

Chatbot Configuration Guide  
Oracle Banking Digital Experience  
Patchset Release 22.2.6.0.0

Part No. F72987-01

April 2025

## Chatbot Configuration Guide

April 2025

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway

Goregaon (East)

Mumbai, Maharashtra 400 063

India

Worldwide Inquiries:

Phone: +91 22 6718 3000

Fax: +91 22 6718 3001

[www.oracle.com/financialservices/](http://www.oracle.com/financialservices/)

Copyright © 2006, 2025, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

---

## Table of Contents

<b>1. Preface .....</b>	<b>1-1</b>
1.1 Purpose .....	1-1
1.2 Audience .....	1-1
1.3 Documentation Accessibility .....	1-1
1.4 Critical Patches .....	1-1
1.5 Diversity and Inclusion .....	1-1
1.6 Conventions .....	1-1
1.7 Screenshot Disclaimer .....	1-2
1.8 Acronyms and Abbreviations .....	1-2
<b>2. Purpose .....</b>	<b>2-1</b>
<b>3. Topology .....</b>	<b>3-1</b>
<b>4. Common Configurations .....</b>	<b>4-1</b>
4.1 ODA Configurations .....	4-1
<b>5. Configuring Channels in ODA .....</b>	<b>5-1</b>
5.1 Web Channel .....	5-1
5.2 Android Channel .....	5-2
5.3 iOS Channel .....	5-2
5.4 Facebook Channel .....	5-4
5.5 Alexa Channel .....	5-8
5.6 WhatsApp Channel .....	5-19
<b>6. Configuration in OBDX .....</b>	<b>6-1</b>
<b>7. Standalone WhatsApp/Facebook messenger integration with OBDX .....</b>	<b>7-1</b>

---

# 1. Preface

## 1.1 Purpose

Welcome to the User Guide for Oracle Banking Digital Experience. This guide explains the operations that the user will follow while using the application.

## 1.2 Audience

This manual is intended for Customers and Partners who setup and use Oracle Banking Digital Experience.

## 1.3 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### **Access to Oracle Support**

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit, <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## 1.4 Critical Patches

Oracle advises customers to get all their security vulnerability information from the Oracle Critical Patch Update Advisory, which is available at [Critical Patches, Security Alerts and Bulletins](#). All critical patches should be applied in a timely manner to ensure effective security, as strongly recommended by [Oracle Software Security Assurance](#).

## 1.5 Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

## 1.6 Conventions

The following text conventions are used in this document:

Convention	Meaning

<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>Italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

## 1.7 Screenshot Disclaimer

The images of screens used in this user manual are for illustrative purpose only, to provide improved understanding of the functionality; actual screens that appear in the application may vary based on selected browser, theme, and mobile devices.

## 1.8 Acronyms and Abbreviations

The list of the acronyms and abbreviations that you are likely to find in the manual are as follows:

Abbreviation	Description
<b>OBDX</b>	Oracle Banking Digital Experience

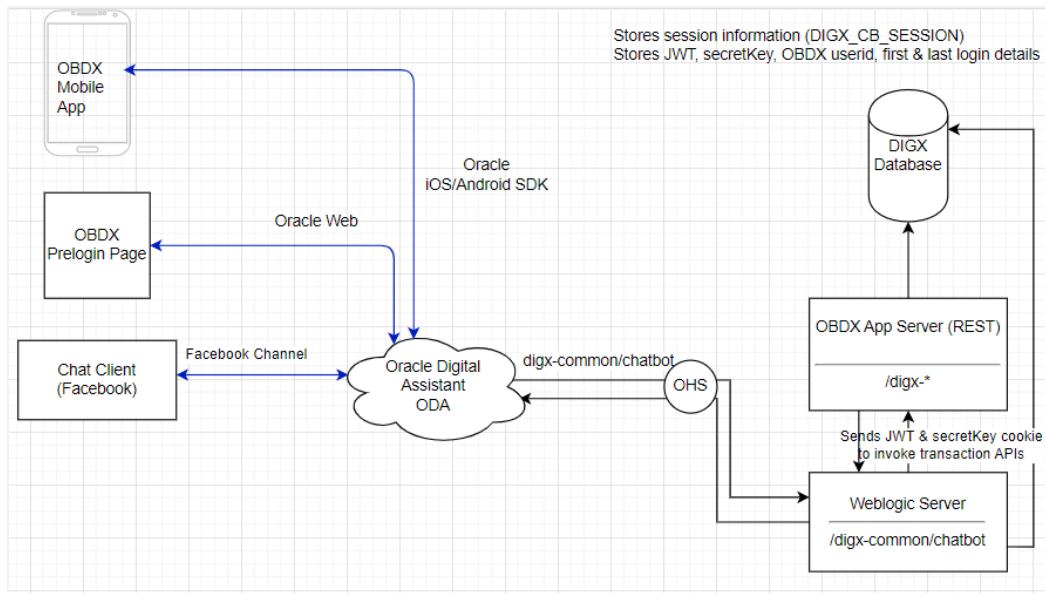
---

## 2. Purpose

OBDX provides interface for Chatbot module, integrated with Oracle Digital Assistant (ODA) out of the box. It provides end users a chat interface to interact with the bank. Transactions like balance enquiry, fund transfers to payees, enquiring about banking products and details of ATM/Branches can be achieved through chat. This document provides steps to setup OBDX chatbot module with ODA. The prerequisites include:

- ODA setup on the ODA console and required channel setup.
- Respective channel setup (IOS, Android and Web) to connect to ODA is specified in the respective Technical guide.

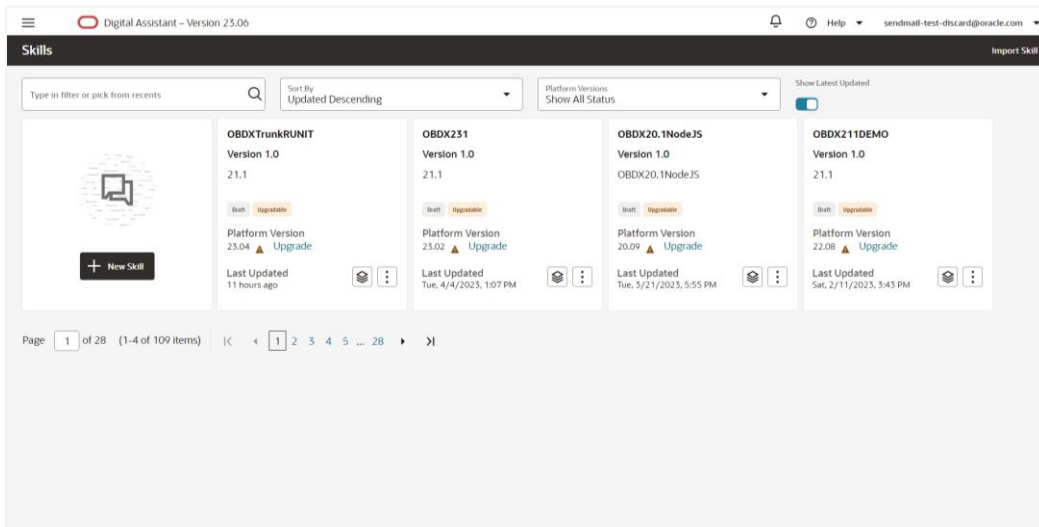
### 3. Topology



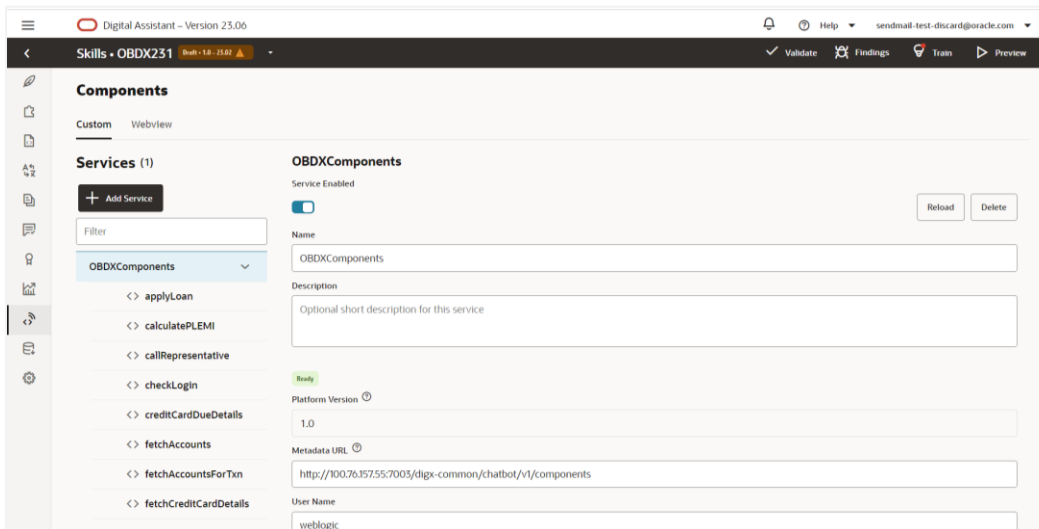
## 4. Common Configurations

### 4.1 ODA Configurations

1. Login to ODA and import the OBDX bot shipped with OBDX installer. There is below folder OBDX\_Installer/installables/OBDX/BASE/<OBDX patchset version>/chatbot/config/oda-text/. Zip the content inside this and Import the zip by clicking the “Import Bot” on ODA dashboard.



