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/

Table of Contents

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

boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
Italic	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:

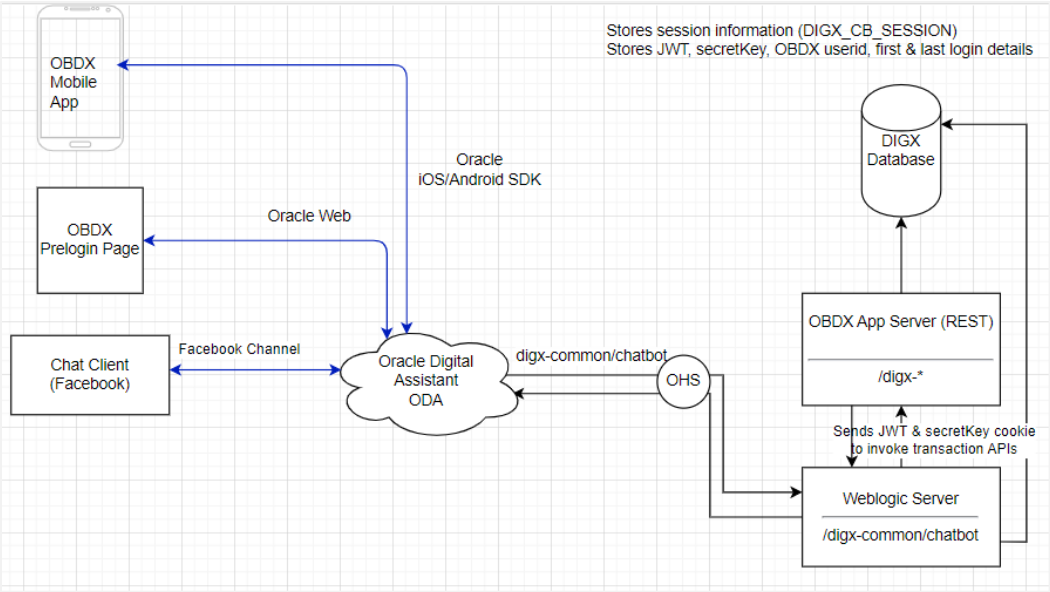
Abbreviation	Description
OBDX	Oracle Banking Digital Experience

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.

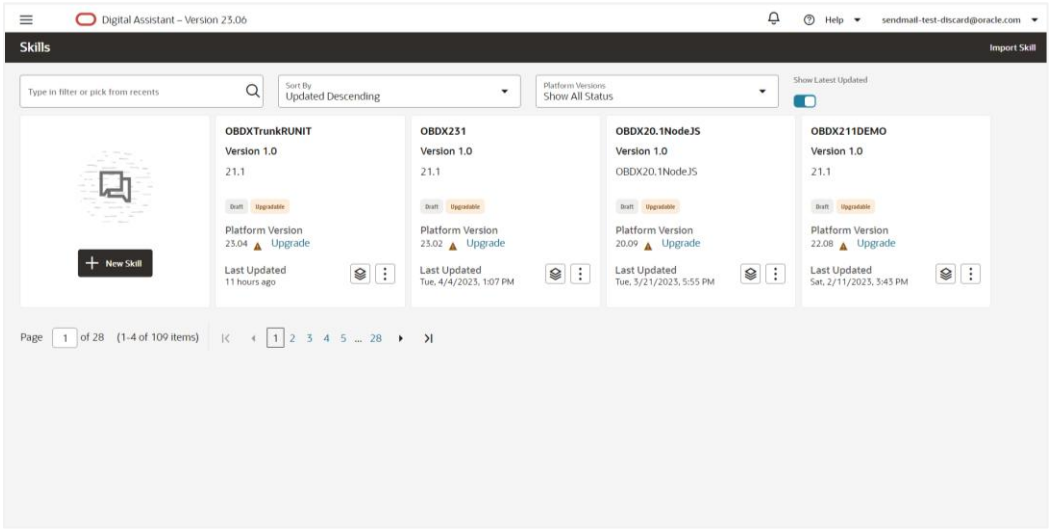
3. Topology


Stores session information (DIGX_CB_SESSION)
Stores JWT, secretKey, OBDX userid, first & last login details
OBDX Mobile App
Oracle iOS/Android SDK
DIGX Database
OBDX Prelogin Page
Oracle Web
OBDX App Server (REST)
/digx-*
Chat Client (Facebook)
Facebook Channel
Oracle Digital Assistant ODA
digx-common/chatbot
OHS
Sends JWT & secretKey cookie to invoke transaction APIs
Weblogic Server
/digx-common/chatbot

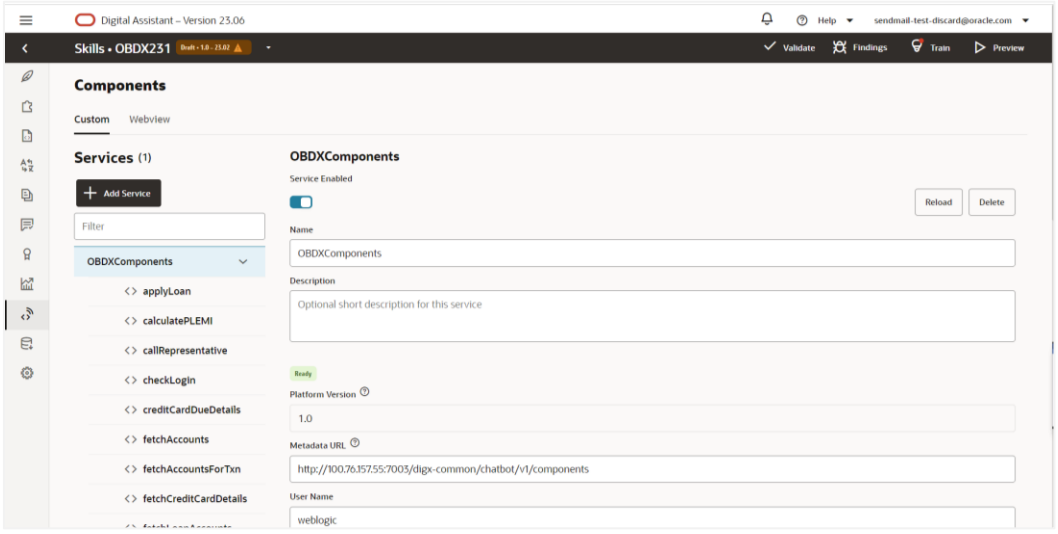
4. Common Configurations

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.

