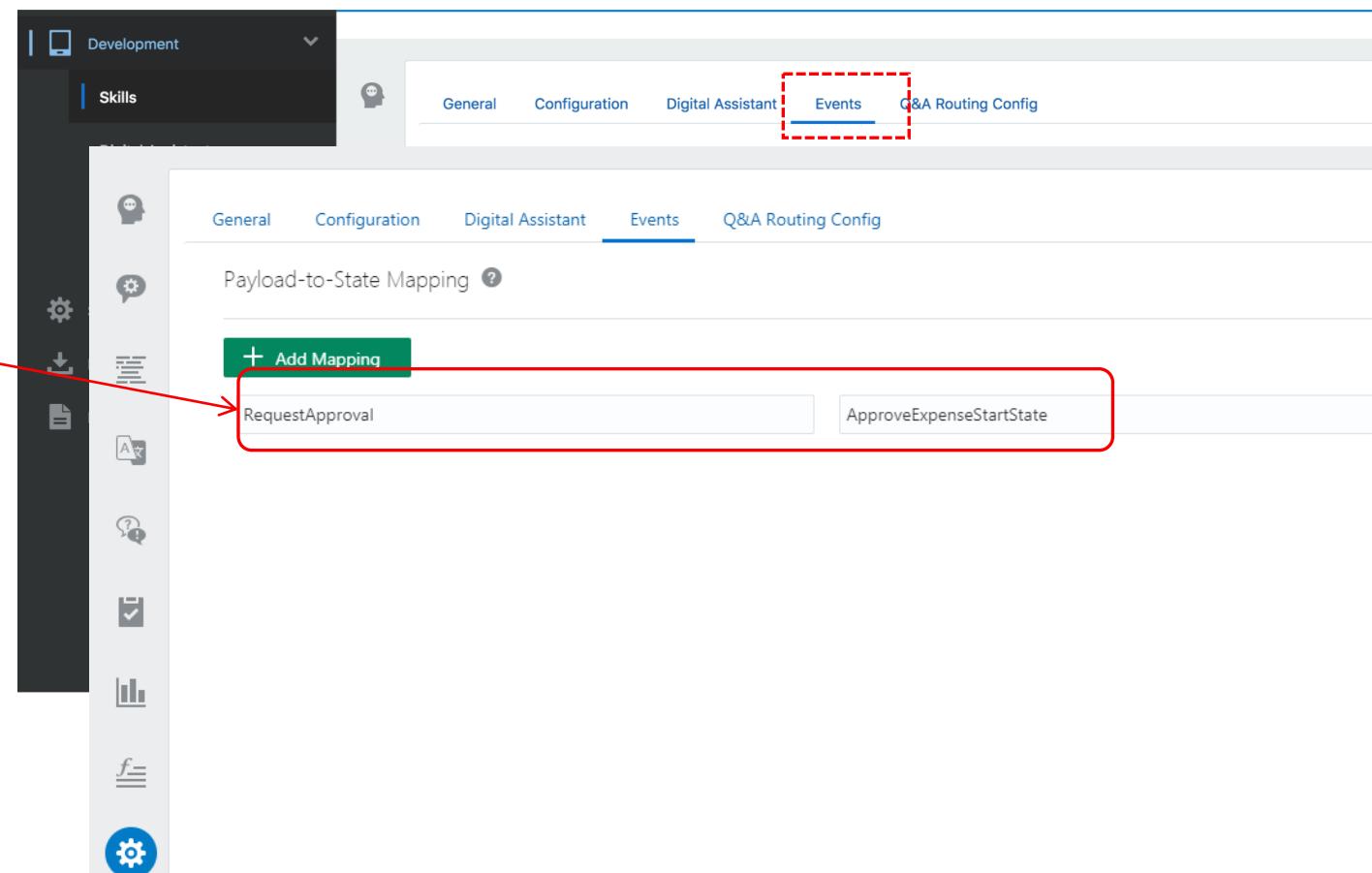
Topic Agenda

- 1 ➤ Application events
- 2 ➤ Configure skill
- 3 ➤ Configure application and user channel
- 4 ➤ Configure external application
- 5 ➤ How to use embedded tester to test AIC

Configure skill – add payload to state mapping

What does message payload look like?

```
{  
  "userId": "+919871996112",  
  
  "messagePayload": {  
  
    "type": "application",  
  
    "payloadType": "RequestApproval",  
  
    "skillName": "ExpenseBot",  
  
    "channelName": "twilioSMS_ch",  
  
    "variables": {  
      "approvalType": "expense"  
    }  
  
  }  
}
```