2. Click on the OBDX Bot and click on the components to add the custom components.



3. Put the OBDX URL here. The OBDX setup and the ODA setup must be accessible over Internet.
4. Add username/password (in HTTP Basic authorization) of any user with Administrators role which can be used to login in OBDX WebLogic server.



5. To configure intent threshold for the skill, go to settings tab in a bot and click configurations to configure threshold frequency (default 0.5) as shown below:

The screenshot shows the 'Settings' page for a skill named 'OBDX231'. The 'Configuration' tab is selected. Under 'System Parameters', the 'Confidence Threshold' is set to 0.5. Other parameters like 'Confidence Win Margin' are set to 0. Error prompts are configured with system-generated text. A URL is visible at the bottom of the page.

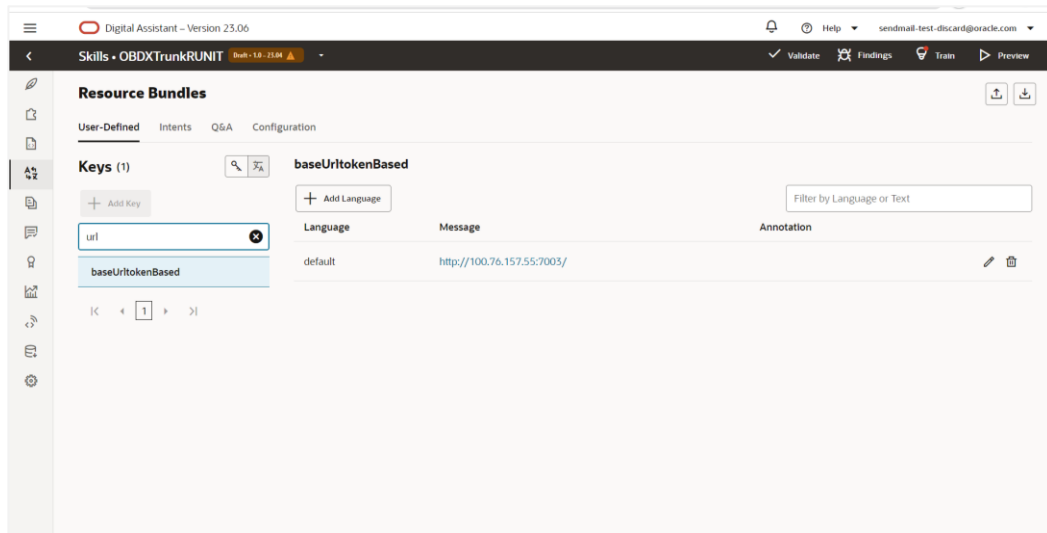
Parameter	Value
Confidence Threshold	0.5
Confidence Win Margin	0
Unexpected Error Prompt	\$(rb('systemConfiguration_errorUnexpectedErrorPrompt'))
Max States Exceeded Error Prompt	\$(rb('systemConfiguration_errorMaxStatesExceededPrompt'))
Expired Session Error Prompt	\$(rb('systemConfiguration_errorExpiredSessionPrompt'))
OAuth Cancel Prompt	\$(rb('systemConfiguration_oauthCancelPrompt'))
OAuth Success Prompt	

6. Configure required entity in a resource bundles (one created/configured in OBDX). At a time chatbot works with single entity. Go to Resource Bundles as shown below-

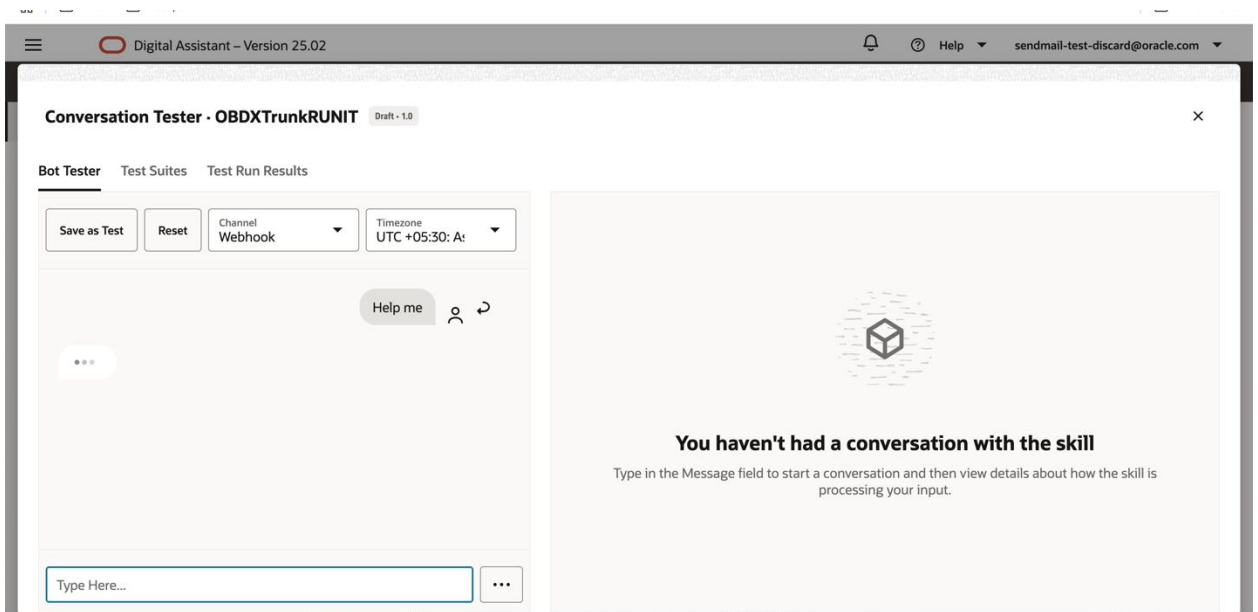
The screenshot shows the 'Resource Bundles' page for the 'OBDX231' skill. The 'User-Defined' tab is selected. A key named 'entity' is added. The 'DefaultHomeEntity' bundle is shown with a table containing one entry for the 'default' language with the message 'OBDX\_BU'.

Language	Message	Annotation
default	OBDX_BU	

7. Configure OBDX managed server URL where OBDX rest Apis are hosted –



8. Login to ODA -> Preview



## 5. Configuring Channels in ODA

OBDX bot can be configured with Oracle Web, Oracle Android and Oracle iOS channels.

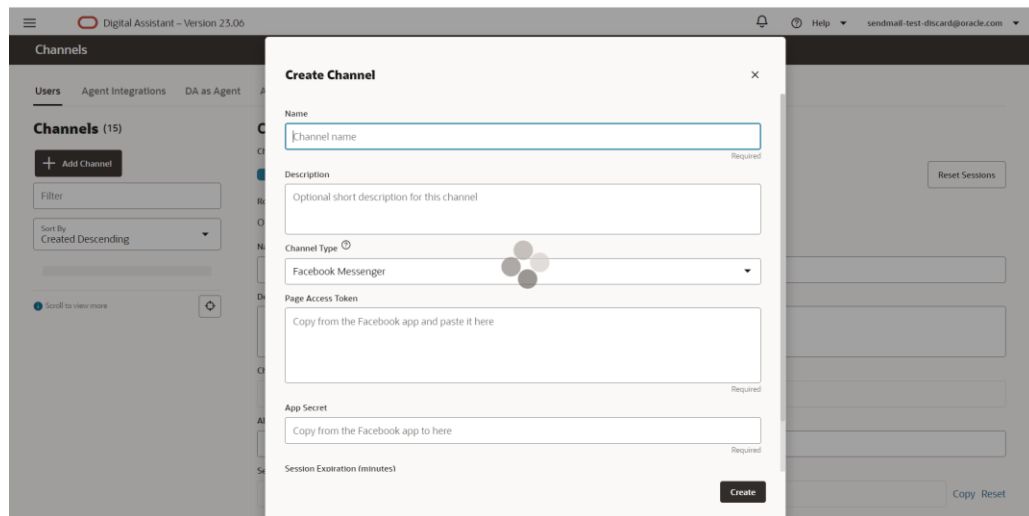
Chatbot widget appears on pre-login page in OBDX UI where the login is with mobile number.(OTP or PIN or Soft Token)

For post login, the browser session is automatically used

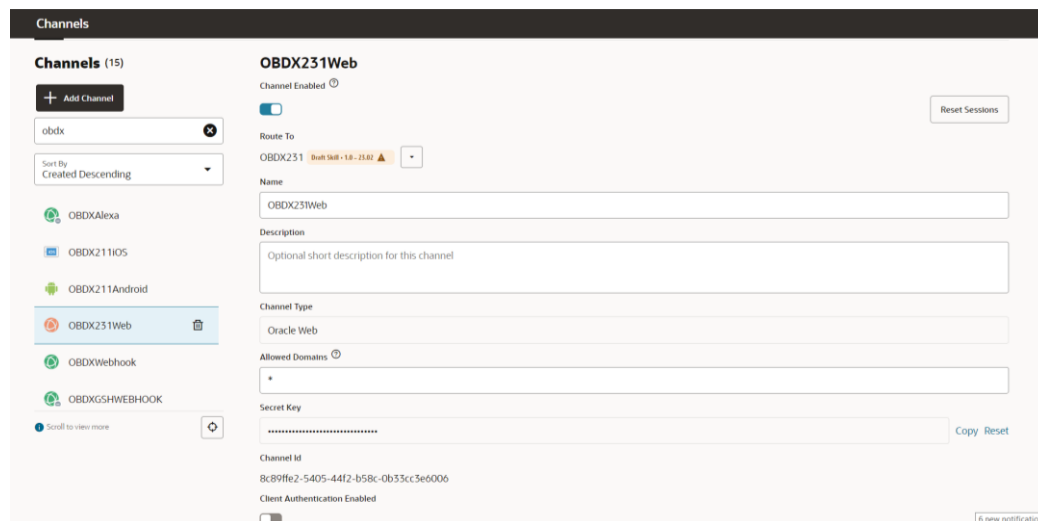
### 5.1 Web Channel

To configure this channel:

1. Choose Development ->Channels-> Users from the menu. Click **Add Channel**.

The screenshot shows the 'Create Channel' modal window in the ODA interface. The modal has a 'Name' field with a placeholder 'Channel name', a 'Description' field with a placeholder 'Optional short description for this channel', a 'Channel Type' dropdown menu currently set to 'Facebook Messenger', a 'Page Access Token' field with a placeholder 'Copy from the Facebook app and paste it here', an 'App Secret' field with a placeholder 'Copy from the Facebook app to here', and a 'Session Expiration (minutes)' field. There are 'Required' labels next to the Name, Description, Page Access Token, App Secret, and Session Expiration fields. A 'Create' button is at the bottom right of the modal. The background shows the 'Channels' list with 15 items and an 'Add Channel' button.