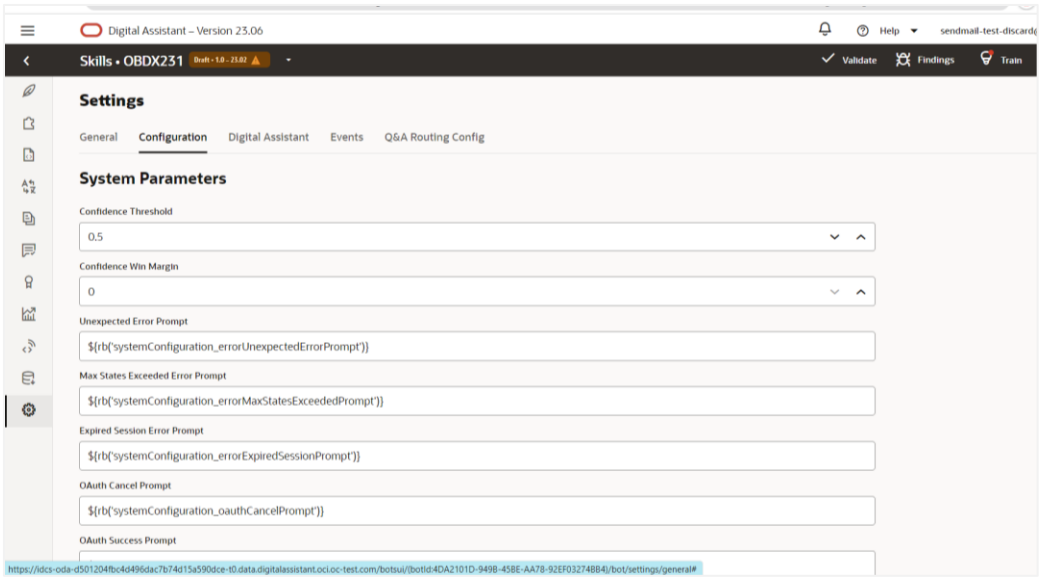


2. Click on the OBDX Bot and click on the components to add the custom components.

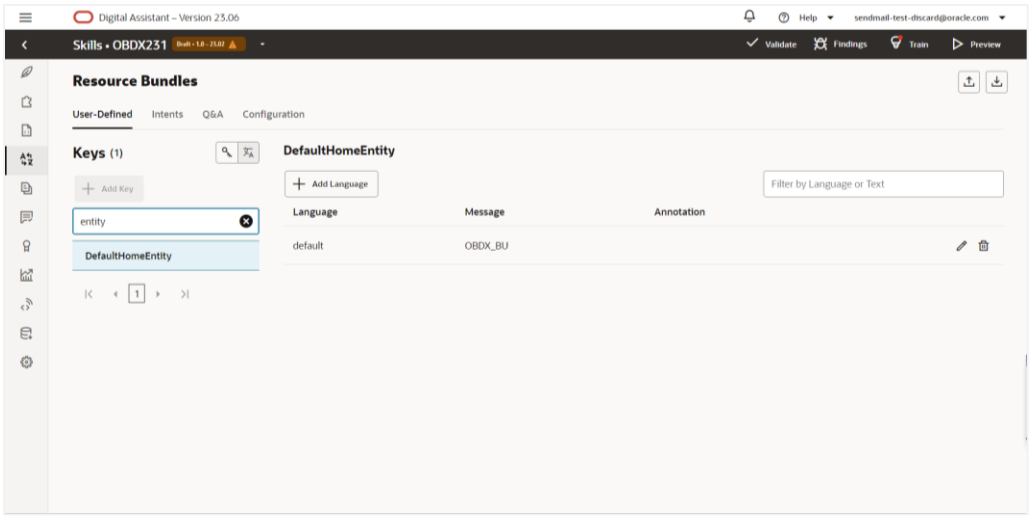


3. Put the OBDX URL here. The OBDX setup and the ODA setup must be accessible over Internet.
4. Add username/password (in HTTP Basic authorization) of any user with Administrators role which can be used to login in OBDX WebLogic server.

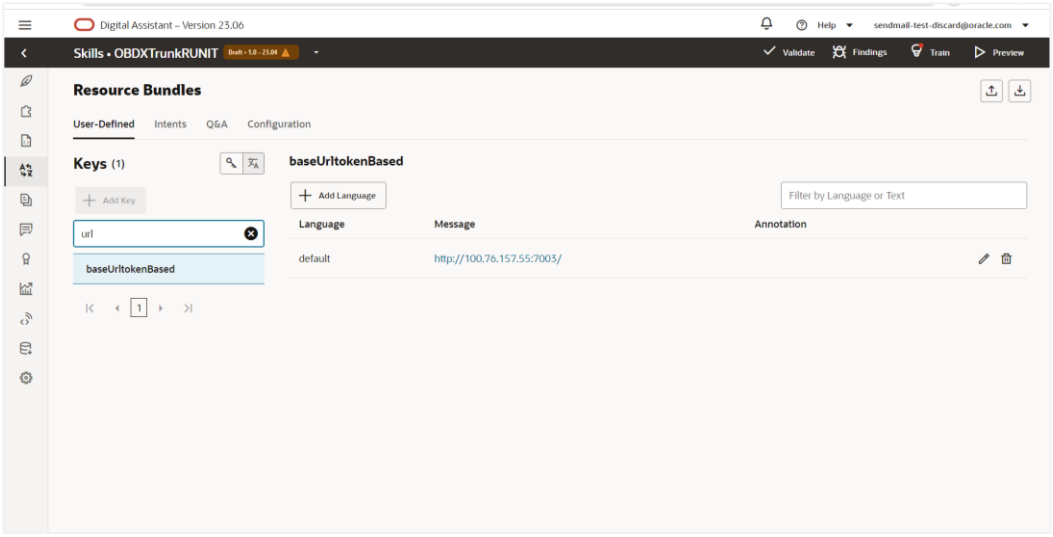
5. To configure intent threshold for the skill, go to settings tab in a bot and click configurations to configure threshold frequency (default 0.5) as shown below:



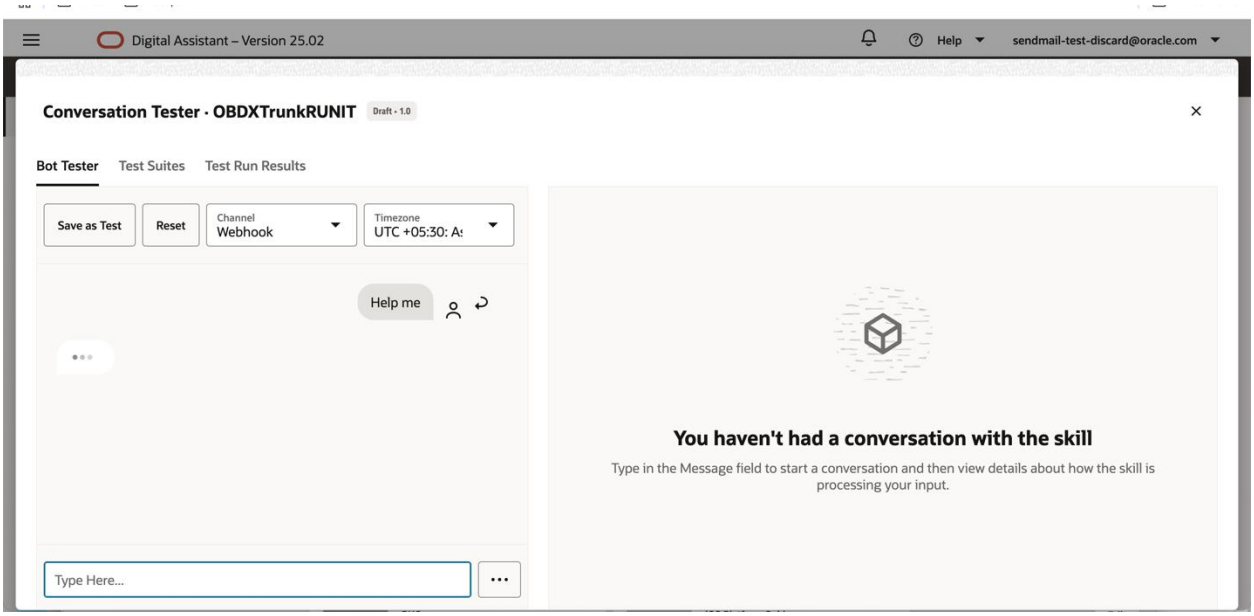
6. Configure required entity in a resource bundles (one created/configured in OBDX). At a time chatbot works with single entity. Go to Resource Bundles as shown below-



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.

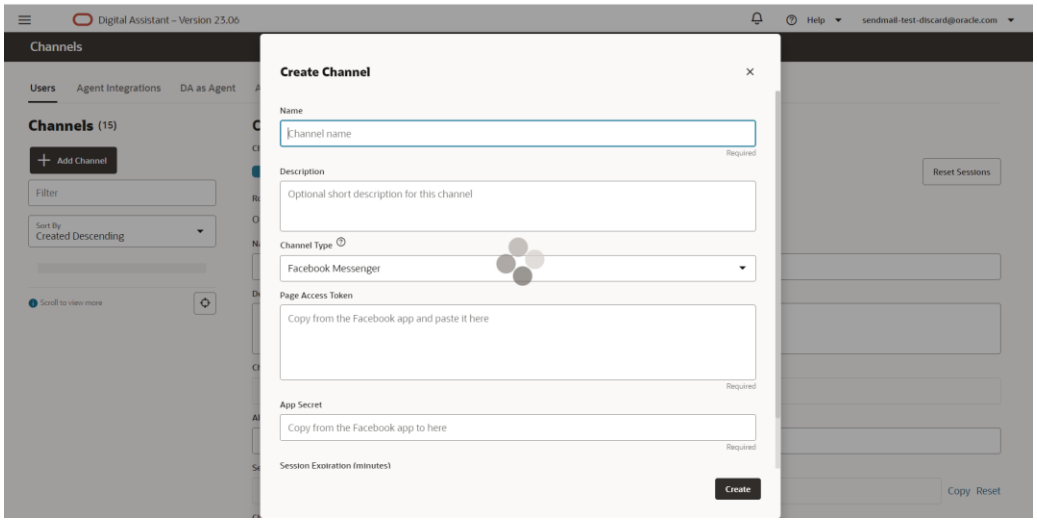
Chatbot widget appears on pre-login page in OBDX UI where the login is with mobile number.(OTP or PIN or Soft Token)

For post login, the browser session is automatically used

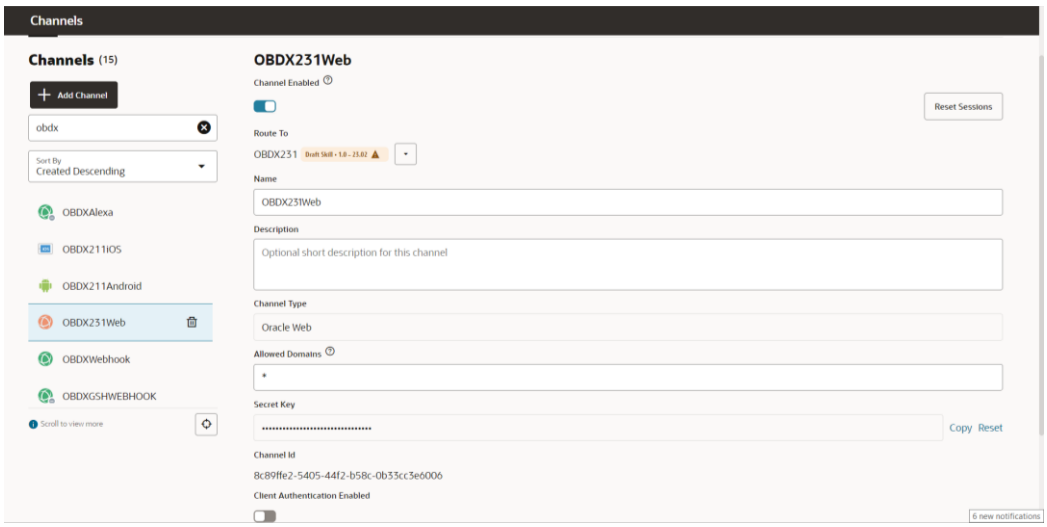
5.1 Web Channel

To configure this channel:

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



2. Create Oracle Web as the channel type.



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

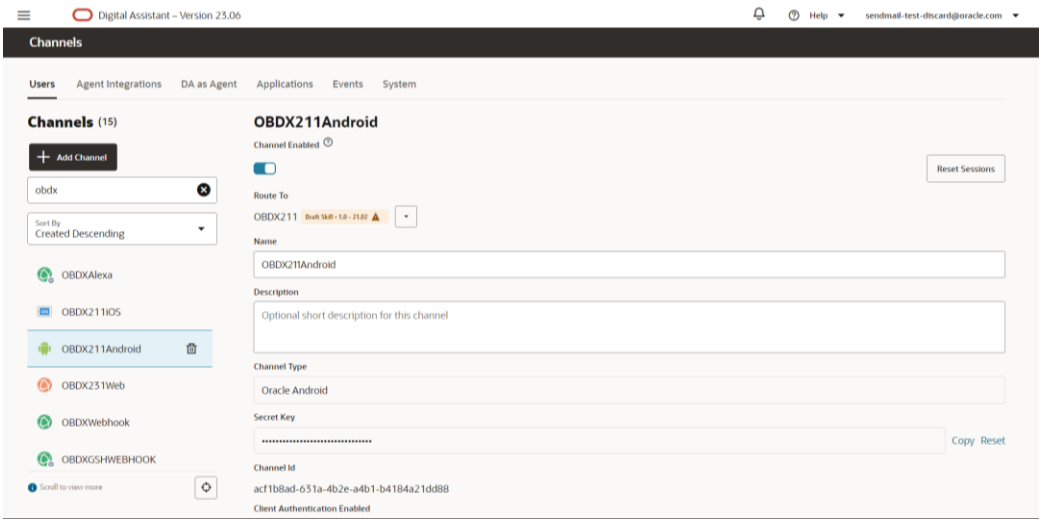
To remove this chatbot widget

- 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.

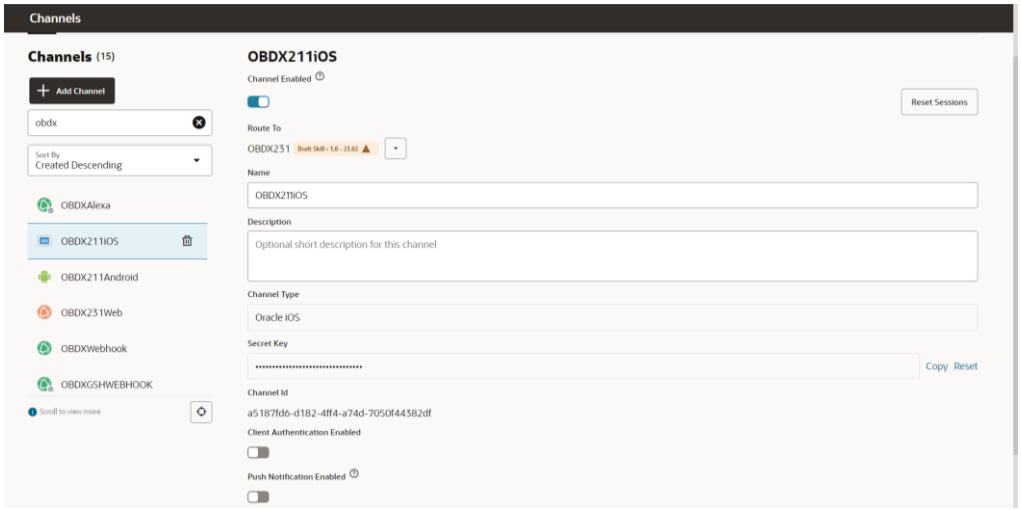


3. Route the channel to your skill or digital assistant.
4. Switch On Channel Enabled switch.
5. Note the channel Id. This is required in application configuration. Refer application configuration guide for more details.

5.3 iOS Channel

To configure this channel:

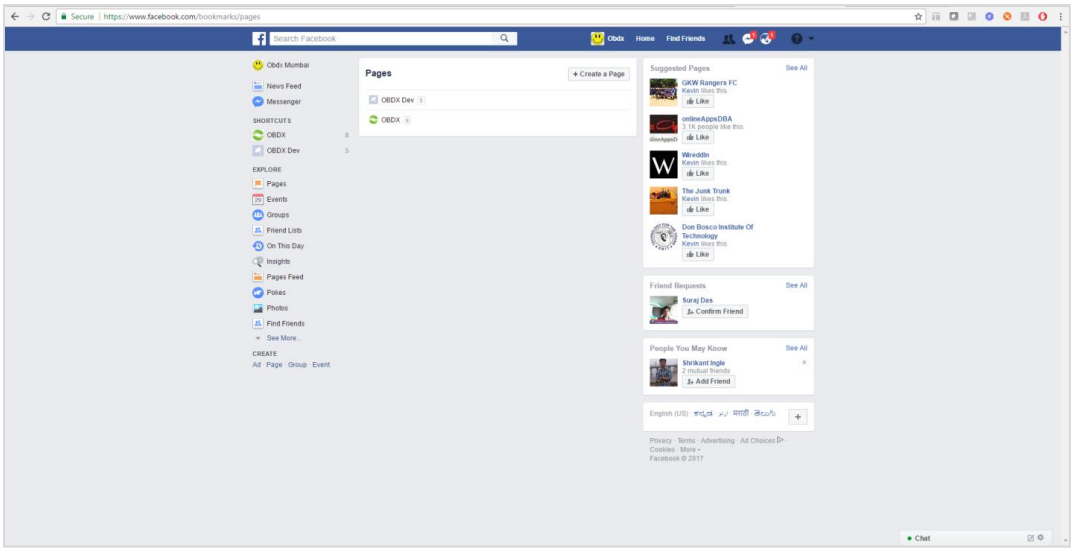
1. Choose Development->Channels->Users from the menu. Click Add Channel.
2. Add Oracle iOS as the channel type.



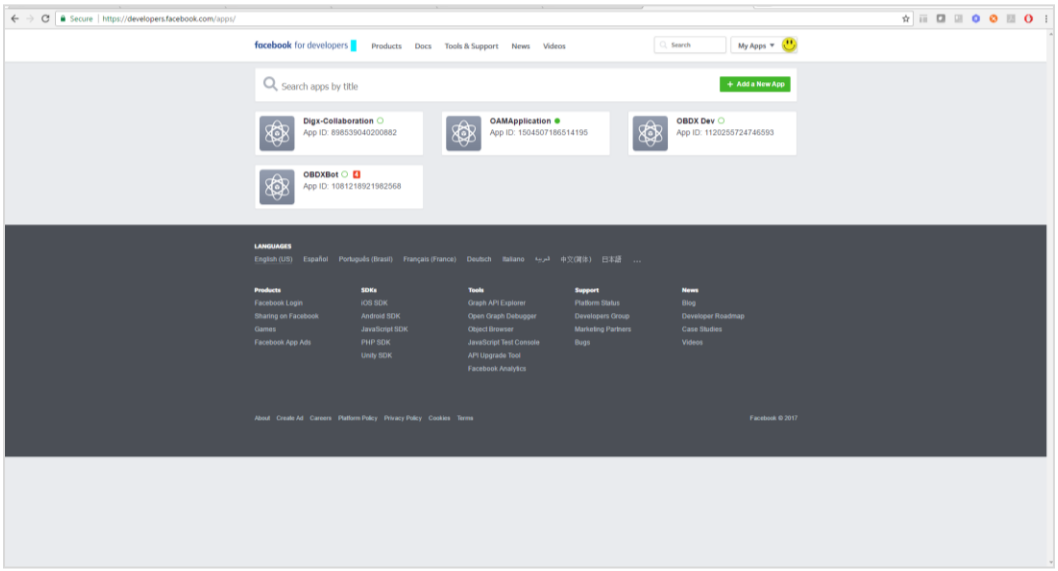
3. Route the channel to your skill or digital assistant.
4. Switch On Channel Enabled switch.
5. Note the channel Id. This is required in application configuration. Refer application configuration guide for more details.