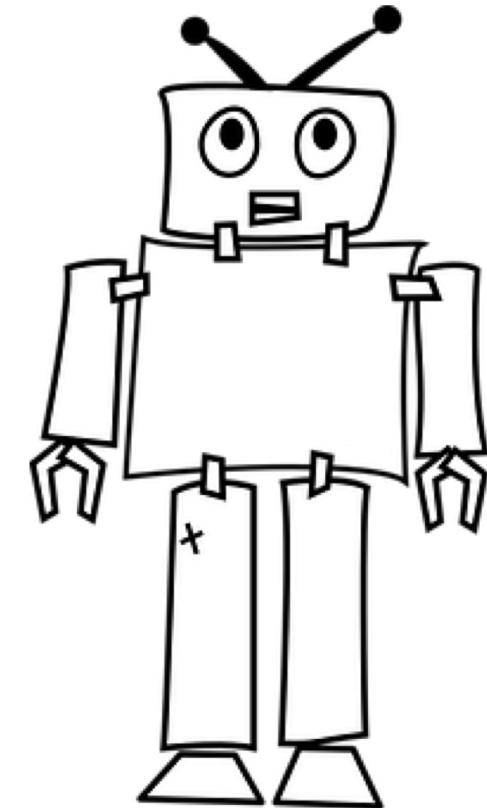
Topic agenda

- 1 ➤ Application events
- 2 ➤ Configure skill
- 3 ➤ Configure application and user channel**
- 4 ➤ Configure external application
- 5 ➤ How to use embedded tester to test AIC

We need to configure two types of channels to achieve AIC.

Twilio/SMS channel for user to interact with skill bot

Application channel through which the messages from the external app are sent to the skill bot



Configuring application and user channel

- Application channel exposes skill to external application
 - This results in URL and secret
 - External app uses this to “trigger” skill

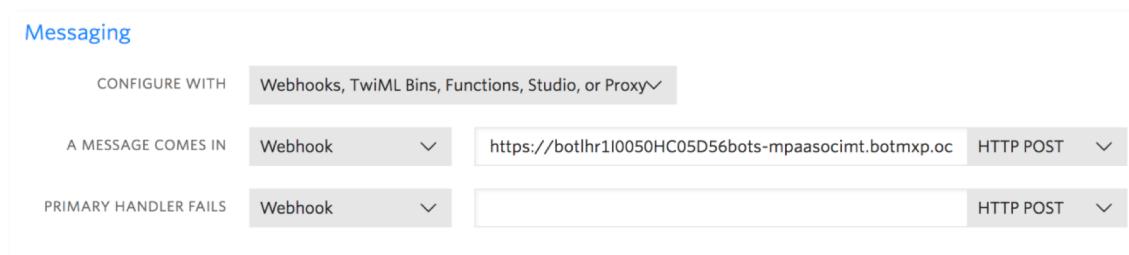
The screenshot shows the Oracle Autonomous Digital Assistant interface. At the top, there's a navigation bar with 'Development' selected, followed by 'Home', 'Development', 'Skills', 'Digital Assistants', and 'Channels'. Below this is a secondary navigation bar with 'Users', 'Agent Integrations', 'Applications' (which is underlined), and 'System'. A large central panel is titled 'ExpenseBotAppChannel' and contains the following fields:

- Application Enabled:** A toggle switch is turned on.
- Name:** ExpenseBotAppChannel
- Description:** Expense Management bot - Manage Approvals
- Outbound Application URL:** A placeholder text field: "An outbound application URL you need to provide".
- Use Authenticated User ID:** A toggle switch is turned off.
- Secret Key:** nlmWG14QCnxEmesvgUacUS8v6KDNb22t (with a 'Regenerate' link)
- Inbound URL:** https://botlhr1I0050HC05D56bots-mpaasocimt.botmvp.ocp.oraclecloud.com:443/connectors/v1/tenants/fdc5-6d466372210e4300bb3f1f4db15e8e96c/listeners/application/channels/c66114d-0f90-4d77-a191-3a037d8bd6e3

A red box highlights the 'Channels' menu item in the top navigation and the inbound URL field in the application configuration panel. To the right of the main interface, there's a sidebar with a robot icon and the text: "You haven't yet integrated a skill or digital assistant with this application. Create a configuration to integrate an external app with this application." It also includes a "Tell me more" button and a "+ Application Configuration" button.