2. Create Oracle Web as the channel type.

The screenshot shows the 'Channels' list on the left and the configuration for the 'OBDX231Web' channel on the right. The 'Channels' list has a search bar with 'obdx' and a 'Sort By Created Descending' dropdown. The 'OBDX231Web' channel is selected. The configuration for 'OBDX231Web' includes a 'Channel Enabled' toggle switch, a 'Route To' dropdown set to 'OBDX231', a 'Name' field with 'OBDX231Web', a 'Description' field with 'Optional short description for this channel', a 'Channel Type' dropdown set to 'Oracle Web', an 'Allowed Domains' field with a placeholder '\*', a 'Secret Key' field with a placeholder '\*\*\*\*\*', a 'Channel Id' field with '8c89ffe2-5405-44f2-b58c-0b33cc3e6006', and a 'Client Authentication Enabled' toggle switch. There are 'Copy' and 'Reset' buttons next to the 'Secret Key' field. A 'Reset Sessions' button is at the top right. A notification bar at the bottom right says '6 new notifications'.

3. Route the channel to your skill or digital assistant.
4. Switch On Channel Enabled switch

5. Add above Chatbot URL and channelled in /ui/framework/js/configurations/config.js in oda tag

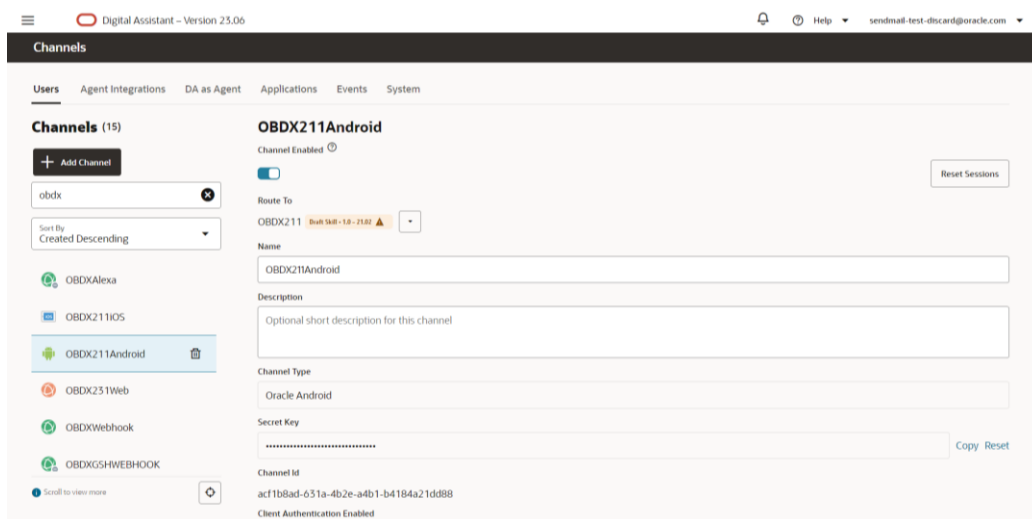
#### To remove this chatbot widget

- Remove URL & channelled in above file
- Remove the “chat-bot” tag in pre-login dashboard → DIGX\_DB\_DASHBOARD → DASHBOARDDESIGN blob for anonymous dashboard

## 5.2 Android Channel

#### To configure this channel:

1. Choose Development->Channels->Users from the menu. Click **Add Channel**.
2. Then add Oracle Android as the channel type.



3. Route the channel to your skill or digital assistant.
4. Switch On Channel Enabled switch.
5. Note the channel Id. This is required in application configuration. Refer application configuration guide for more details.

## 5.3 iOS Channel

#### To configure this channel:

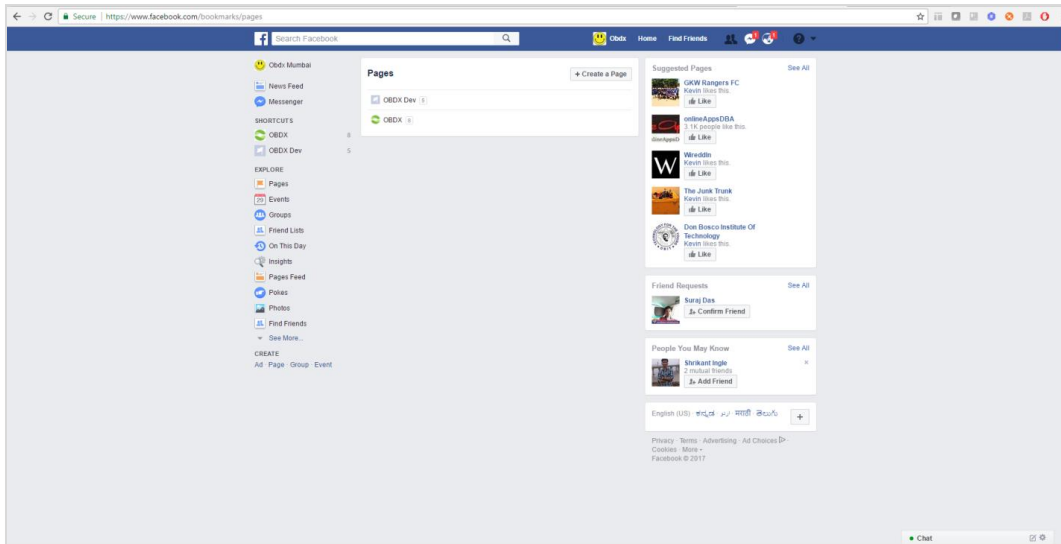
1. Choose Development->Channels->Users from the menu. Click Add Channel.
2. Add Oracle iOS as the channel type.

The screenshot displays the Oracle Channels management console. On the left, a sidebar titled 'Channels (15)' contains a search bar with 'obdx', a 'Sort By Created Descending' dropdown, and a list of channel types: OBDXAlexa, OBDX211iOS (selected), OBDX211Android, OBDX231Web, OBDXWebhook, and OBDXGSHWEBHOOK. The main area shows the configuration for the 'OBDX211iOS' channel. It includes a 'Channel Enabled' toggle switch (checked), a 'Route To' dropdown set to 'OBDX231', and a 'Reset Sessions' button. The 'Name' field is 'OBDX211iOS', and the 'Description' field contains the placeholder 'Optional short description for this channel'. The 'Channel Type' is 'Oracle iOS'. The 'Secret Key' field is masked with asterisks, with 'Copy' and 'Reset' buttons. The 'Channel Id' is 'a5187fd6-d182-4ff4-a74d-7050f44382df'. At the bottom, there are two more toggle switches: 'Client Authentication Enabled' (unchecked) and 'Push Notification Enabled' (unchecked).

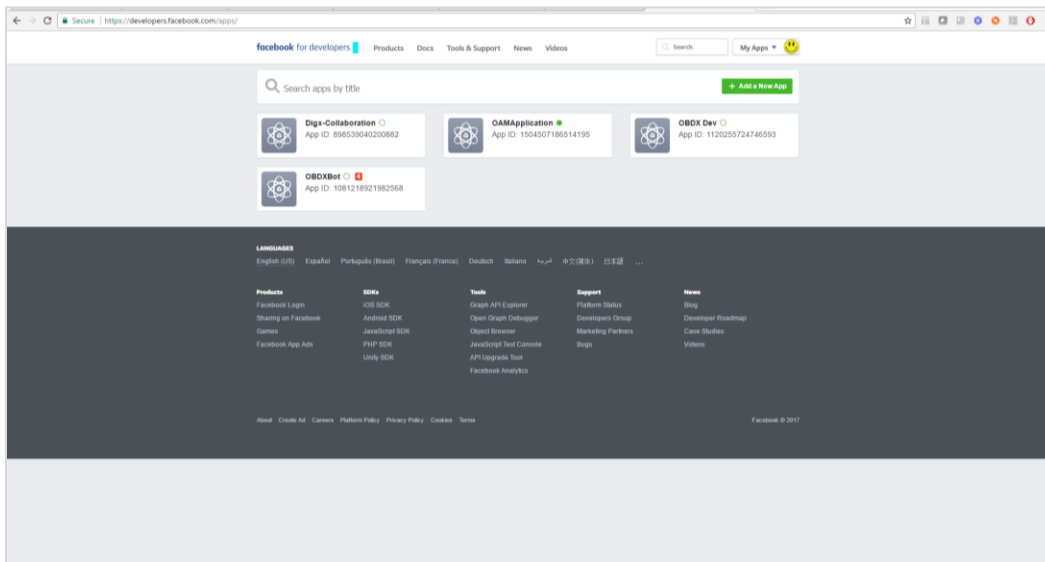
3. Route the channel to your skill or digital assistant.
4. Switch On Channel Enabled switch.
5. Note the channel Id. This is required in application configuration. Refer application configuration guide for more details.