5.4 Facebook Channel

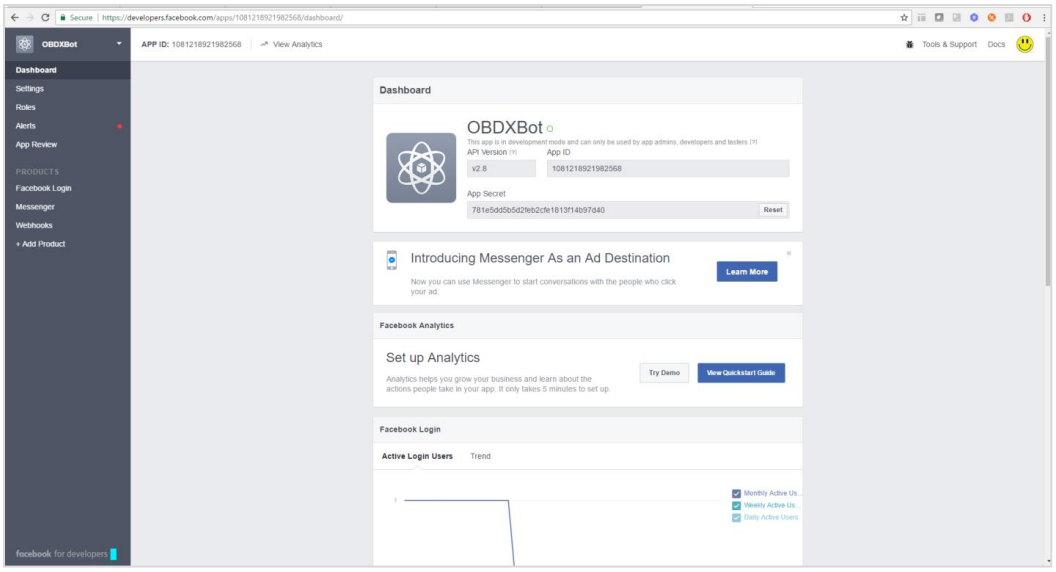
1. Create a Facebook account for the Bank. Login to Facebook with credentials.
2. Create a new page.



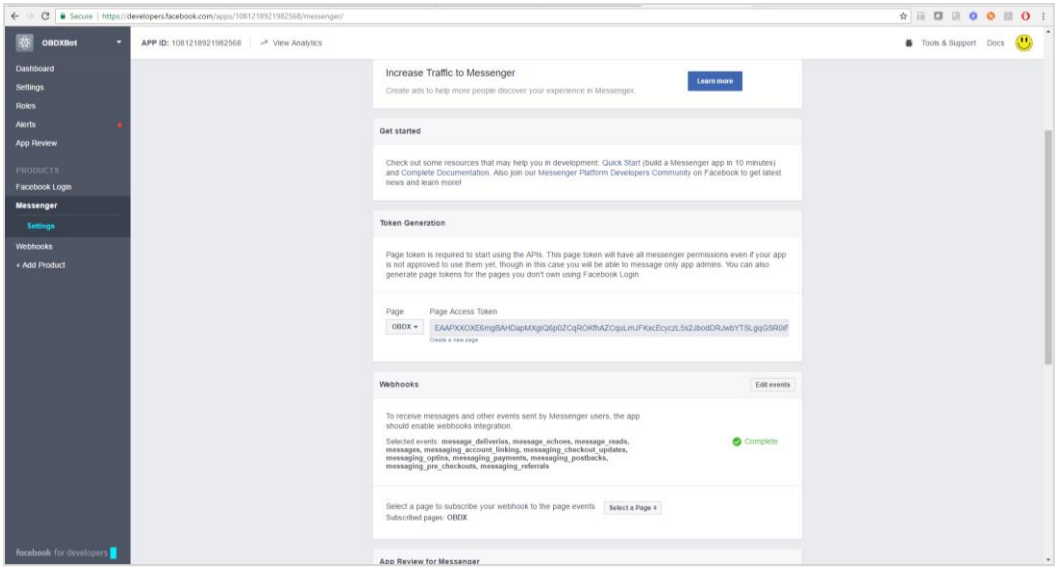
3. Navigate to developer link and create an application as shown below



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

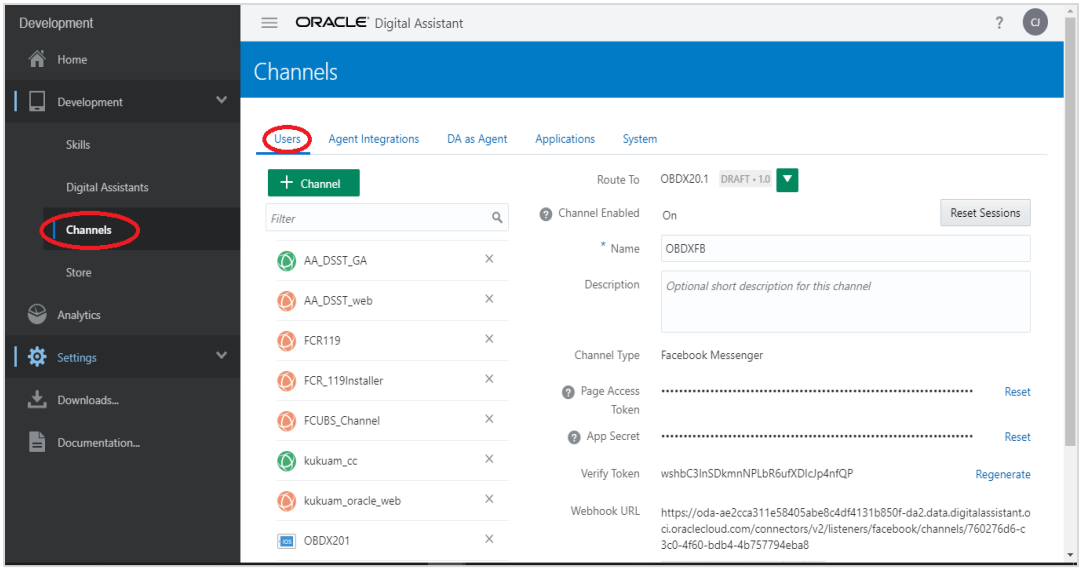


5. Navigate to Messenger > Settings page from left panel and in token generation section select the page created previously. Note the page access token.

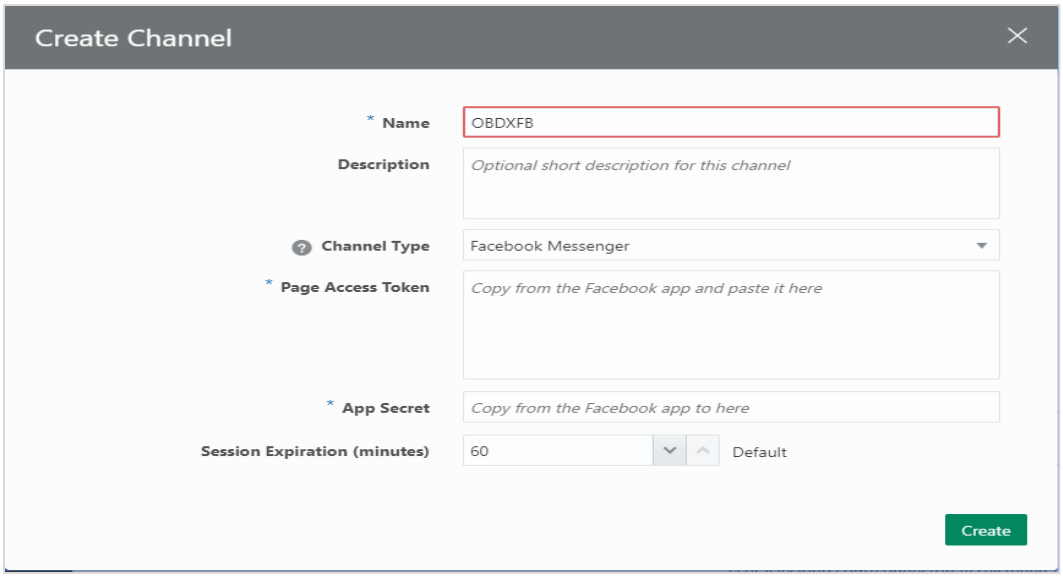


Create Facebook Channel in ODA

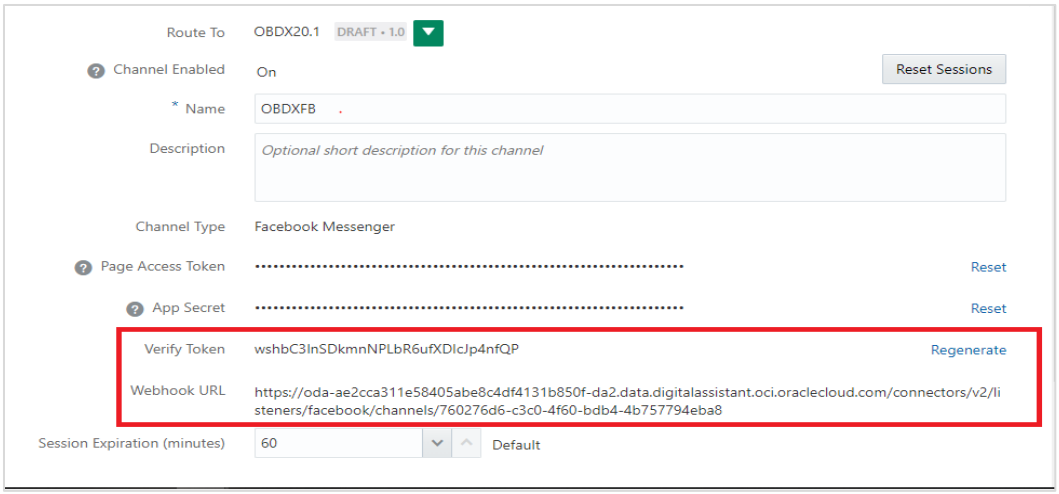
1. In ODA ,click channels in the left panel/menu and then click on users.