Configuring application and user channel

- User channel exposes skill to Twilio SMS service
 - Allows skill to communicate with user via SMS
 - Twilio account required for setting up
 - Provide Twilio account SID and token
- Paste webhook URL back into Twilio
 - *A Message Comes In* field



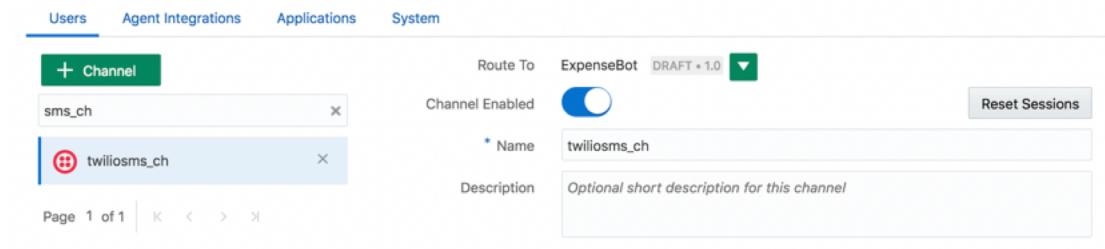
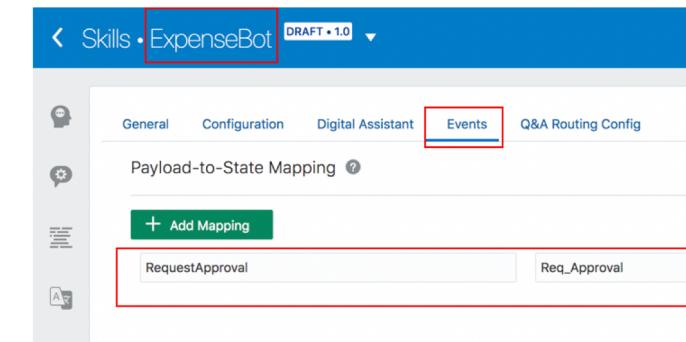
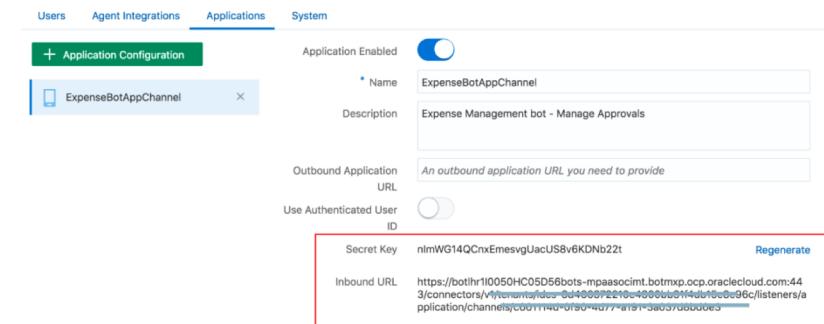
The screenshot shows the 'Create Channel' dialog box in the Oracle Autonomous Digital Assistant. The 'Channel Type' dropdown is set to 'Twilio SMS' and is highlighted with a red box. Other fields include 'Name' (Channel name), 'Description' (Optional short description for this channel), 'Account SID' (Copy from Twilio account), 'Auth Token' (Copy from Twilio account), 'Phone Number' (Copy from Twilio account), and 'Session Expiration (minutes)' (60). A 'Create' button is at the bottom right.

Topic Agenda

- 1 ➤ Application events
- 2 ➤ Configure skill
- 3 ➤ Configure application and user channel
- 4 ➤ Configure external application
- 5 ➤ How to use embedded tester to test AIC

Configuring external application

- External application needs to know
 - Inbound URL and secret from application channel
 - Twilio SMS channel name
 - Phone number of person receiving the SMS
 - Skill payload name
 - DA or skill name with version



Configure the external app (contd.)

