

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.

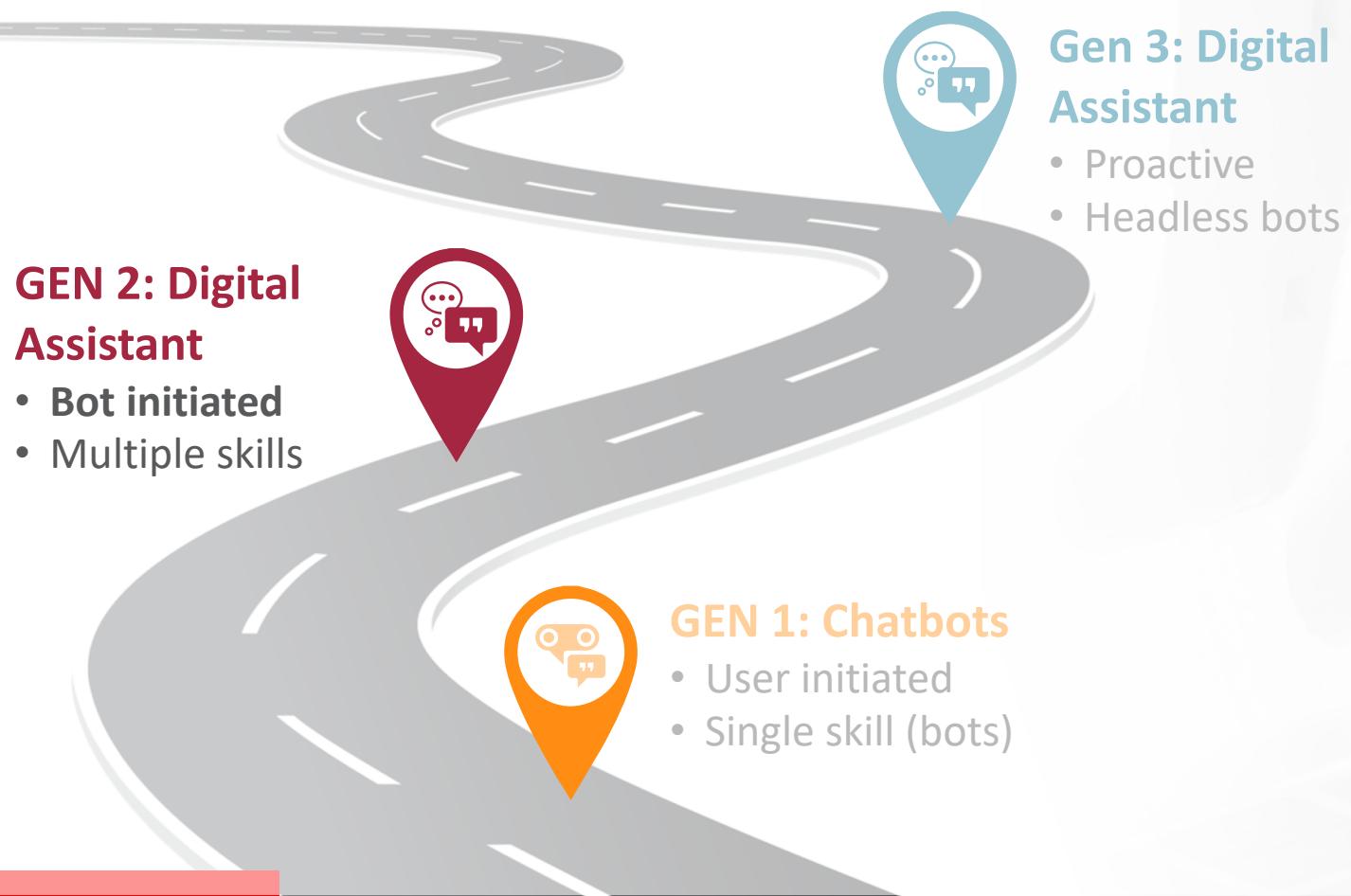
Topic agenda

- 1 ➤ About application initiated conversations
- 2 ➤ How application initiated conversation works
- 3 ➤ Setup
- 4 ➤ Channel configuration
- 5 ➤ Configure external application
- 6 ➤ How to test application initiated conversations

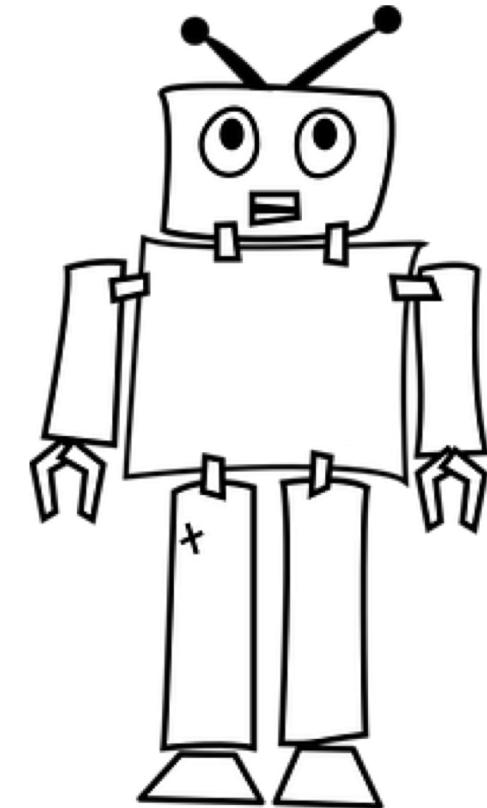
Topic agenda

- 1 ➤ About application initiated conversations
- 2 ➤ How application initiated conversation works
- 3 ➤ Setup
- 4 ➤ Channel configuration
- 5 ➤ Configure external application
- 6 ➤ How to test application initiated conversations