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



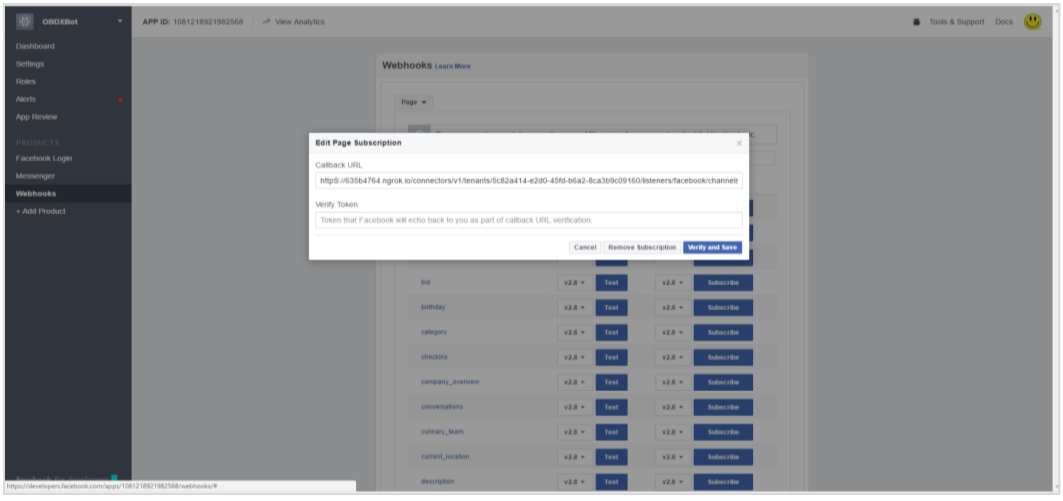
4. In the Page Access Token field, paste the page access token that you generated previously in the Set Up Facebook Messenger procedure.
5. In the App Secret field, paste the app secret that you copied previously in the Set Up Facebook Messenger procedure and click Create.
6. In the Channels page, copy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



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

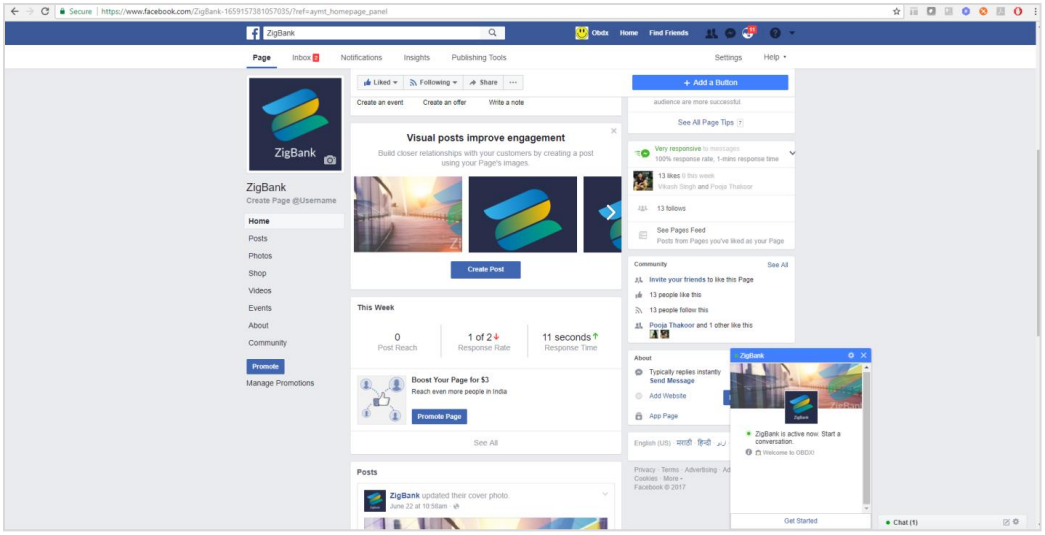
1. In Facebook Messenger, be sure that you've selected the project that you initially created for the webhook.
2. Click **Messenger** and then choose Settings.
3. Click **Subscribe to Events** to open the New Page Subscription dialog.
4. Copy the Webhook URL that you got from the Digital Assistant Channels 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.



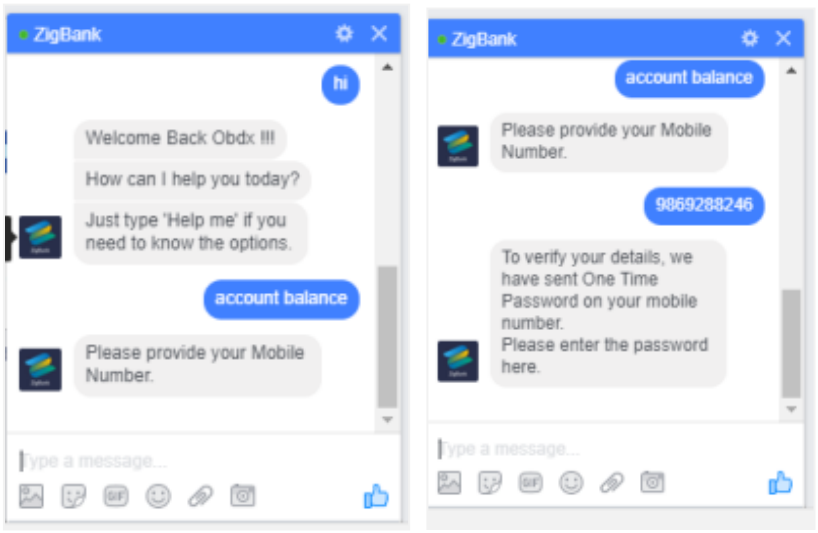
6. Under Subscription Fields, select the messages and messaging_postbacks callback events. The messages event is triggered whenever someone sends a message to your Facebook page.
7. Click Verify and Save.
8. In the Webhooks section of the Messenger settings, select the Facebook page for your digital 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.