- POST URL
 - Will be Application type channel's inbound URL
- Headers
 - X-Hub-Signature header with the message signature signed with SHA256
- Message Payload

```
https://xxx.ocp.oraclecloud.com:443/  
connectors/v1/tenants/idcs-xxx/  
listeners/application/channels/  
c661114d-0f90-4d77-a191-3a037dxxxxe3
```

```
Content-Type: application/json  
X-Hub-Signature: sha256={{secretkey}}
```

```
{  
  "userId": "+919871996112",  
  "messagePayload": {  
    "type": "application",  
    "payloadType": "RequestApproval",  
    "skillName": "ExpenseBot",  
    "channelName": "twilioSMS_ch",  
    "variables": {  
      "approvalType": "expense"  
    }  
  }  
}
```

Sample request code snippet

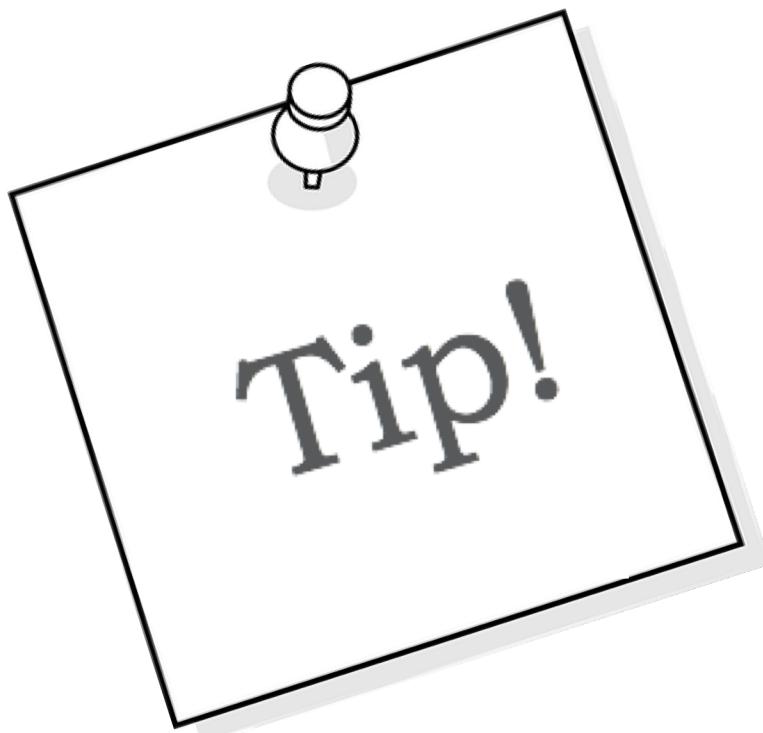
```
var request = require("request");

var options = {
  method: 'POST',
  url: 'https://XXX.botmxp.ocp.oraclecloud.com:443/connectors/v1/tenants/
idcs-XX466372210e4300bb31f4db1XXXc/listeners/application/
channels/c661114d-0f90-4d77-a191-3a037d8bd6e3',
  headers: {
    'X-Hub-Signature': 'sha256=9f0d75336379aaa5f87bcc3b84f488f0c9eac50985006c4206b4a828494aXXX',
    'Content-Type': 'application/json'
  },
  body: {
    userId: '+919871996112',
    messagePayload: {
      type: 'application',
      payloadType: 'RequestApproval',
      skillName: 'ExpenseBot',
      channelName: 'twilioSMS_ch',
      variables: {
        approvalType: 'expense'
      }
    }
  },
  json: true
};

request(options, function (error, response, body) {
  if (error) throw new Error(error);

  console.log(body);
});
```

Response status codes



- 202 Accepted – signifies that request is accepted
- 403 Forbidden – X-Hub-Signature is incorrect
- 404 Not found – POST URL is incorrect

Topic Agenda

- 1 ➤ Application events
- 2 ➤ Configure skill
- 3 ➤ Configure application and user channel
- 4 ➤ Configure external application
- 5 ➤ How to use embedded tester to test AIC