From chatbots to digital assistants



Digital Assistant can **initiate** the
conversation with a user



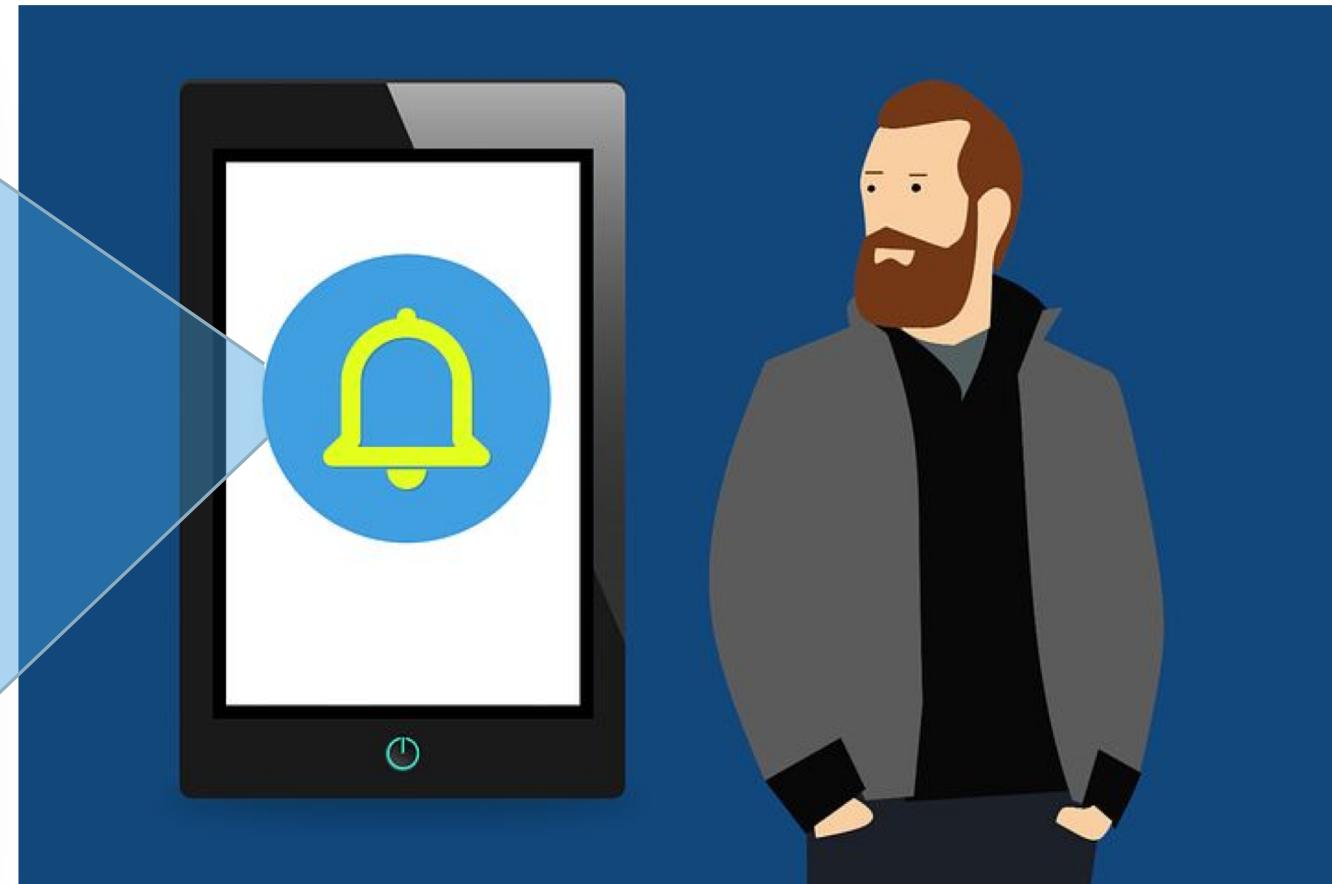
Business Usecase

Expense approval:

John Smith
735.00 USD

SFO OOW Hotel

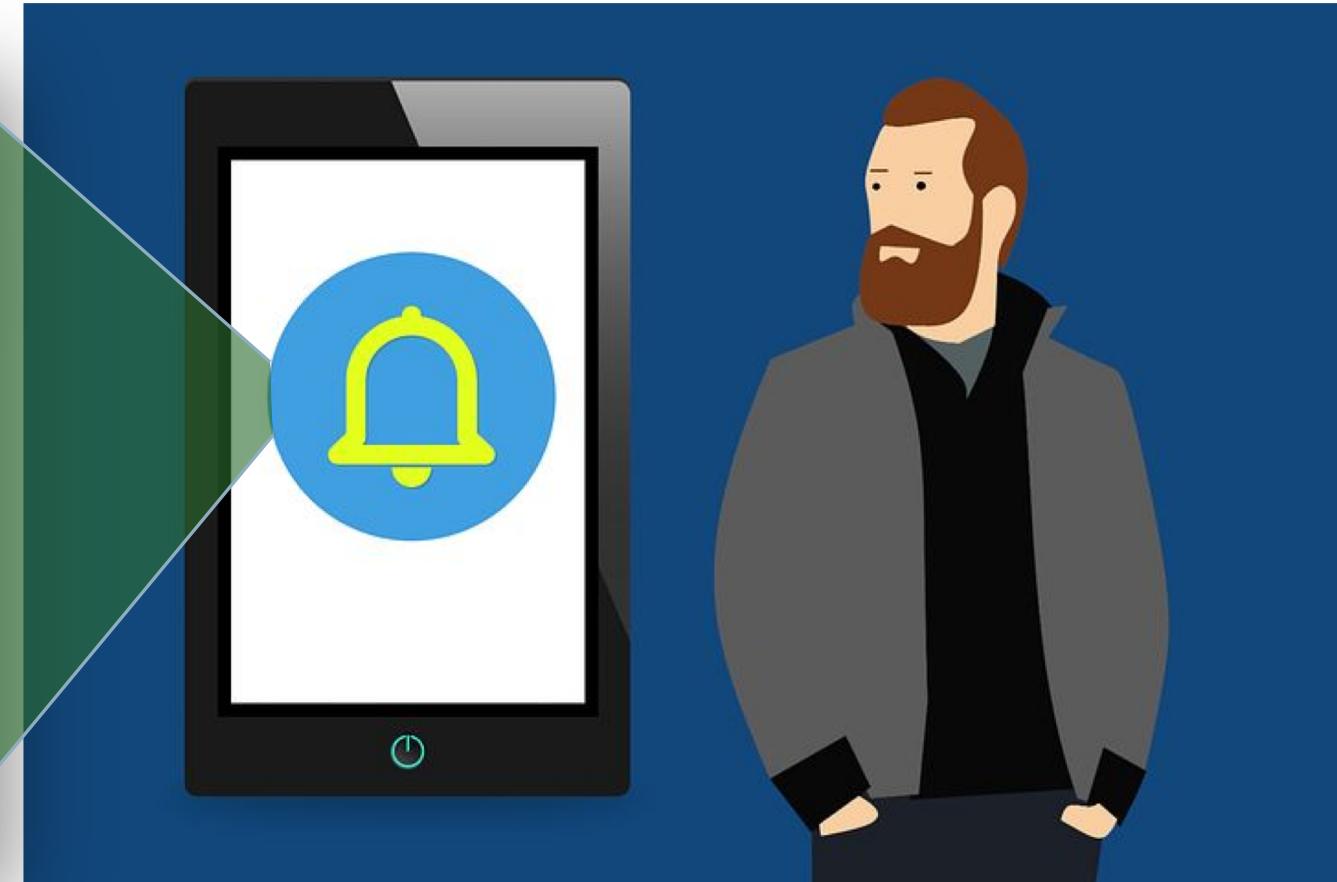
- 1. View
- 2. Approve
- 3. Reject



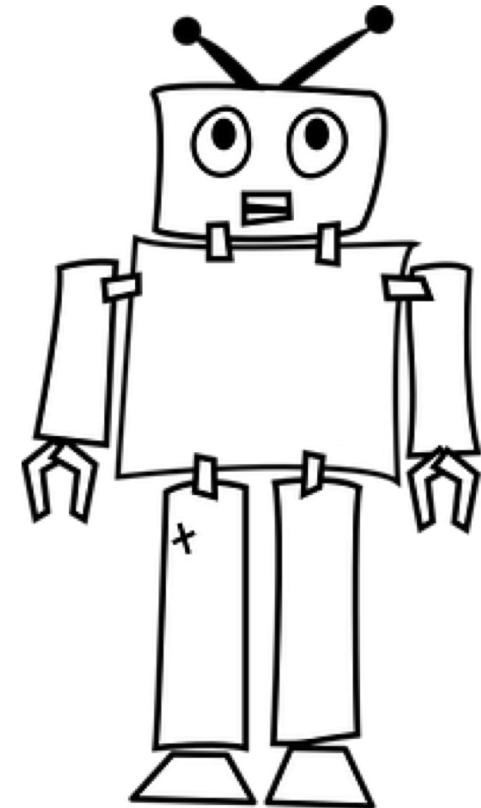
Consumer Usecase

By June 1, 2019, all Digital Bank customers must get their magnetic stripe cards replaced.

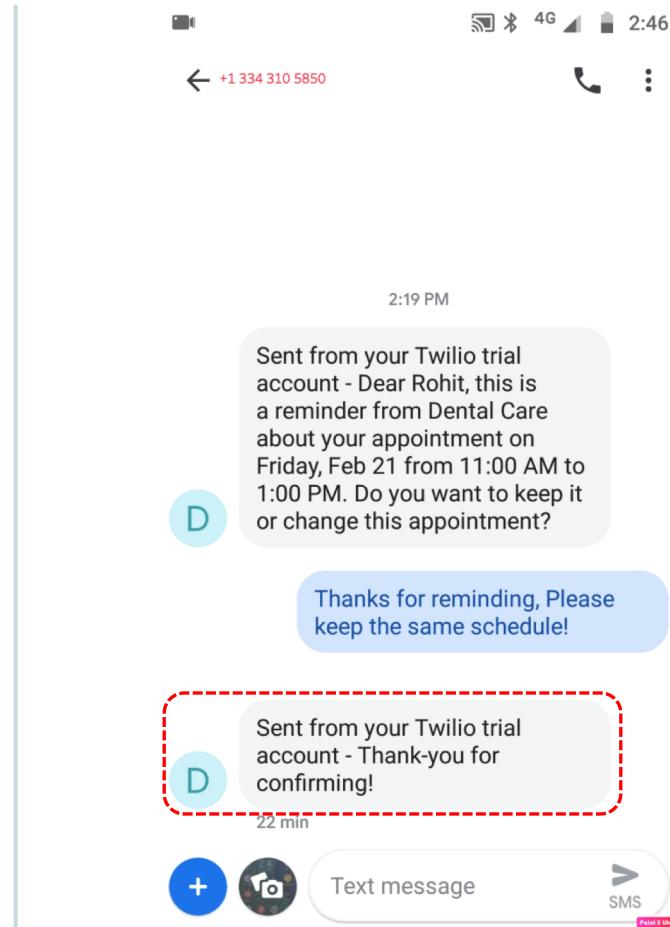
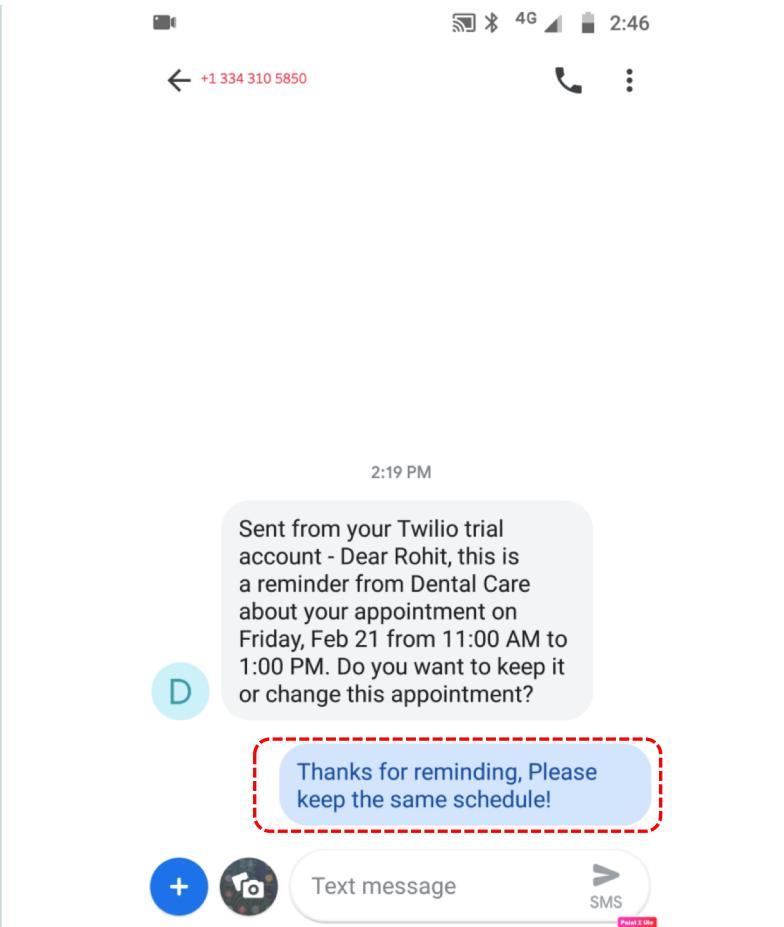
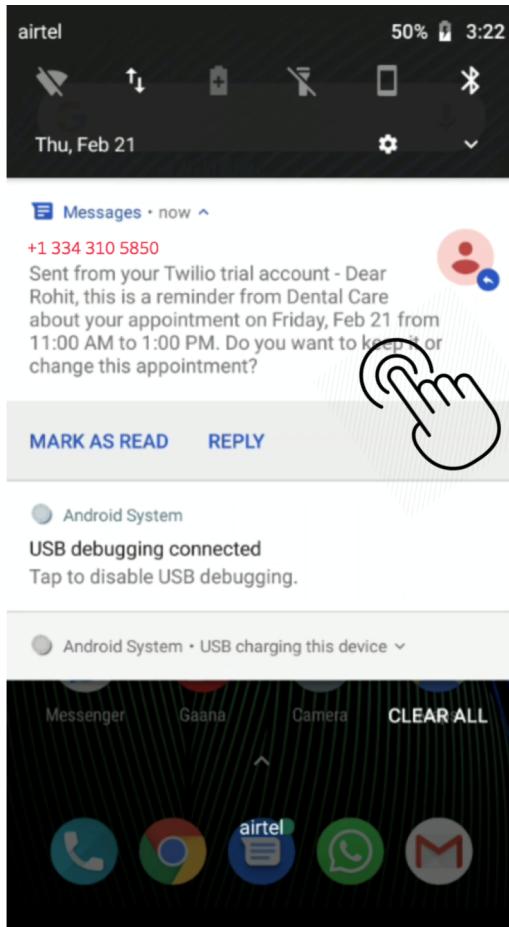
- 1. Detailed Information
- 2. Replace Card
- 3. Remind Later