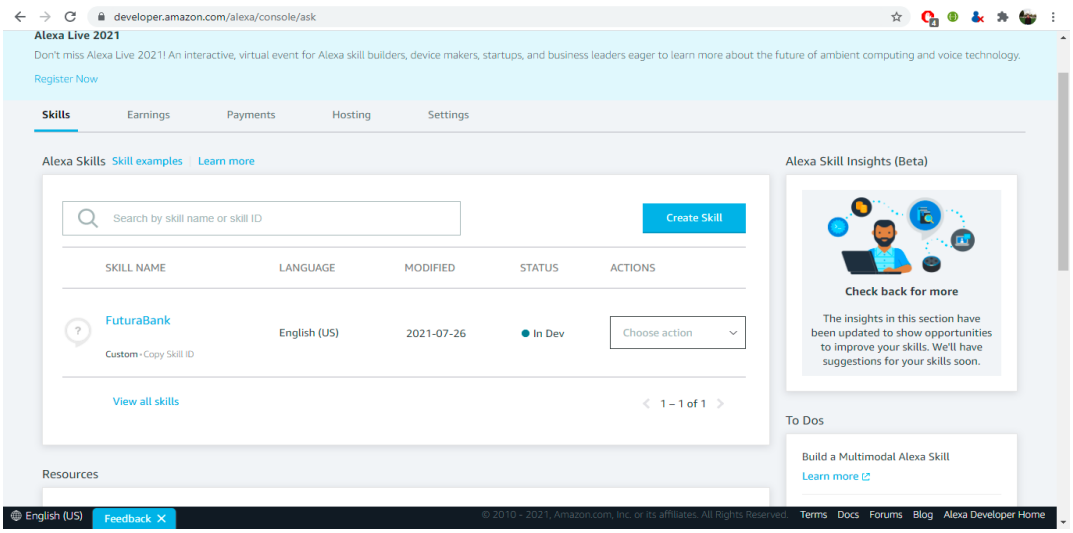
5.5 Alexa Channel

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

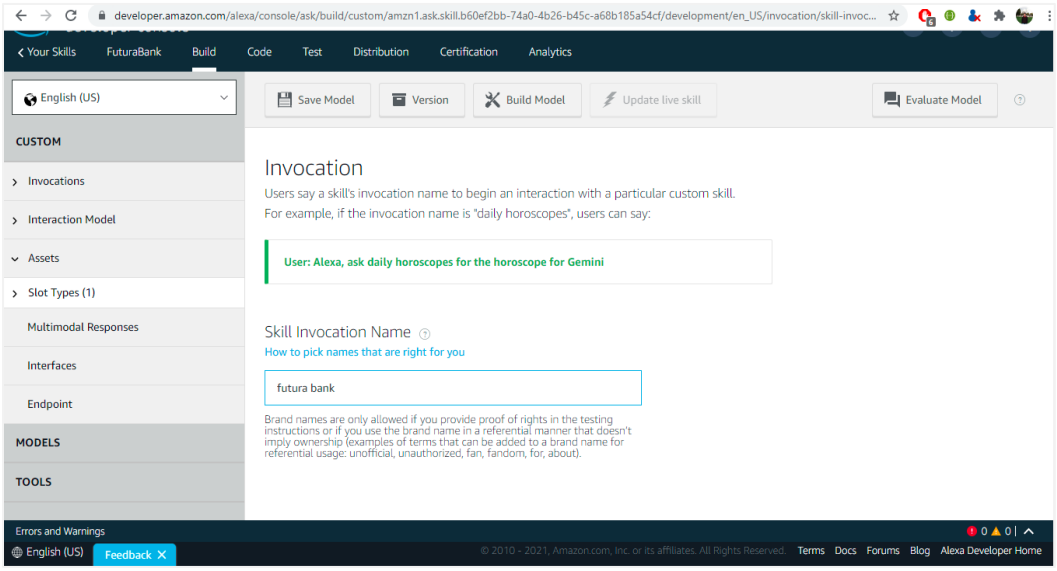
Add the skill information

Set up a developer account in the Amazon Developer Portal.

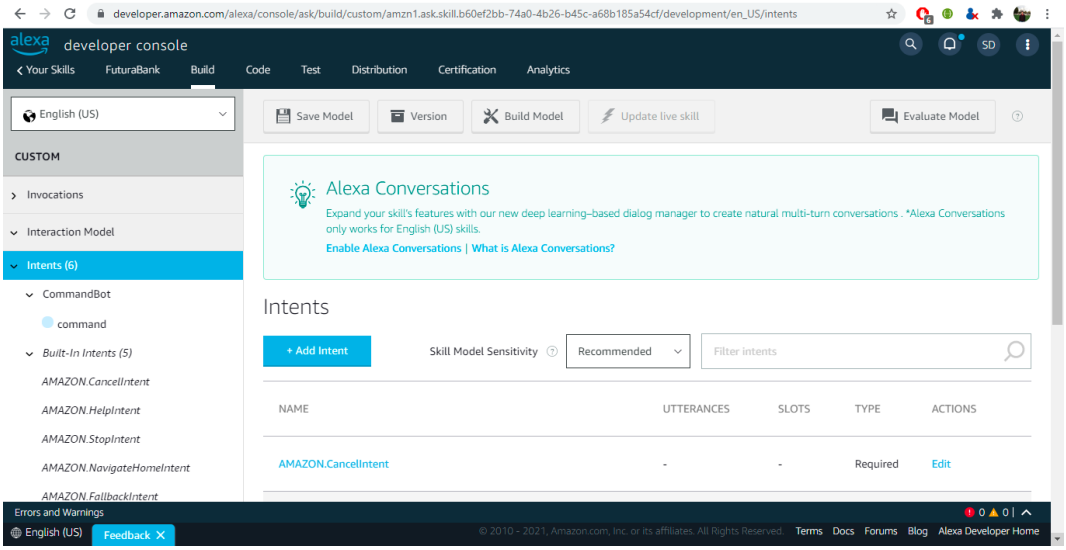
1. Open the **Amazon Developer Console**.
2. Click on **'Create Skill'**



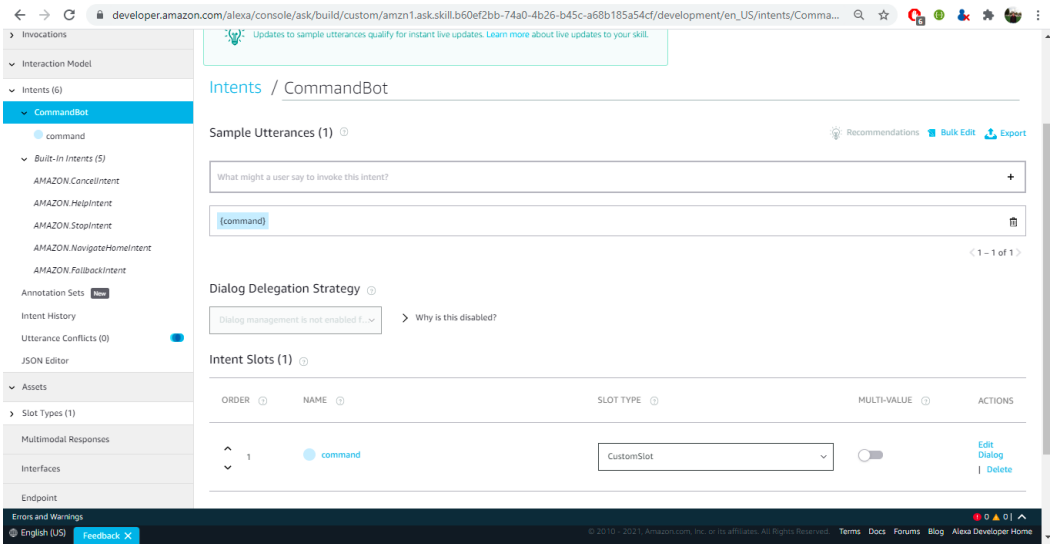
3. Enter Futura Bank (or any name that you want to use to invoke this skill) as the Invocation Name.



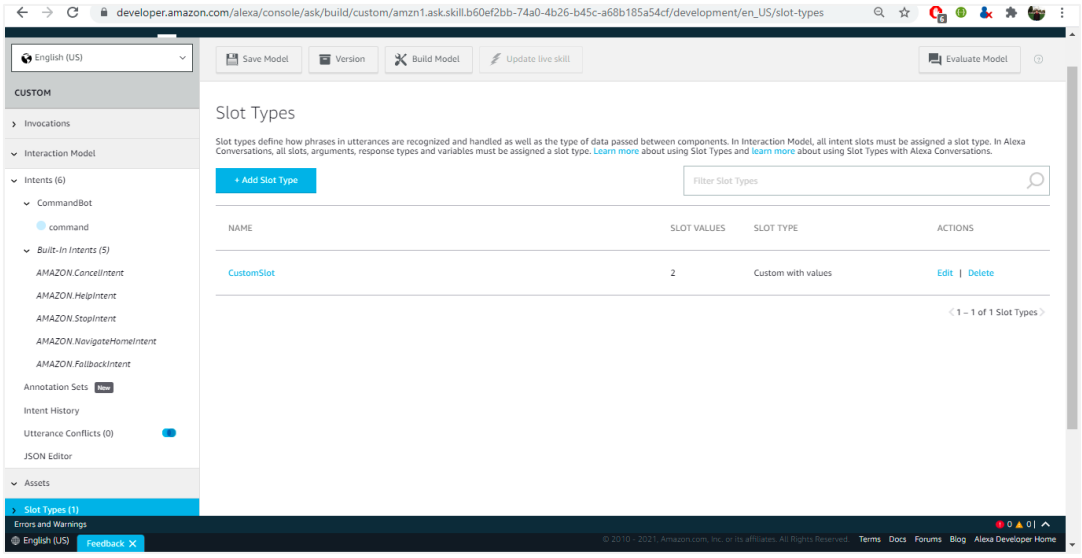
4. Add new intent as CommandBot.