## 5.4 Facebook Channel

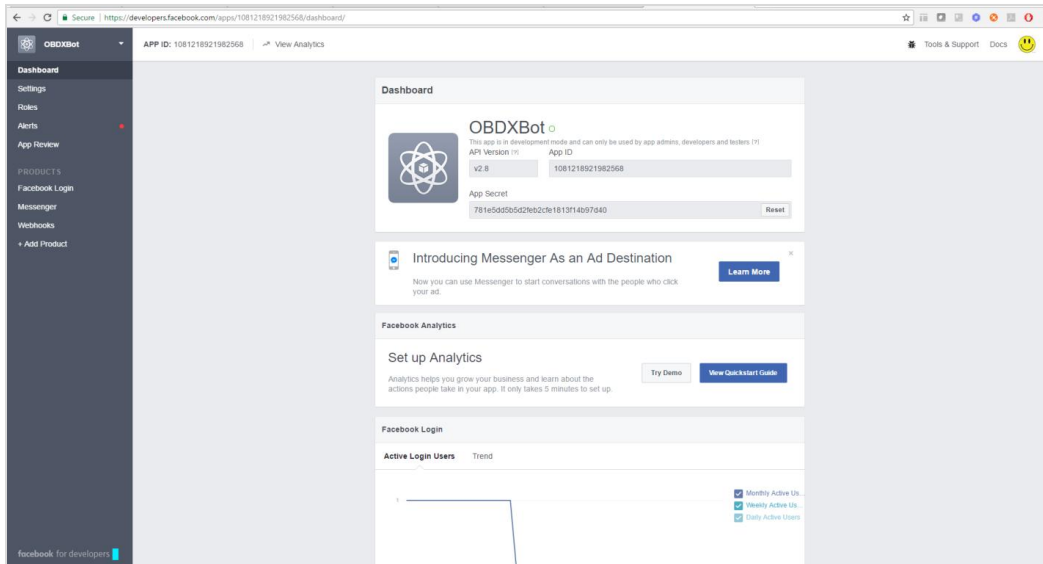
1. Create a Facebook account for the Bank. Login to Facebook with credentials.
2. Create a new page.



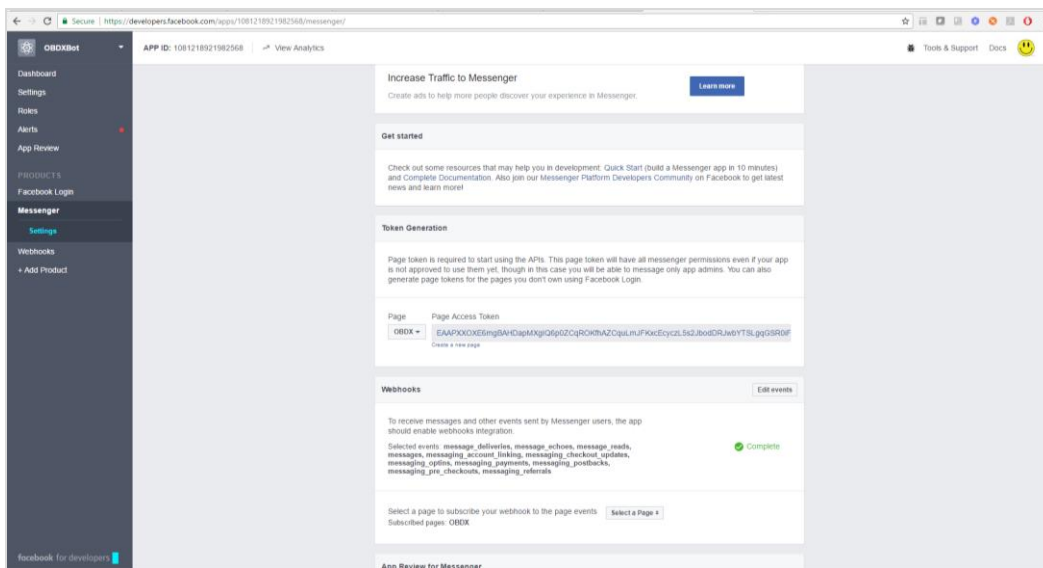
3. Navigate to developer link and create an application as shown below



4. Navigate to dashboard page and note the app secret as it will require in future steps.

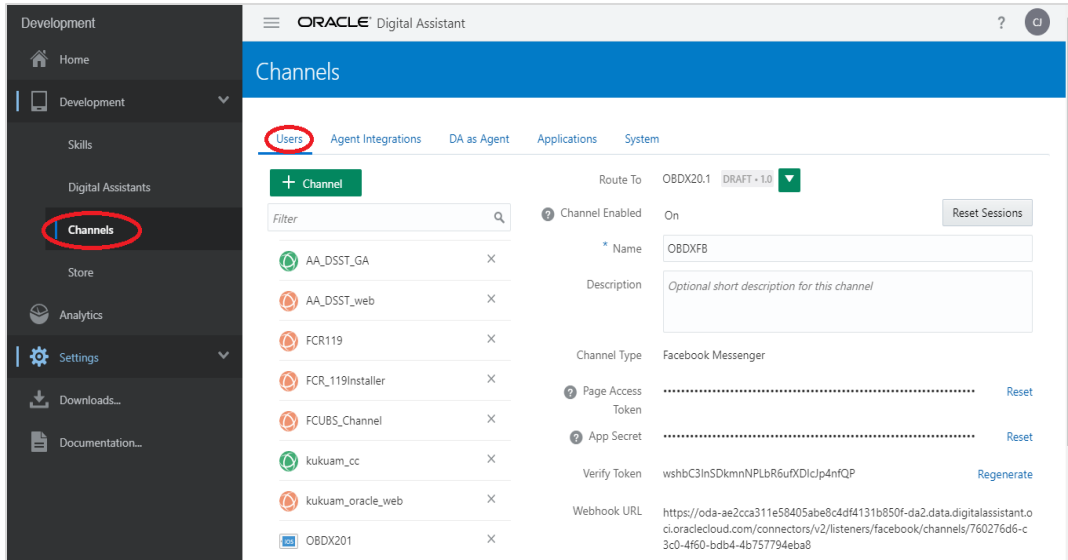


5. Navigate to Messenger > Settings page from left panel and in token generation section select the page created previously. Note the page access token.



## Create Facebook Channel in ODA

1. In ODA ,click channels in the left panel/menu and then click on users.



2. Next, click **Add Channel** to open the Create Channel dialog.
3. Choose Facebook Messenger as the channel type.

The 'Create Channel' dialog box is shown. It contains the following fields and values:

- Name:** OBDXFB
- Description:** Optional short description for this channel
- Channel Type:** Facebook Messenger
- Page Access Token:** Copy from the Facebook app and paste it here
- App Secret:** Copy from the Facebook app to here
- Session Expiration (minutes):** 60

A green 'Create' button is located at the bottom right of the dialog.

4. In the Page Access Token field, paste the page access token that you generated previously in the Set Up Facebook Messenger procedure.
5. In the App Secret field, paste the app secret that you copied previously in the Set Up Facebook Messenger procedure and click Create.
6. In the Channels page, copy both the Verify Token and Webhook URL and paste them somewhere convenient on your system. You'll need these to configure the Facebook webhook.



Route To: OBDX20.1 DRAFT - 1.0

Channel Enabled: ☒ On Reset Sessions

Name: OBDXFB

Description: *Optional short description for this channel*

Channel Type: Facebook Messenger

Page Access Token: ..... Reset

App Secret: ..... Reset

Verify Token: wshbC3lnSDkmnNPLbR6ufXDIclp4nfQP Regenerate

Webhook URL: https://oda-ae2cca311e58405abe8c4df4131b850f-da2.data.digitalassistant.oci.oraclecloud.com/connectors/v2/li  
steners/facebook/channels/760276d6-c3c0-4f60-bdb4-4b757794eba8

Session Expiration (minutes): 60 Default

7. Select the digital assistant or skill that you want to associate with the channel. Switch on the Channel Enabled control to enable it.

### Configure the Facebook Messenger Webhook

1. In Facebook Messenger, be sure that you've selected the project that you initially created for the webhook.
2. Click **Messenger** and then choose Settings.
3. Click **Subscribe to Events** to open the New Page Subscription dialog.
4. Copy the Webhook URL that you got from the Digital Assistant Channels page and paste it in the Callback URL field in the New Page Subscription dialog.
5. Copy the Verify Token generated by Digital Assistant and paste it into the Verify Token field.