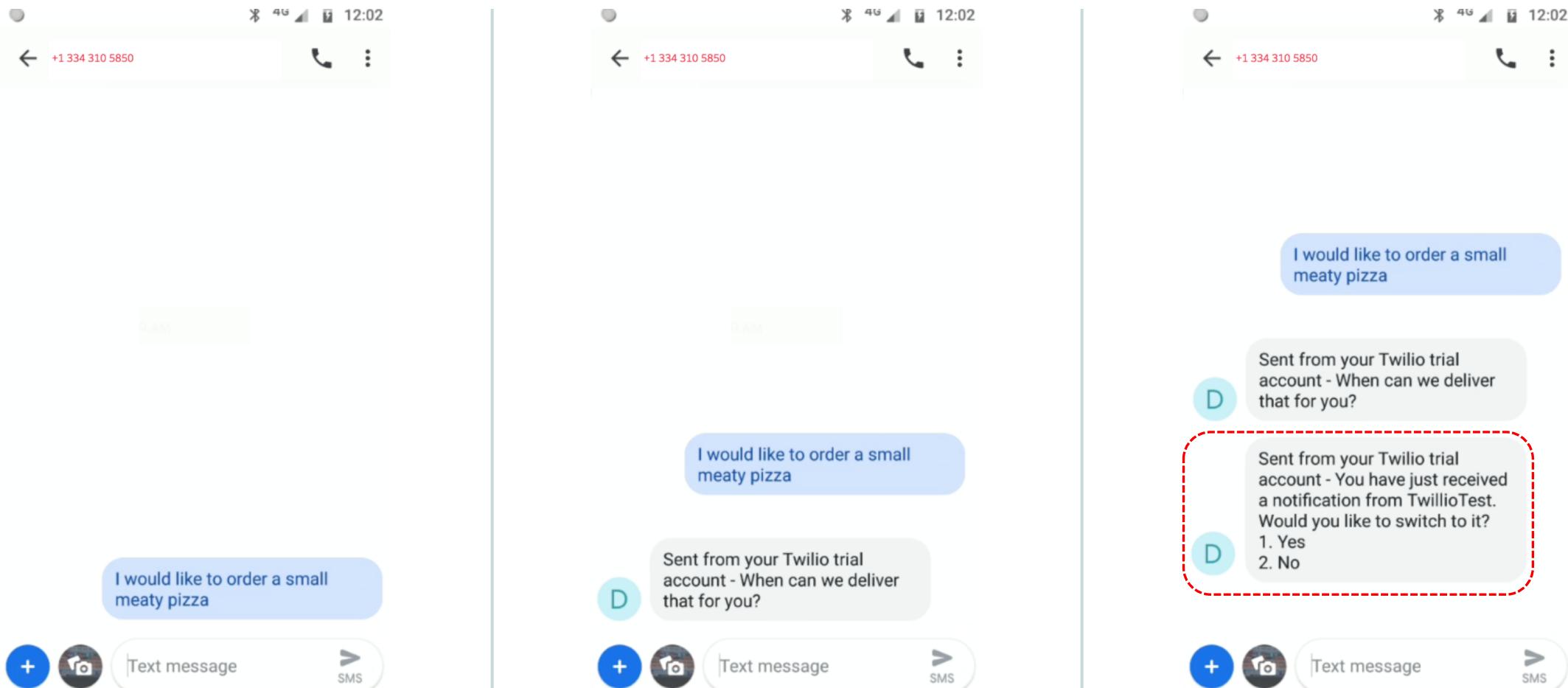
Users may OR may not be in a conversation when receiving a notification.