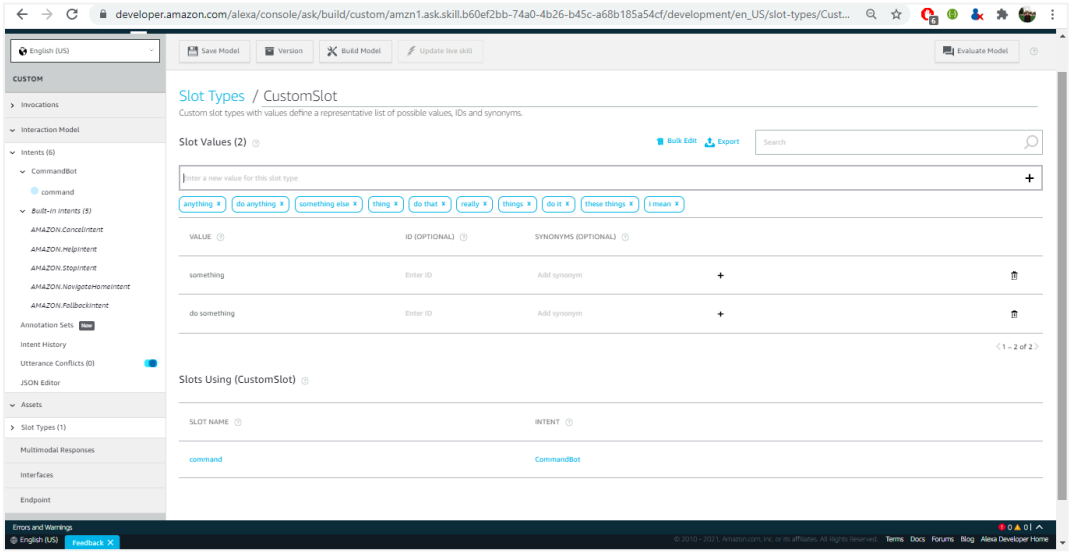


5. Add sample utterance to it.

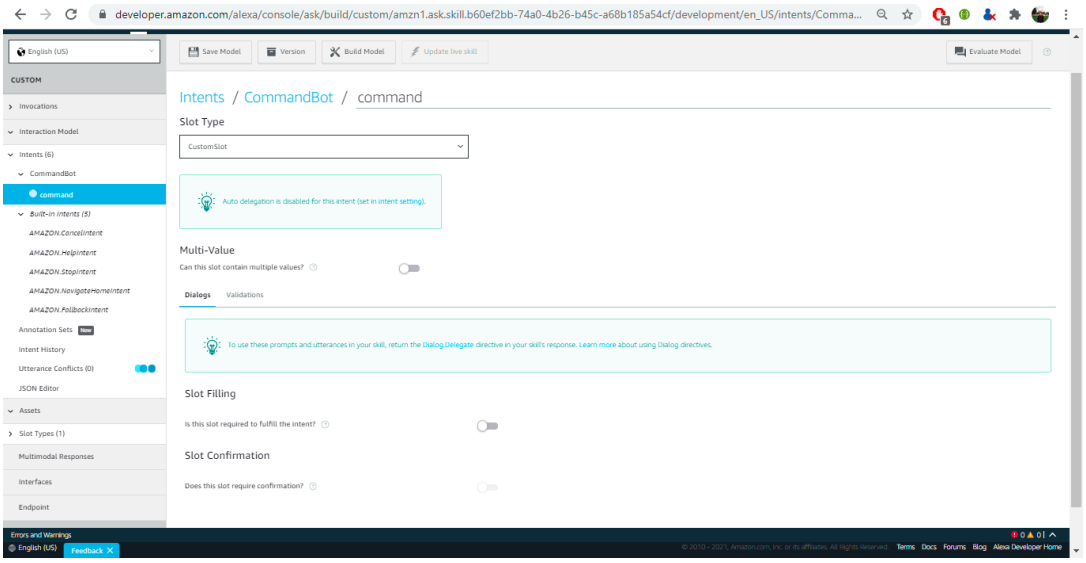


6. Create custom slot.

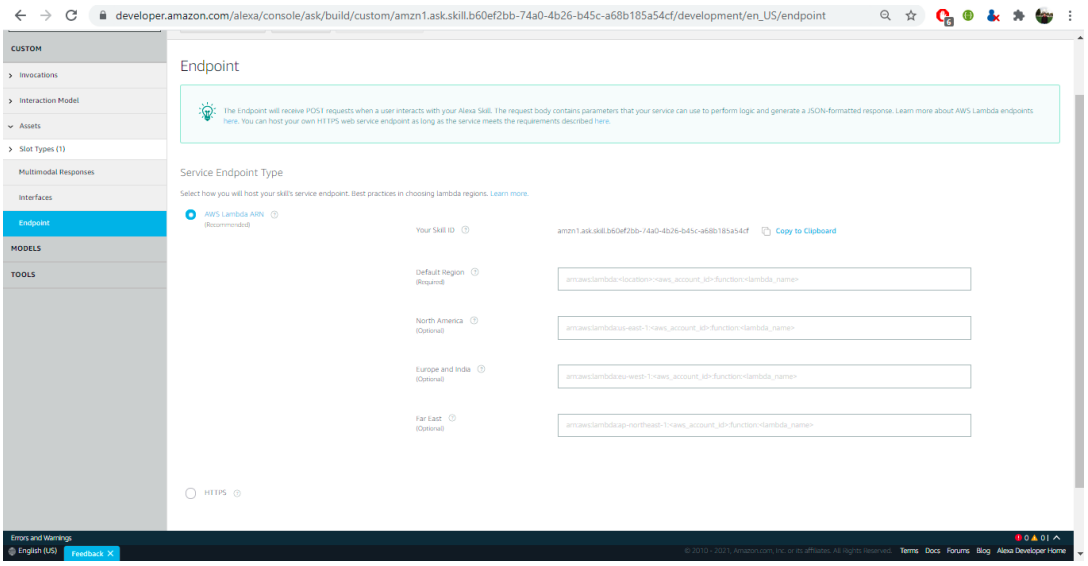




7. Update this CustomSlot in previously added intent.

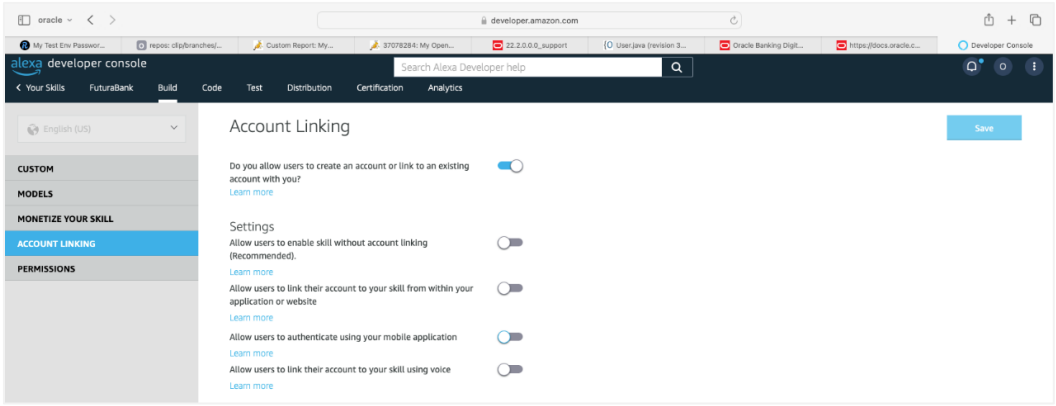


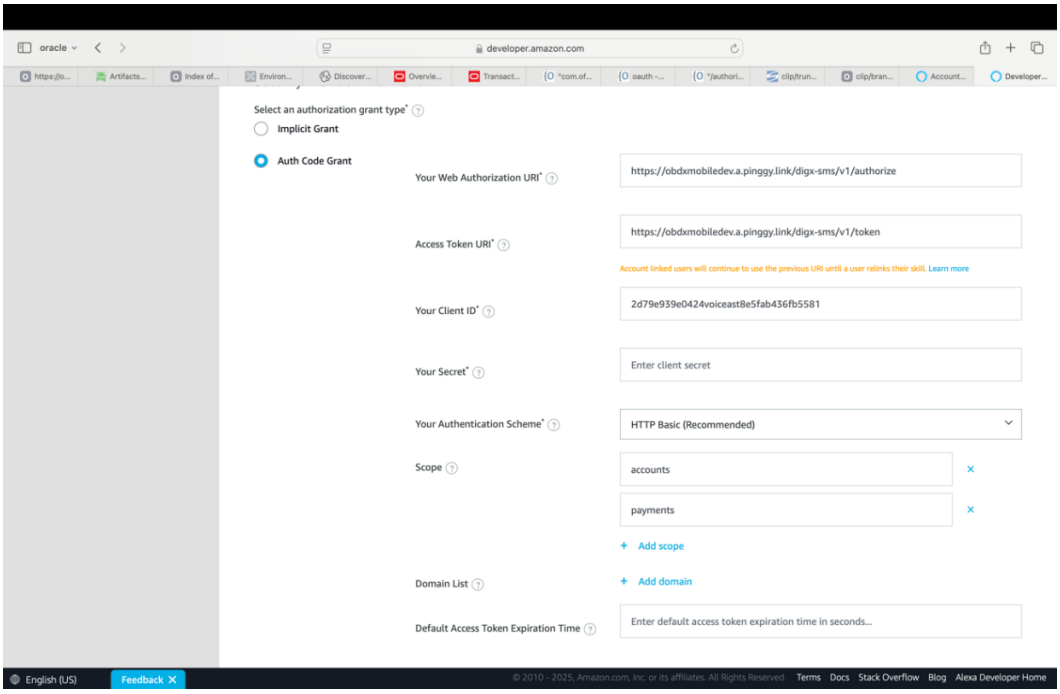
8. Click Build Model in Build tab.
9. Update endpoint and add CA.
10. Note the skill ID.