Webhooks Learn More

Page

**Edit Page Subscription**

Callback URL: http://633b4764.ngrok.io/connectors/v1/tenants/5c12a114-e200-4563-b6a2-8ca369c09160/steners/facebook/channels

Verify Token: Token that Facebook will echo back to you as part of callback URL verification

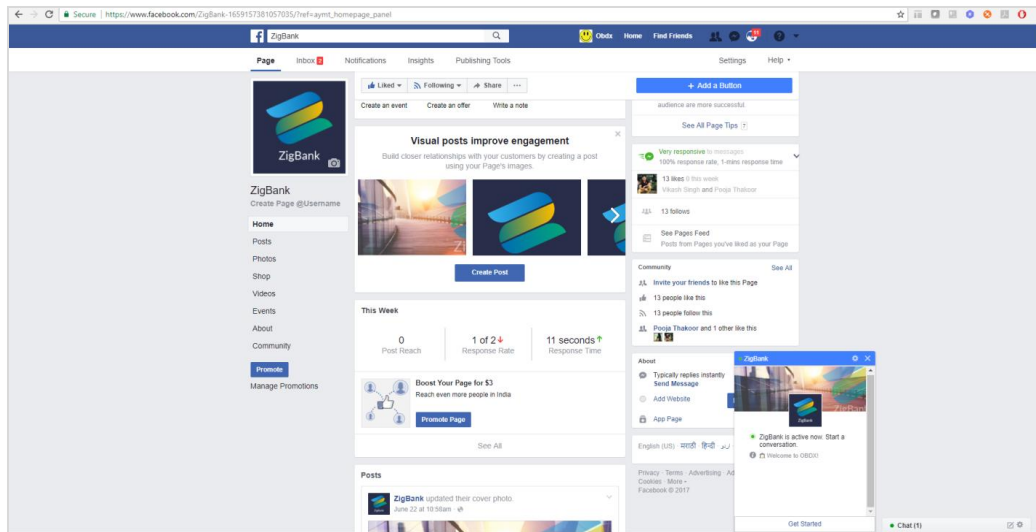
Cancel Remove Subscription Verify and Save

Field	Version	Test	Subscribe
bio	v2.0	Test	Subscribe
birthday	v2.0	Test	Subscribe
category	v2.0	Test	Subscribe
checkins	v2.0	Test	Subscribe
company_overview	v2.0	Test	Subscribe
conversations	v2.0	Test	Subscribe
country_team	v2.0	Test	Subscribe
current_location	v2.0	Test	Subscribe
description	v2.0	Test	Subscribe

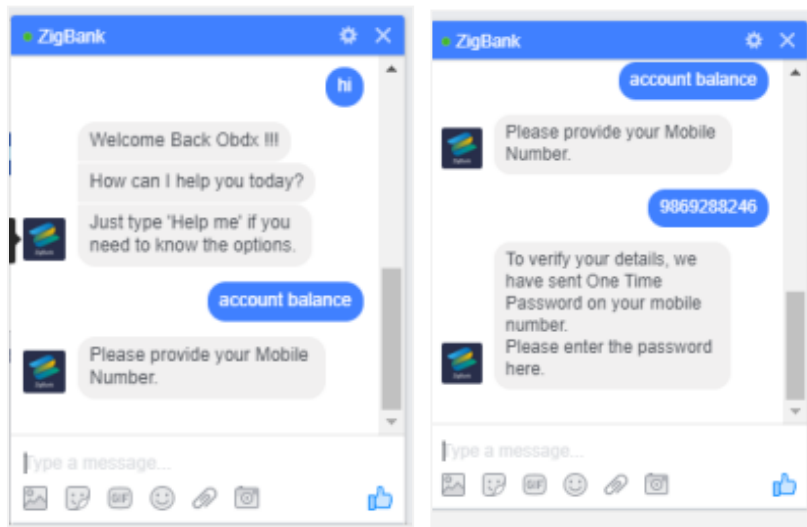
6. Under Subscription Fields, select the messages and messaging\_postbacks callback events. The messages event is triggered whenever someone sends a message to your Facebook page.
7. Click Verify and Save.
8. In the Webhooks section of the Messenger settings, select the Facebook page for your digital assistant (or standalone skill). Click **Subscribe**.

## Validating configurations

1. Login to Facebook > Navigate to the page and click > Send message



2. Click **Get Started** in the chat window > You should receive welcome message from ODA
3. Enquire about account balance > OTP should be received on the registered email address of the party in core banking.



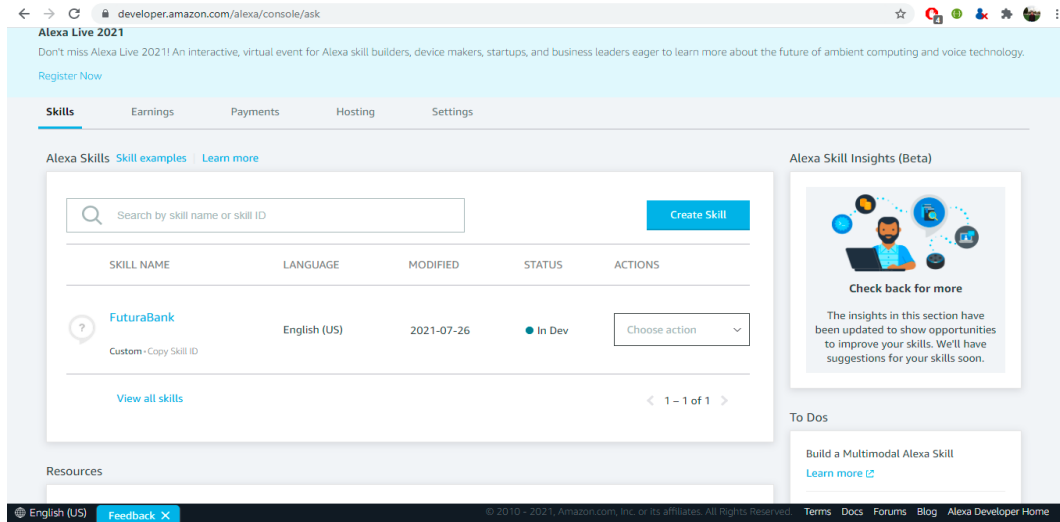
## 5.5 Alexa Channel

Creating an Alexa skill called *Futura Bank* along with a Webhook channel lets you chat with a specific bot.

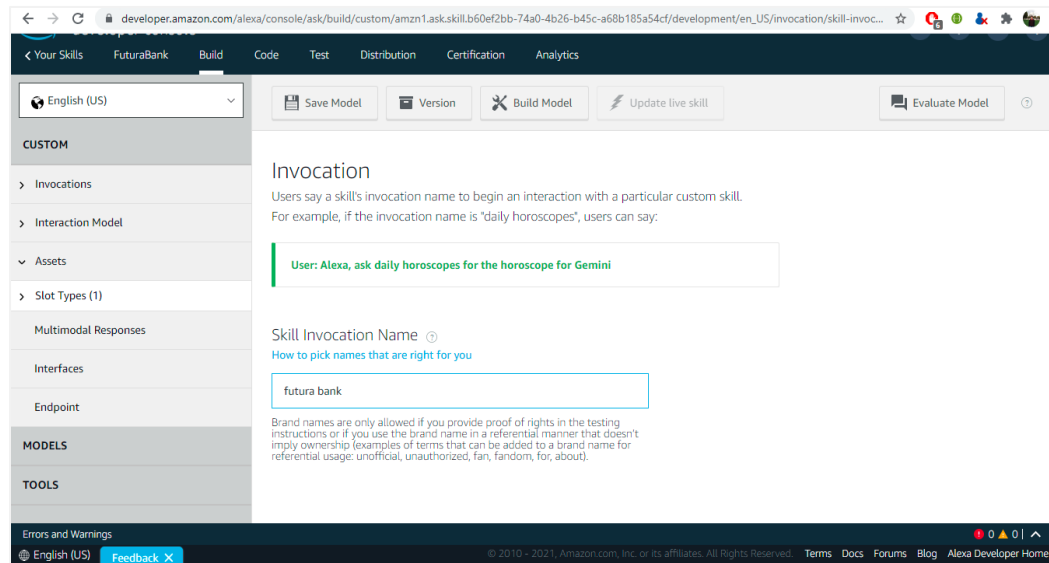
Add the skill information

Set up a developer account in the Amazon Developer Portal.

1. Open the **Amazon Developer Console**.
2. Click on '**Create Skill**'



3. Enter Futura Bank (or any name that you want to use to invoke this skill) as the Invocation Name.



4. Add new intent as CommandBot.

developer console

English (US)

Save Model Version Build Model Update live skill Evaluate Model

**CUSTOM**

- Invocations
- Interaction Model
- Intents (5)**
  - CommandBot
    - command
  - Built-In Intents (5)
    - AMAZON.CancelIntent
    - AMAZON.HelpIntent
    - AMAZON.StopIntent
    - AMAZON.NavigateHomeIntent
    - AMAZON.FallbackIntent

**Alexa Conversations**  
Expand your skill's features with our new deep learning-based dialog manager to create natural multi-turn conversations. \*Alexa Conversations only works for English (US) skills.  
[Enable Alexa Conversations](#) | [What is Alexa Conversations?](#)

### Intents

+ Add Intent Skill Model Sensitivity Recommended Filter intents

NAME	UTTERANCES	SLOTS	TYPE	ACTIONS
AMAZON.CancelIntent	-	-	Required	<a href="#">Edit</a>

Errors and Warnings

English (US) Feedback

© 2010 - 2021, Amazon.com, Inc. or its affiliates. All Rights Reserved. Terms Docs Forums Blog Alexa Developer Home

## 5. Add sample utterance to it.

Intents / CommandBot

### Sample Utterances (1)

What might a user say to invoke this intent?

[command]

Dialog Delegation Strategy

Dialog management is not enabled for this intent. Why is this disabled?