No active user conversation



Active user conversation



Topic agenda

- 1 ➤ About application initiated conversations
- 2 ➤ How application initiated conversation works
- 3 ➤ Setup
- 4 ➤ Channel configuration
- 5 ➤ Configure external application
- 6 ➤ How to test application initiated conversations

Actors involved

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 digital assistant
 - 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

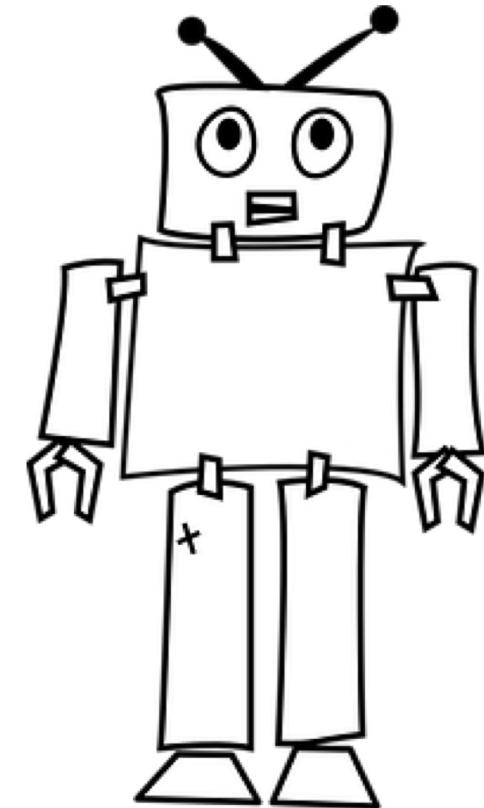
2. Oracle Digital Assistant

3. Twilio

4. SMS client
on user phone

- Sends/receives SMS messages to/from Twilio

**Application initiated conversation
need to be configured for the skill
and digital assistant.**



Topic agenda

- 1 ➤ About application initiated conversations
- 2 ➤ How application initiated conversation works
- 3 ➤ Setup
- 4 ➤ Channel configuration
- 5 ➤ Configure external application
- 6 ➤ How to test application initiated conversations

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 event with parameters

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 event with parameters
 - Payload documentation
 - <https://docs.oracle.com/en/cloud/paas/digital-assistant/use-chatbot/application-initiated-conversations.html#GUID-5DF067AA-E30B-4711-9C95-CEEE91F5F99B>

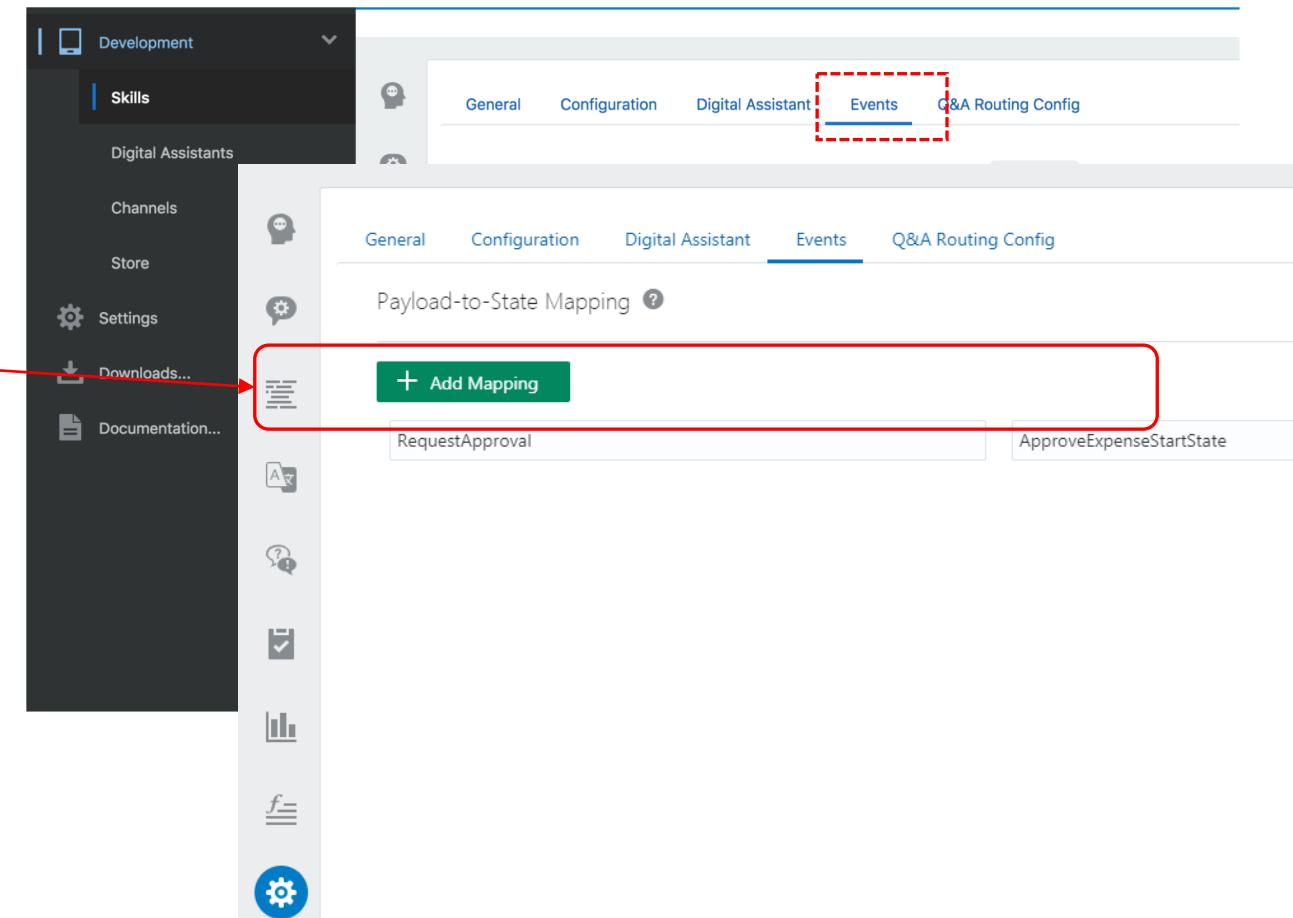
- Include these properties in the message payload:
 - `userId`—The actual phone number of the user. This is one of the numbers that are associated with the phone number that's both assigned to the Twilio account and used by the Twilio channel configuration.
 - `payloadType`—The name of the payload that's 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"  
    }  
  }  
}
```

Configure skill – add payload to state mapping

What does message payload look like?

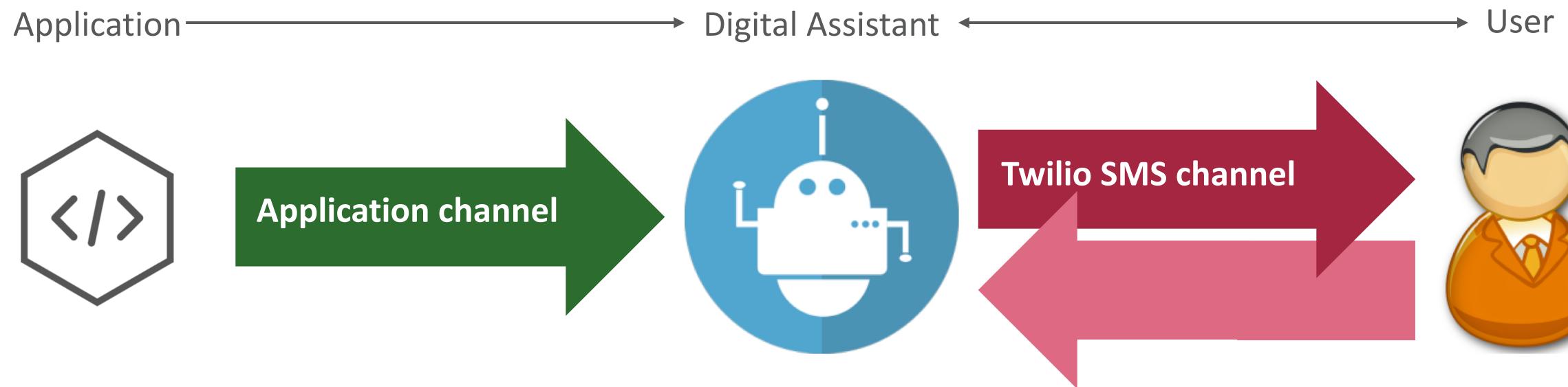
```
{  
  "userId": "+919871996112",  
  