How to use embedded tester to test SMS

System Bot Tester

Channels

Users Agent Integrations Applications System

 System_Global_Test

Channel Enabled

Name **System_Global_Test**

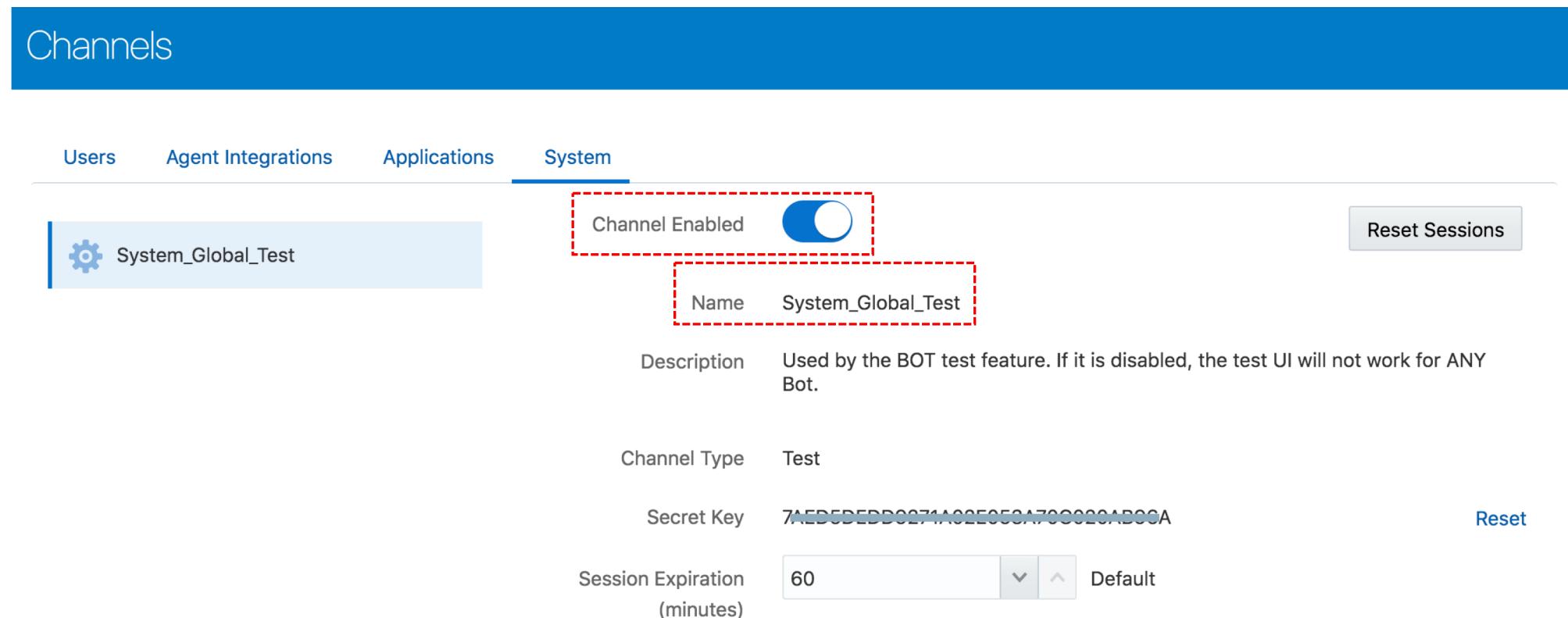
Description Used by the BOT test feature. If it is disabled, the test UI will not work for ANY Bot.

Channel Type Test

Secret Key **7AED5DEDD0271A02E053A700020AB0CA** [Reset](#)

Session Expiration (minutes) **60** [▼](#) [▲](#) Default

Reset Sessions



How to use embedded tester to test SMS

Find user ID for system test channel

The screenshot shows the Oracle Autonomous Digital Assistant interface. On the left, a conversation with the Lululemon RetailBot is displayed. The bot has asked for the user's gift card number. On the right, the developer tools Network tab is open, showing an XHR request. The request URL is `/line.routing.routerResponse"}]}}, "userId": "4076119"}}`. The `responses?_=1547789516676` entry in the list is highlighted with a red box. The developer tools also show a list of other responses with IDs like `1547789516673`, `1547789516674`, etc.