### Intent Slots (1)

ORDER	NAME	SLOT TYPE	MULTI-VALUE	ACTIONS
1	command	CustomSlot	<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Delete</a>

Errors and Warnings

English (US) Feedback

© 2010 - 2021, Amazon.com, Inc. or its affiliates. All Rights Reserved. Terms Docs Forums Blog Alexa Developer Home

## 6. Create custom slot.

Slot Types

Slot types define how phrases in utterances are recognized and handled as well as the type of data passed between components. In Interaction Model, all intent slots must be assigned a slot type. In Alexa Conversations, all slots, arguments, response types and variables must be assigned a slot type. [Learn more](#) about using Slot Types and [learn more](#) about using Slot Types with Alexa Conversations.

[+ Add Slot Type](#)

NAME	SLOT VALUES	SLOT TYPE	ACTIONS
CustomSlot	2	Custom with values	<a href="#">Edit</a>   <a href="#">Delete</a>

< 1 - 1 of 1 Slot Types >

Slot Types / CustomSlot

Custom slot types with values define a representative list of possible values, IDs and synonyms.

[Bulk Edit](#) [Export](#)

Enter a new value for this slot type

[anything](#) [do anything](#) [something else](#) [thing](#) [do that](#) [really](#) [things](#) [do it](#) [these things](#) [i mean](#)

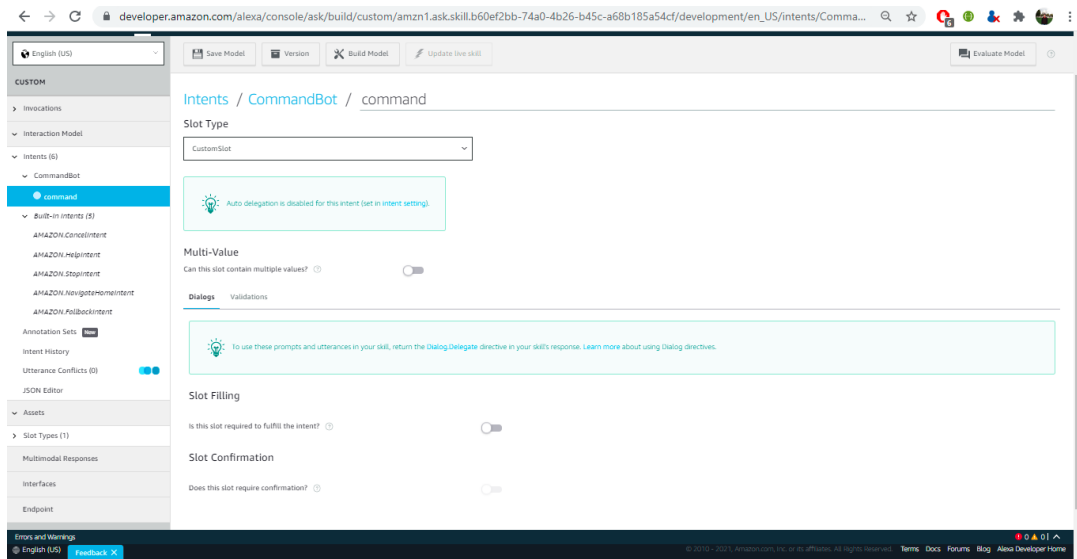
VALUE	ID (OPTIONAL)	SYNONYMS (OPTIONAL)	
something	<input type="text" value="Enter ID"/>	<input type="text" value="Add synonym"/>	+
do something	<input type="text" value="Enter ID"/>	<input type="text" value="Add synonym"/>	+

< 1 - 2 of 2 >

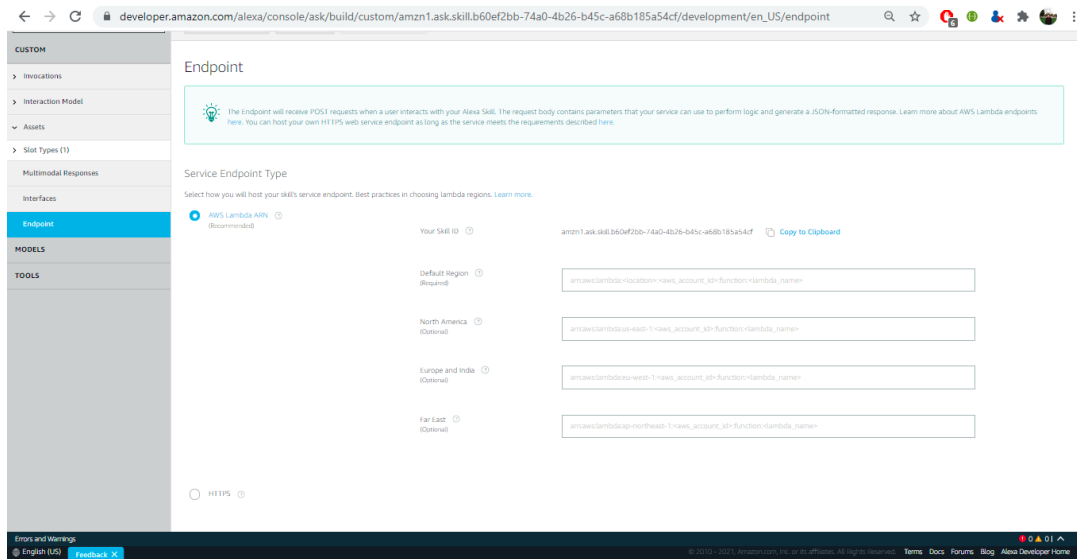
Slots Using (CustomSlot)

SLOT NAME	INTENT
command	CommandBot

7. Update this CustomSlot in previously added intent.

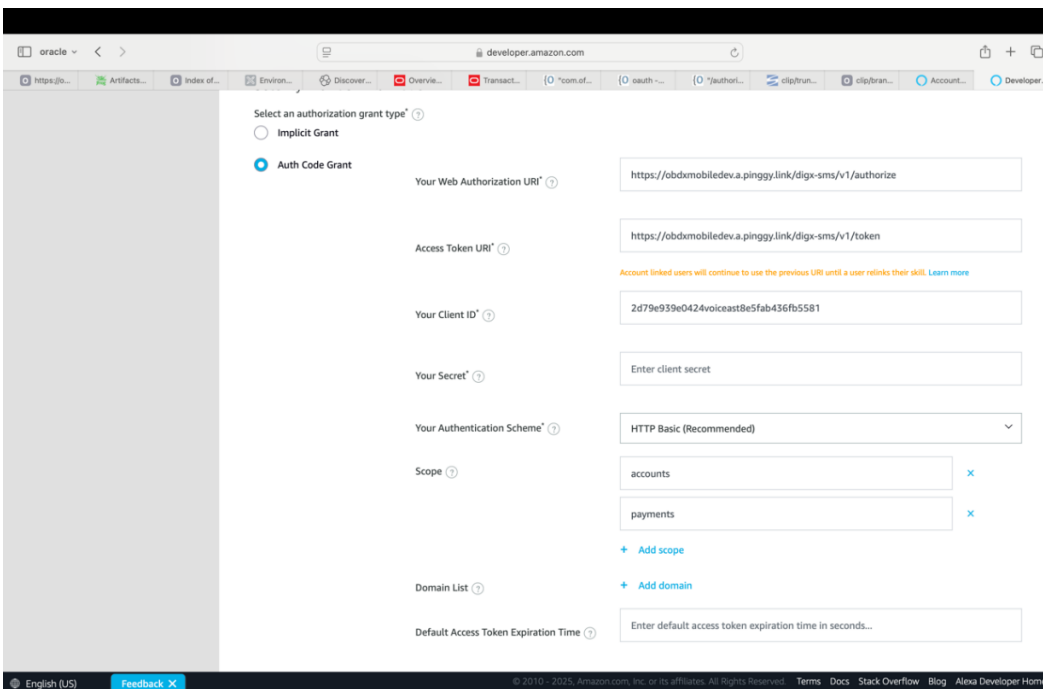
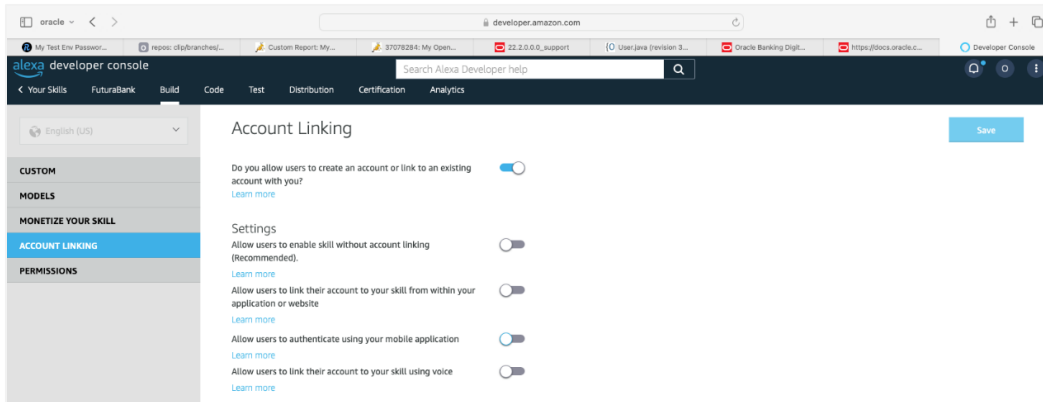


8. Click Build Model in Build tab.
9. Update endpoint and add CA.
10. Note the skill ID.



## Setup Account Linking for Alexa

1. Go to Account Linking Tab in Alexa console and enable “allow users to create an account or link to an existing account”.
2. In Security Provider Information select “Auth Code Grant” and set your web authorization Uri `https://<your server url>/digx-sms/v1/authorize` and access token URI: `https://<your server url>/digx-sms/v1/token`
3. Set scope to “accounts”.