  "messagePayload": {  
  
    "type": "application",  
  
    "payloadType": "RequestApproval",  
  
    "skillName": "ExpenseBot",  
  
    "channelName": "twilioSMS_ch",  
  
    "variables": {  
      "approvalType": "expense"  
    }  
  
  }  
}
```



Topic agenda

- 1 ➤ About application initiated conversations
- 2 ➤ How application initiated conversation works
- 3 ➤ Setup
- 4 ➤ Channel configuration
- 5 ➤ Configure external application
- 6 ➤ How to test application initiated conversations

Channels to setup



Application channel configuration

- Application channel exposes skill to external application
 - Results in URL and secret
 - External app uses it to “trigger” user conversation with a skill

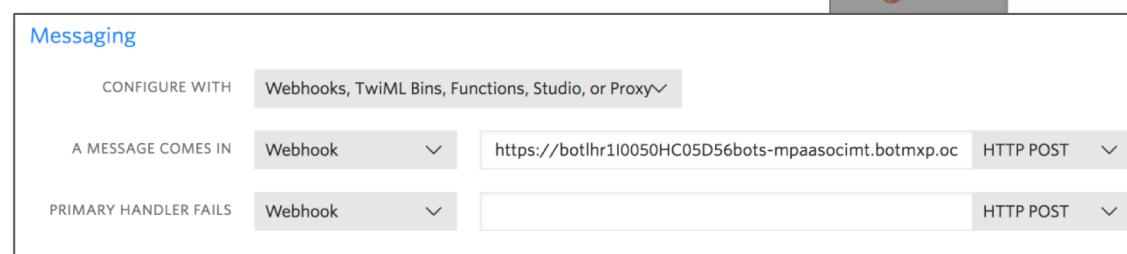
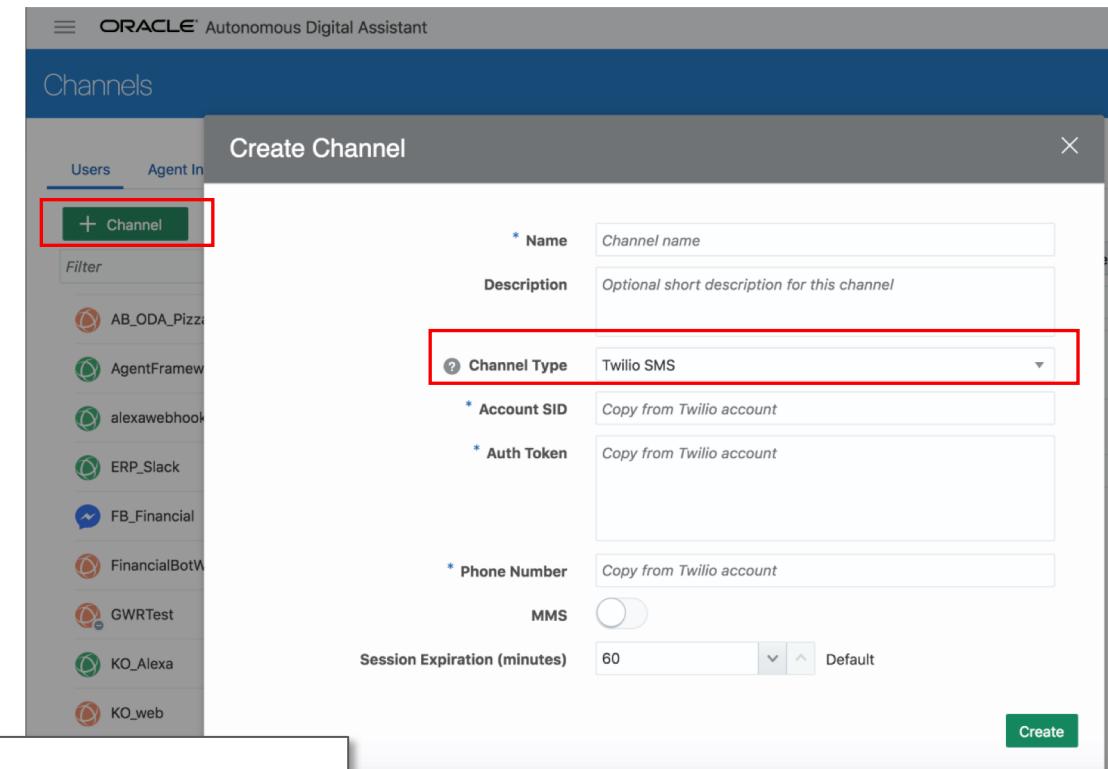
The screenshot shows the Oracle Autonomous Digital Assistant interface. The top navigation bar includes 'Development' (selected), 'Home', 'Development', 'Skills', 'Digital Assistants', and 'Channels'. The main menu tabs are 'Users', 'Agent Integrations', 'Applications' (selected), and 'System'. A sidebar on the left lists 'Application Configuration' (ExpenseBotAppChannel selected) and 'System'. The main content area displays configuration for 'ExpenseBotAppChannel':

- Application Enabled:** On
- Name:** ExpenseBotAppChannel
- Description:** Expense Management bot - Manage Approvals
- Outbound Application URL:** An outbound application URL you need to provide
- Use Authenticated User ID:** Off
- Secret Key:** nlmWG14QCnxEmesvgUacUS8v6KDNb22t (Regenerate button)
- Inbound URL:** https://botlhr1I0050HC05D56bots-mpaasocimt.botmvp.ocp.oraclecloud.com:443/connectors/v1/tenants/fdc5-6d466372210e4300bb3f1f4db15e8e96c/listeners/application/channels/c661114d-0f90-4d77-a191-3a037d8bd6e3

A red box highlights the 'Channels' link in the top navigation and the inbound URL field in the configuration form. To the right, there's a placeholder message: 'You haven't yet integrated a skill or digital assistant with this application. Create a configuration to integrate an external app with this application.' A 'Tell me more' link and a '+ Application Configuration' button are also present.