How to use embedded tester to test SMS

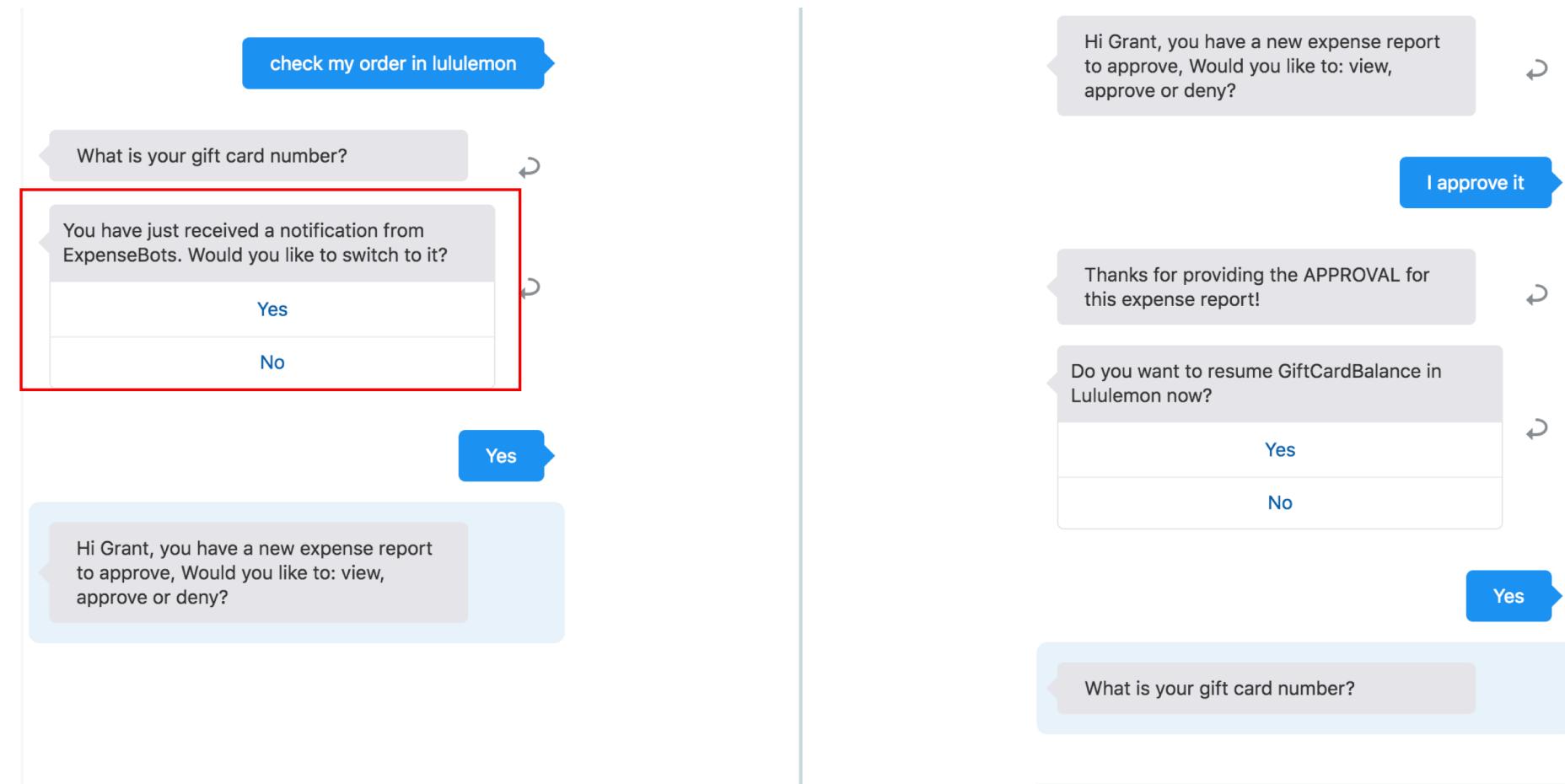
Message payload

The screenshot shows the Postman application interface for testing an API. The top navigation bar includes 'POST' (selected), the URL 'https://botlhr1I0050HC05D56bots-mpaasocimt.botmxp.ocp.oraclecloud.com:443/connectors/v1/tenan...', 'Params', 'Send' (blue button), and 'Save'. Below the URL are tabs for 'Authorization', 'Headers (2)', **Body** (selected), 'Pre-request Script', 'Tests', 'Cookies', and 'Code'. Under 'Body', there are options for 'form-data', 'x-www-form-urlencoded', 'raw' (selected), 'binary', and 'JSON (application/json)'. The JSON payload is displayed in a code editor:

```
1 [
2   "userId": "4076119",
3   "messagePayload": {
4     "type": "application",
5     "payloadType": "RequestApproval",
6     "skillName": "ExpenseBots",
7     "channelName": "System_Global_Test"
8   }
9 ]
10
```

The 'Body' tab is selected at the bottom left, along with 'Cookies', 'Headers (6)', and 'Test Results'. At the bottom right, the status is shown as 'Status: 202 Accepted'.

How to use embedded tester to test SMS





Oracle Digital Assistant Hands-On

TBD

Integrated Cloud Applications & Platform Services

ORACLE®