4. Set the client Id for client from client Maintenance page for “OBDXVoiceAstDomain” identity domain client.
5. Click **Edit** and set the client secret for the above client which can be generated from the Client Maintenance Page.

Client Maintenance

Client Definition

Client Id: 2d79e939e0424voiceast8e5fab436fb5581

Client Name: OBDXVoiceAstClient

Client Description: OBDXVoiceAstClient

Identity Domain: OBDXVoiceAstDomain

Client Type: Confidential Client

Grant Type: AUTHORIZATION\_CODE, PASSWORD, REFRESH\_TOKEN

Routing Method: Normal OAuth

Redirect URL

Redirect URL: https://pitangulamazon.com/api/skill/link/M2VXLT89KWICR, https://alexa.amazon.co.jp/api/skill/link/M2VXLT89KWICR, https://alexa.amazon.com/api/skill/link/M2VXLT89KWICR

Scope

Resource Server: OBDXVoiceAstServer

6. Go to Resource Server Maintenance page and select Identity domain to OBDXVoiceAstDomain.
7. Edit the Scope Name of SCO2 to “accounts” and Scope Description to “accounts”.

Resource Server Maintenance

Resource Server Definition

Resource Server Name: OBDXVoiceAstServer

Resource Server Description: OBDXVoiceAstServer

Identity Domain: OBDXVoiceAstDomain

Scopes

Scope Name: accounts, Scope Description: accounts

Scope Name: SCO5, Scope Description: Payments

Add Scope

Save, Cancel, Back

8. Go to Touch Point Maintenance Page and search for “Voice Assistant Banking” touch point name.
9. Add “accounts” scope to the list of scopes and add upload your bank logo.
10. Enable Touch point status.



**Touch Point Maintenance**

Touch Point Id: APVOICEAST

Touch Point Name: Voice Assistant Banking

Touch Point Type: ☐ Internal ☒ External

Client Id: 2d79e939e0424voiceast8e5fab436fb5581

Scope:
   
Account Balance Inquiry X
   
Account Details Inquiry X
   
Domestic Transfers X
   
International Transfers X
   
Internal Transfers X
   
accounts X

Upload Logo: ☐ pngtree-rainbow-curves-abstract-colorful-background-image\_2164067.jpg

Touch Point Status: ☐

Headless Mode: ☐

Two Factor Authentication: ☐

Self On Board Touch Points: ☐

Skip First Time Login Flow: ☐

Consent Required: ☐

[Save](#) [Cancel](#) [Back](#)

11. Go to role transaction Mapping page and create an external user and add accounts scope.

**Role Transaction Mapping**

Application Role Creation 2 Map Transaction

Application Role Name: accounts

Description: accounts

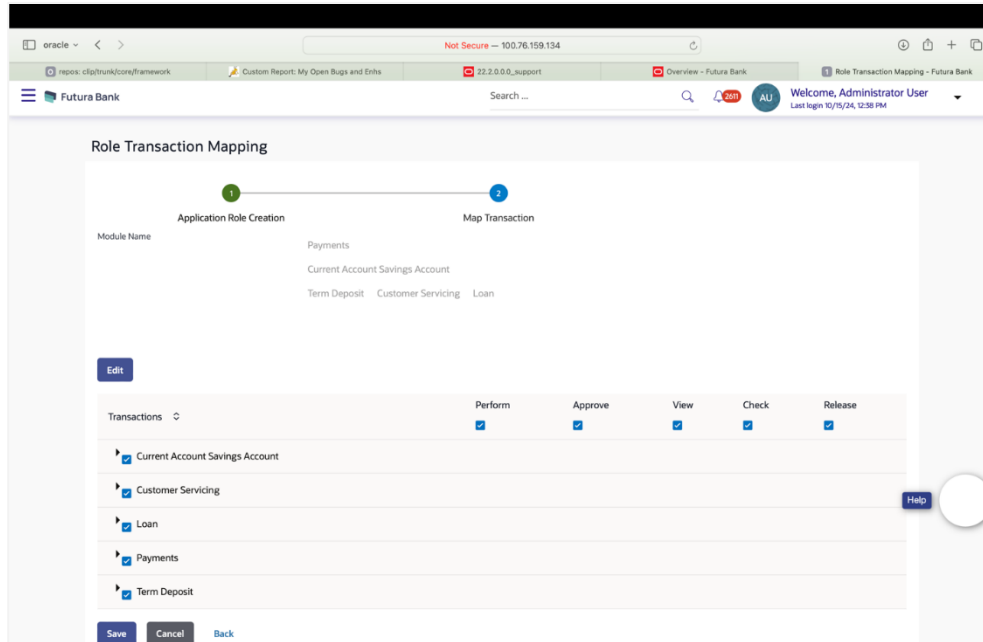
User Type: Retail User

Touch Point Type: ☐ Internal ☒ External

Scope Name: accounts

[Map Transaction](#) [Cancel](#) [Back](#)

12. Add module “current account savings account”, “term deposit”, “customer servicing” and “loan”, “payments”.



13. Set the 'CODE\_CHALLENGE\_FLAG' , 'OPAQUE\_ACCESS\_TOKEN\_FLAG' flags to N in auth\_config table.

```
update auth_config set prop_value='N' where prop_id='CODE_CHALLENGE_FLAG';
```

```
update auth_config set prop_value='N' where prop_id='OPAQUE_ACCESS_TOKEN_FLAG';
```

14. Update token\_expiry in digx\_au table for OBDXVoiceAstDomain domain\_id

```
update DIGX_AU_tokendetail set TOKEN_EXPIRY=600 where DOMAIN_ID='c733dee8-6e4f-4400-8170-cf7c6ef98165'.
```

15. Restart the server.

## Create a Webhook channel in ODA

1. In the Bot Builder, create a webhook channel for your bot. In the Create Channel dialog, enter the outgoing Webhook URL as **https://bots-samples-nodejs:8889/ext/apps/alexa-singleBot/singleBotWebhook/messages**. This URL is where your bot will send its responses back to the Alexa Futura Bot skill.

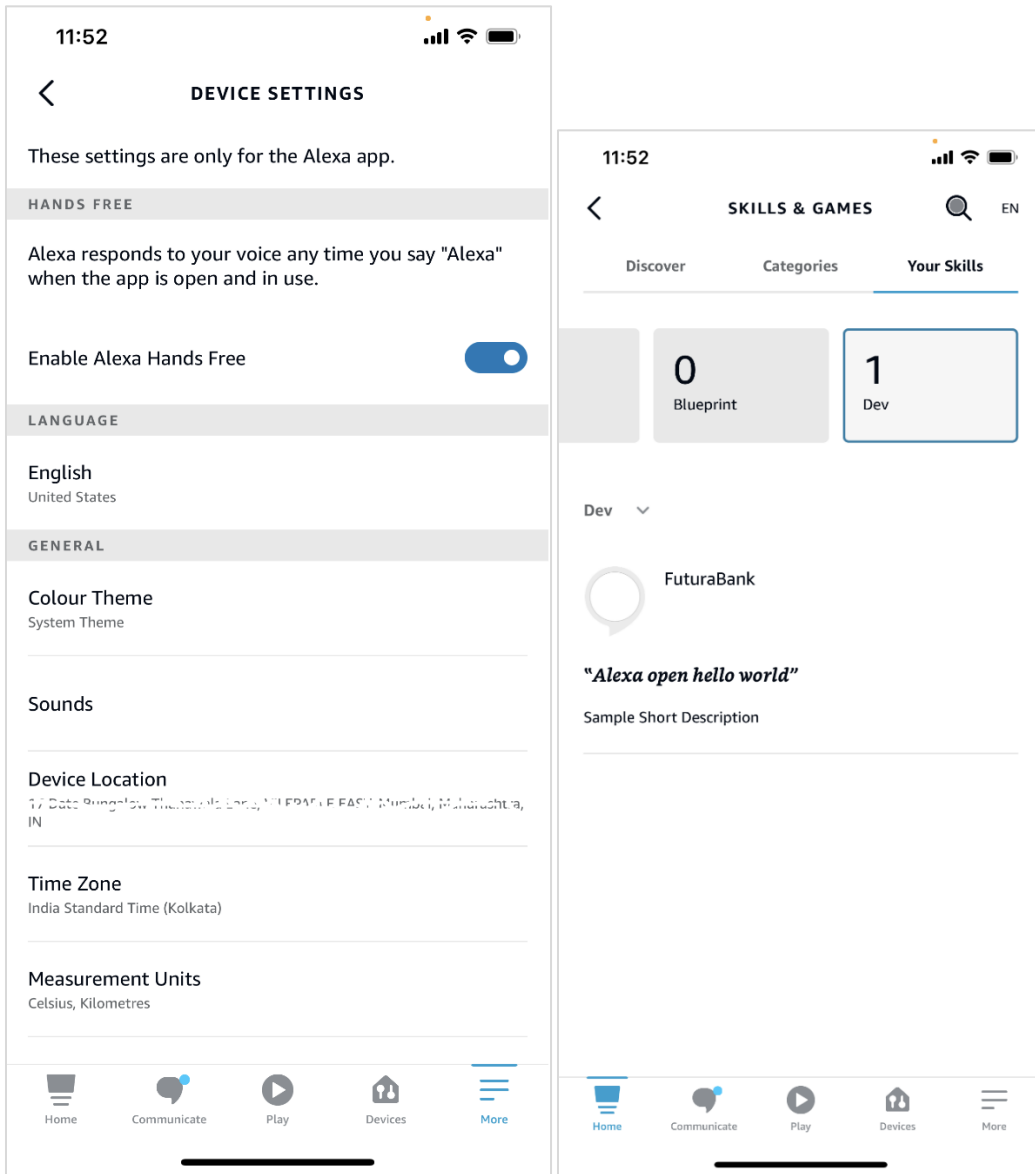
The screenshot shows the 'Channels' configuration page in the ODA interface. On the left, there's a sidebar with 'Channels (1)' and a list containing 'OBDXAlexa'. The main area shows the configuration for 'OBDXAlexa'. It includes a 'Channel Enabled' toggle, a 'Route To' dropdown set to 'OBDX201', and a 'Name' field with 'OBDXAlexa'. The 'Description' field is empty. The 'Channel Type' is set to 'Webhook'. The 'Payload Version' is set to 'Conversation Model'. The 'Outgoing Webhook URL' is set to 'https://vhl00ebe.in.oracle.com:13000/singleBotWebhook/messages'. The 'Secret Key' is 'PcmklEjmeffNZJ8oocMY1QykhOo9BLvt'. A 'Webhook URL' field at the bottom shows the full URL: 'https://ids-oda-d501204fbc4d496dac7b74d15a590dce-t0.data.digitalassistant.oci.oc-test.com/connectors/vZ/listeners/webhook/channels/65223783-8657-4287-906e-3e34010b005e'.