Twilio SMS channel

- User channel exposes skill to Twilio SMS service
 - Allows skill to communicate with user via SMS
 - Twilio account required
 - Provide Twilio account SID and token
- Provide webhook URL to Twilio
 - Added to "A Message Comes In" field

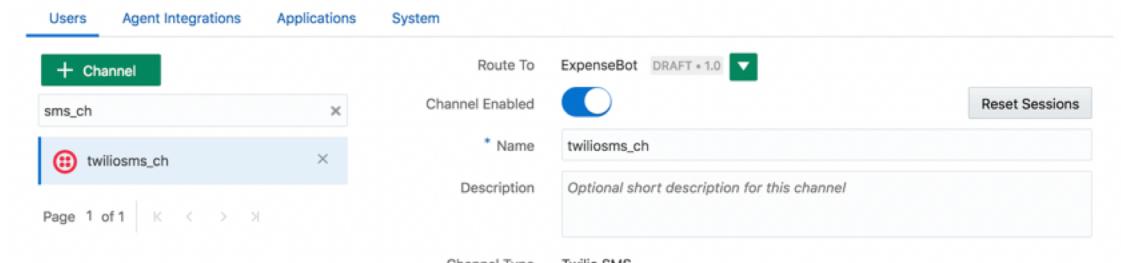
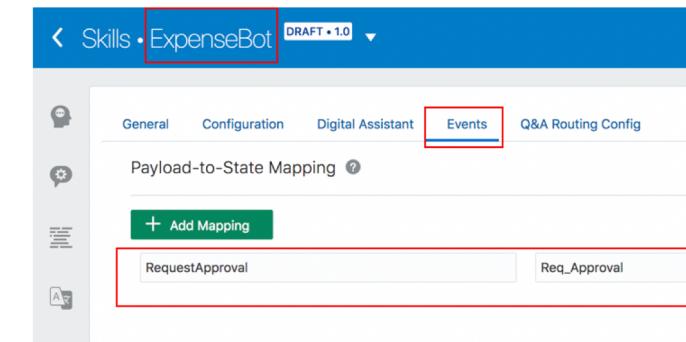
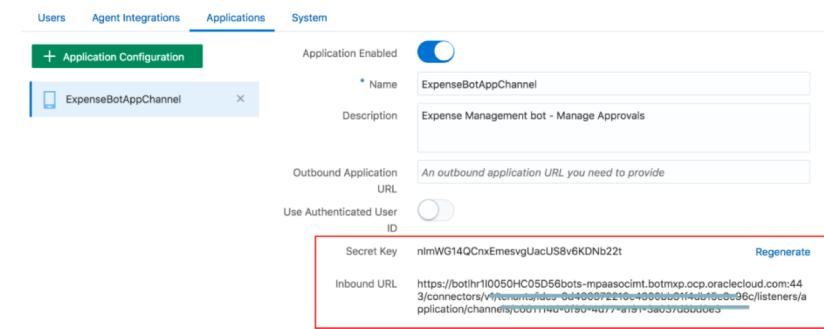


Topic agenda

- 1 ➤ About application initiated conversations
- 2 ➤ How application initiated conversation works
- 3 ➤ Setup
- 4 ➤ Channel configuration
- 5 ➤ Configure external application
- 6 ➤ How to test application initiated conversations

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
 - Digital assistant or skill name with version



External application inbound message

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

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

`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}}`

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 – Request accepted
- 403 Forbidden – X-Hub-Signature is incorrect
- 404 Not found – POST URL is incorrect

Topic agenda

- 1 ➤ About application initiated conversations
- 2 ➤ How application initiated conversation works
- 3 ➤ Setup
- 4 ➤ Channel configuration
- 5 ➤ Configure external application
- 6 ➤ How to test application initiated conversations

How to use embedded tester to test SMS

Get internal tester name

Channels

Users Agent Integrations Applications System

 System_Global_Test

Channel Enabled

Name **System_Global_Test**

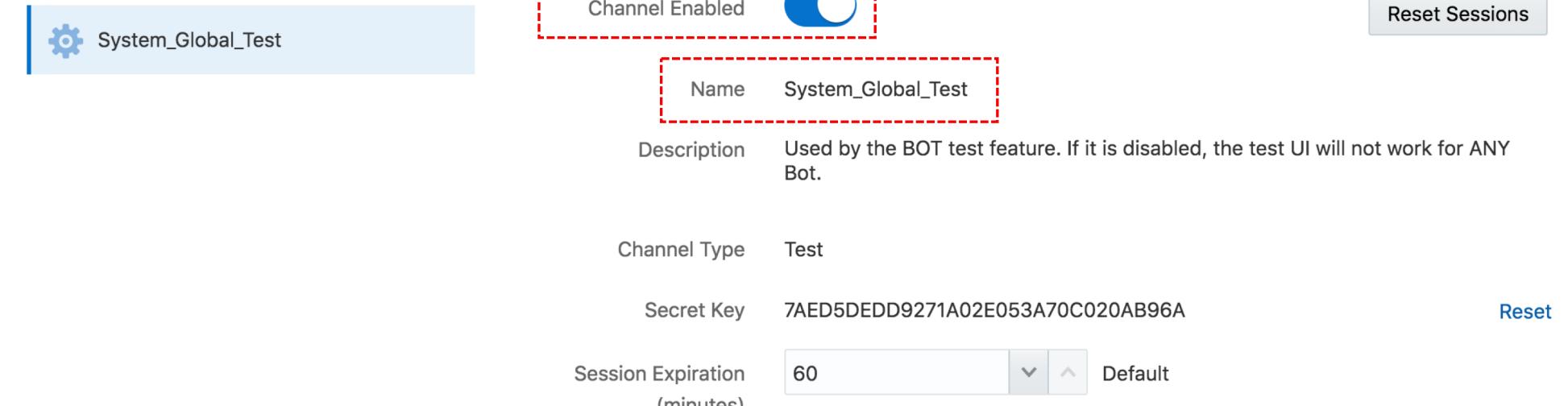
Description Used by the BOT test feature. If it is disabled, the test UI will not work for ANY Bot.

Channel Type Test

Secret Key 7AED5DEDD9271A02E053A70C020AB96A [Reset](#)

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

Reset Sessions



How to use embedded tester to test SMS

Find user ID for system test channel

The screenshot shows the FinanceAssist Tester interface. On the left, a conversation with a 'RetailBot' is displayed. The bot has sent a welcome message: "Welcome! Here are some things you can do:" followed by a list of options for Lululemon: "go to lululemon", "check my order in lululemon", and "return my order to lululemon". The user has responded with "check my order in lululemon". A follow-up message from the bot asks "What is your gift card number?". On the right, the 'Variables' section shows a 'system' variable. In the top right corner, a screenshot of a browser's developer tools Network tab is shown, filtered for XHR requests. It lists several responses, with one specific entry highlighted: "responses?_=1547789516676", which contains the JSON object: "1 line.routing.routerResponse"}]}}, "userId": "4076119"}]. This JSON object includes the user ID "4076119".

How to use embedded tester to test SMS

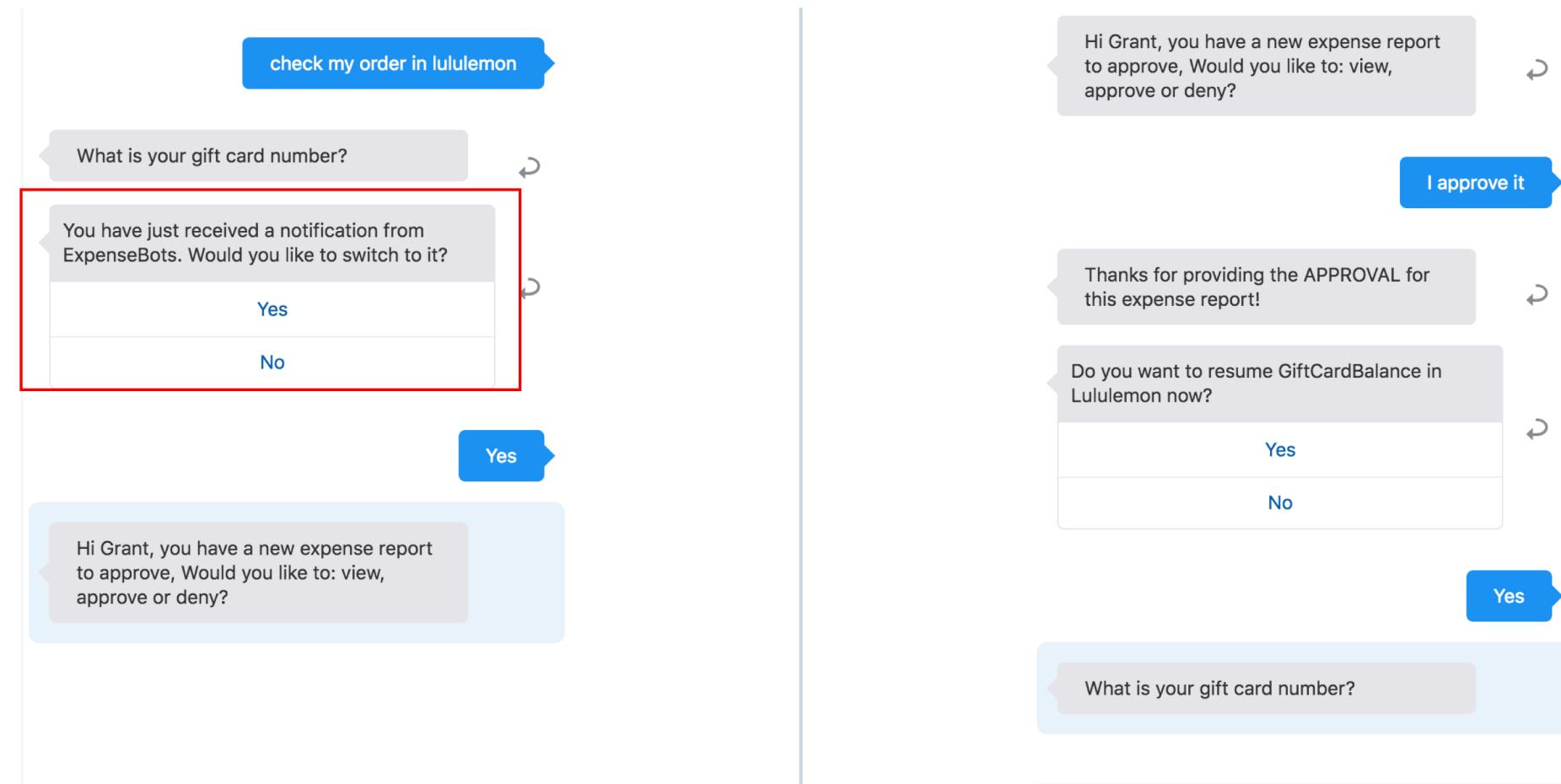
Message payload

The screenshot shows the Postman application interface for testing an API. The request method is set to POST, and the URL is <https://botlhr1I0050HC05D56bots-mpaasocimt.botmxp.ocp.oraclecloud.com:443/connectors/v1/tenan...>. The 'Body' tab is selected, showing a raw JSON payload. The payload content is:

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

A red dashed box highlights the 'userId' field value '4076119'. The 'Headers' tab shows two entries: 'Content-Type: application/json' and 'User-Agent: PostmanRuntime/7.15.0'. The 'Params' tab is empty. The 'Send' button is blue, and the 'Save' button has a dropdown arrow. Below the request area, the status bar shows 'Status: 202 Accepted', 'Time: 756 ms', and 'Size: 273 B'. The bottom navigation bar includes tabs for Body, Cookies, Headers (6), and Test Results.

How to use embedded tester to test SMS



Integrated Cloud Applications & Platform Services

ORACLE®



Oracle Digital Assistant Hands-On