- a. Update amzn\_appld, channelSecretKey & channelUrl in service.js
- b. Run nodeJS script
- c. npm install
- d. node index.js
- e. This must be publicly hosted in https URL.
- f. Enter this URL in below screen

The screenshot shows the 'Service Endpoint Type' configuration page in the Amazon Alexa Developer Console. The 'HTTPS' option is selected. The 'Default Region' is set to 'us-east-1'. The 'Enter URL' field contains 'https://vhl00ebe.in.oracle.com:13000/singleBotWebhook/messages'. The 'Select SSL certificate type' dropdown is set to 'My development endpoint is a sub-domain of a domain that has a wildcard certificate from a certificate authority'. The 'North America', 'Europe and India', and 'Far East' regions are also listed with their respective 'Enter URL' and 'Select SSL certificate type' fields.

2. In the Alexa Mobile App
  - a. Ensure that the skill is visible

b. Change the language to English US



3. Link your amazon account for your skill through alexa app.

## 5.6 WhatsApp Channel

1. Create a channel of type Webhook.
2. Map the Skill to this channel & enable the same.

The screenshot shows the 'Channels' configuration page in the ODA console. On the left, a sidebar lists 'Channels (1)' with a search bar containing 'whatsapp' and a list item for 'OBDX\_Whatsapp'. The main area displays the configuration for 'OBDX\_Whatsapp', which is 'Channel Enabled'. Fields include 'Route To' (OBDXTrunkRUNT), 'Name' (OBDX\_Whatsapp), 'Description' (Optional short description for this channel), 'Channel Type' (Webhook), 'Payload Version' (Conversation Model), 'Outgoing Webhook URL' (https://oda-adui.onrender.com/bot/message), 'Secret Key' (ao3HasDqREZUVVJSTWwWNYB7wFdzou), and 'Webhook URL' (https://oda-ae2cca311e58405abe8c40f4131b850f-da2.data.digitalassistant.oc1.oraclecloud.com/connectors/v2/listeners/webhook/channels/6395807d-e7a6-42a3-9688-3792ad555a7). A 'Session Expiration (minutes)' field is set to 1,000. A 'Reset Sessions' button is in the top right.

3. Obtain the NodeJS script from installer from this path –  
"OBDX\_Installer/installables/OBDX/BASE/<OBDX patchset version>/chatbot/config/oda-whatsapp/index.js"  
This script resides between WhatsApp Server & ODA.
4. Complete the configurations as given in Sec 3.8 WhatsApp configurations of Alert Configuration Guide.
5. Note the Webhook URL & secret from above screen & update in NodeJS script.
6. Update the long term WhatsApp token in MYTOKEN variable in the script.
7. Run the shared NodeJS script on any public server.
8. Update the URL in below WhatsApp Console. The verify token is the one used in "mytoken" variable in script. Update the outgoing webhook URL in ODA channel.

The screenshot shows the 'WhatsApp' configuration page in the 'Meta for Developers' console. The page is titled 'Quickstart > Configuration'. It includes sections for 'Webhook' (with a 'Callback URL' and 'Verify token'), 'Webhook fields' (for messages), 'Permanent token' (with a link to learn how to create one), 'Phone numbers' (with a link to manage phone numbers), and 'Test account' (with a link to delete your business).

---

## 6. Configuration in OBDX

1. This configuration is applicable for WEB/ANDROID/IOS channel.
2. Login to OBDX website with admin user.
3. Go to Touch Point Maintenance and search for APCHATBOT.
4. Enable APCHATBOT touchpoint for chatbot to work with OBDX.
5. For 2fa, enable 2fa Authentication. Enabling this will enable 2fa authentication for transactions withing chatbot as per authentication type configured for the transaction.
6. Login 2fa option available are PIN/OTP/Soft token.
  - a. For PIN, user can login to OBDX website -> Profile Settings ->Password and security ->Set SMS and chatbot PIN
  - b. Update the type of authentication required in ODA in fw\_config\_all\_b

select \* from digx\_fw\_config\_all\_b where prop\_id like '%MobileAuthenticator%'

values are LOCAL- OTP

R\_SOFT\_TOKEN- HOTP based 2fa

T\_SOFT\_TOKEN, PIN – TOPT base 2fa

PIN

## 7. Standalone WhatsApp/Facebook messenger integration with OBDX

Bank can also setup WhatsApp without Oracle Digital Assistant in OBDX. In this release, basic intents are supported

- Help
- Balance
- Recent Transactions
- Upcoming Payments

All the above transactions will have authentication and authorization as in OBDX for a OBDX user.

Login will be performed based on OTP.

Below is the configuration setup:

1. The relevant antifactory for server node application will be available from OBDX installer from this path:

“OBDX\_Installer/installables/OBDX/BASE/<OBDX patchset version>/chatbot/config/digital-assistant/”

2. Navigate to Facebook Developers Console → WhatsApp Configuration. Add the Webhook Callback URL and set up the Verification Token to enable webhook integration

Quickstart > Configuration

**Webhook**  
To get alerted when you receive a message or when a message's status has changed, you need to set up a Webhooks endpoint for your app. [Learn how to configure Webhooks.](#)

Callback URL

Verify token

☐ Attach a client certificate to Webhook requests. [Learn more.](#)

[Remove subscription](#) [Verify and save](#)

Field	Version	Test	Subscribe
account_alerts	v22.0	Test	<input type="radio"/> Unsubscribed
account_review_update	v22.0	Test	<input type="radio"/> Unsubscribed
account_update	v22.0	Test	<input type="radio"/> Unsubscribed
business_capability_update	v22.0	Test	<input type="radio"/> Unsubscribed
business_status_update	v22.0	Test	<input type="radio"/> Unsubscribed
campaign_status_update	v22.0	Test	<input type="radio"/> Unsubscribed

3. For setting up node server,
  - a. Install node using homebrew
  - b. Then open the path of the folder extracted from the installer package on terminal. This is be the path where package.json is located.
  - a. Run “npm install”

4. In environment variables add the following variables:
  - a. **PHONE\_NUMBER\_ID**: can be fetched from Facebook developer's console under: Quickstart > API Setup
  - b. **VERIFY\_TOKEN**: should be same as one set in Facebook developers console/WhatsApp configuration
  - c. **PAGE\_ID**: is your Facebook page ID:
    - Go to your Facebook Page.
    - Click on "**About**" (left-hand menu).
    - Scroll down to "**Page ID**", and you will see a numeric ID.
  - d. **CHANNEL**: can be set as **WhatsApp** or **Facebook** respectively to use any one of the channels (WhatsApp or Facebook). Only one at a time is supported.
  - e. **BASE\_URL**: should be a https URL of the environment you want to point to.
  - f. **DEFAULT\_HOME\_ENTITY**: to be set based on requirement in OBDX
  - g. **MYTOKEN**: This token is required to send messages via the WhatsApp API.:
    - Go to [Meta for Developers](#).
    - Navigate to "**My Apps**" and select your app.
    - Under "**WhatsApp**", go to "**API Settings**".
    - Find the "**Temporary Access Token**" or generate a permanent token via Business Manager.
    - Use this token as **MYTOKEN** in your `.env` file.
  - h. **PAGE\_ACCESS\_TOKEN**: This token is needed to send messages via the Facebook Messenger API:
    - Go to [Meta for Developers](#).
    - Select your app and navigate to **Messenger > Settings**.
    - Under **Access Tokens**, choose your Facebook Page and generate a token.
    - Save this token as **PAGE\_ACCESS\_TOKEN** in your `.env` file.
5. After configuring the environment variables, deploy the code and start the application by running 'node index.js' to start the node server.
6. Based on the channel that is set you can now test the chatbot on WhatsApp/Facebook.

[/digital-assistant/src/config/messages.json](#) contains all text messages which can be customized.

---

Note: There is language translation service in between the intent and WhatsApp server.

---

[/digital-assistant/src/services/IntentService.js](#) has mapped intent. These intents can be added in the array.