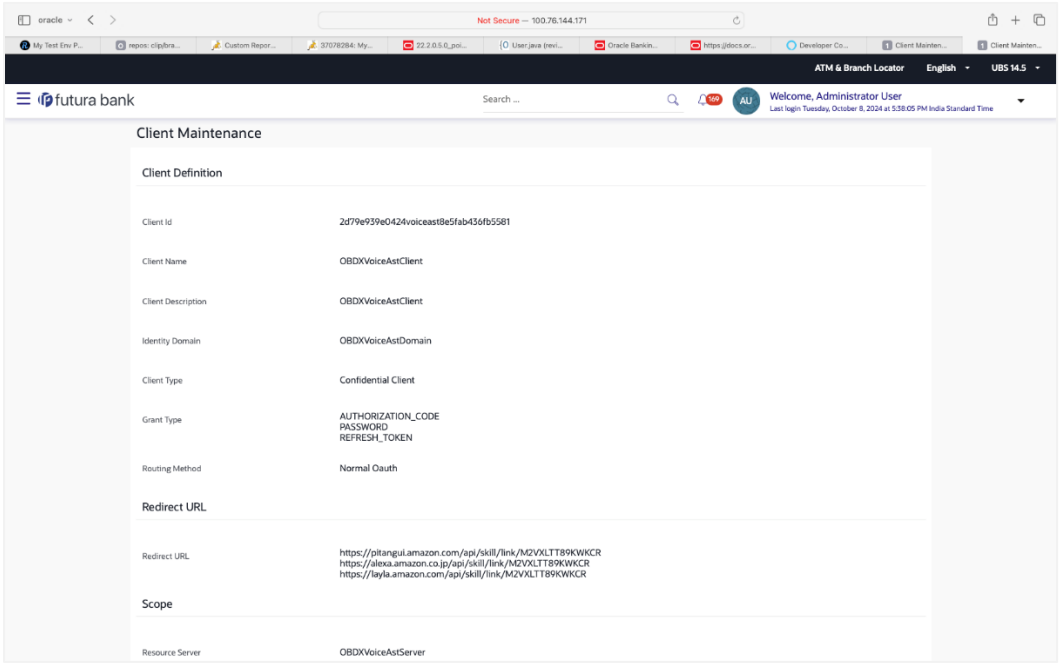
Setup Account Linking for Alexa

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”.

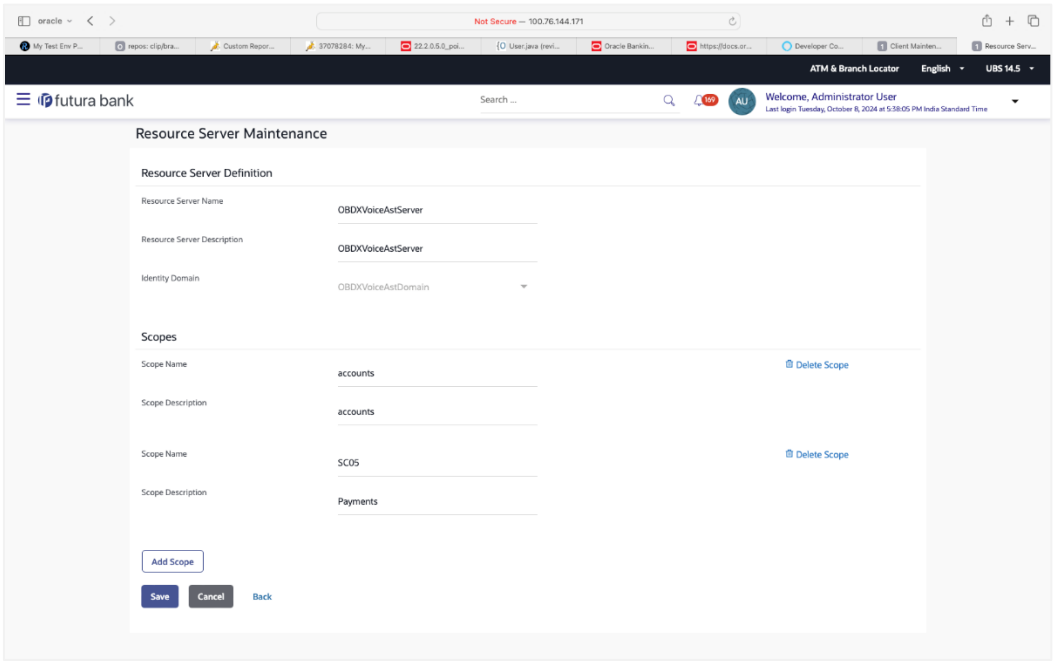




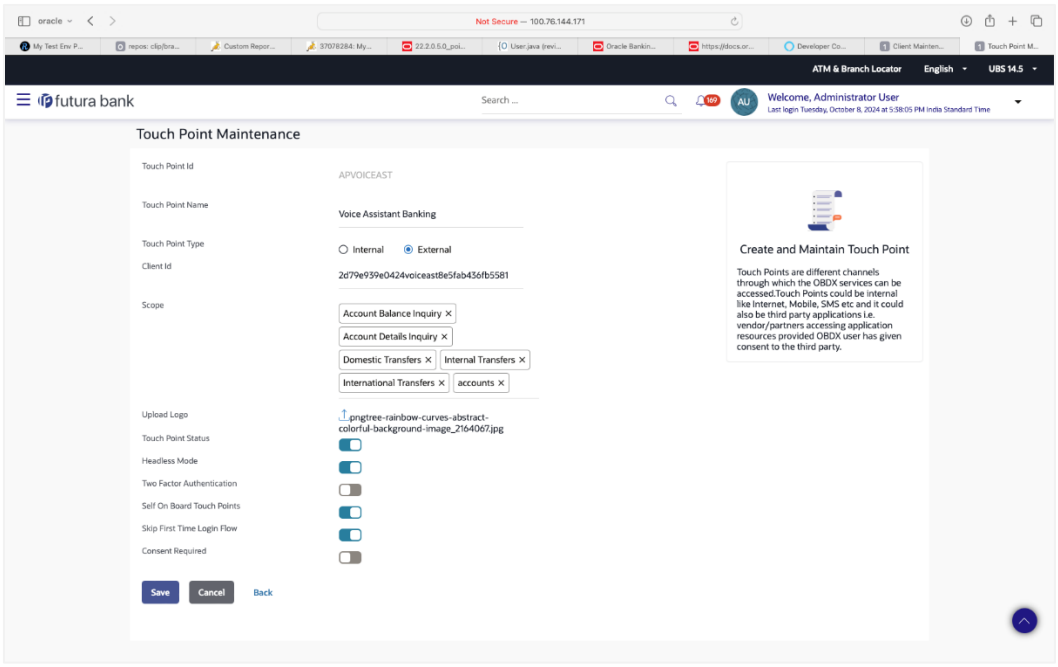
4. Set the client Id for client from client Maintenance page for “OBDXVoiceAstDomain” identity domain client.
5. Click **Edit** and set the client secret for the above client which can be generated from the Client Maintenance Page.



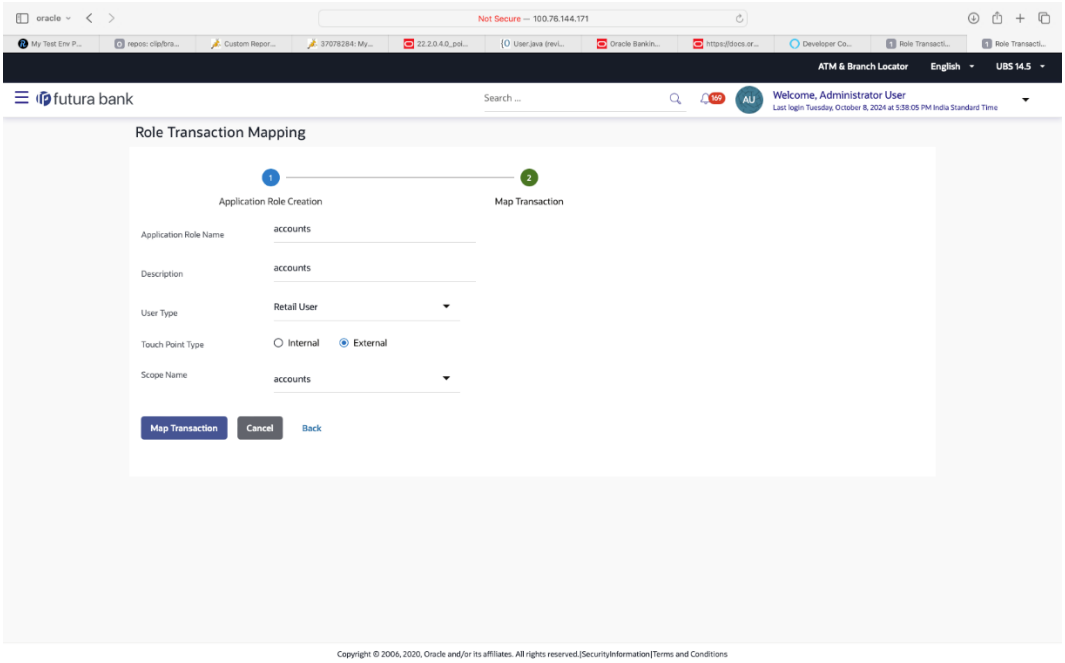
6. Go to Resource Server Maintenance page and select Identity domain to OBDXVoiceAstDomain.
7. Edit the Scope Name of SCO2 to “accounts” and Scope Description to “accounts”.



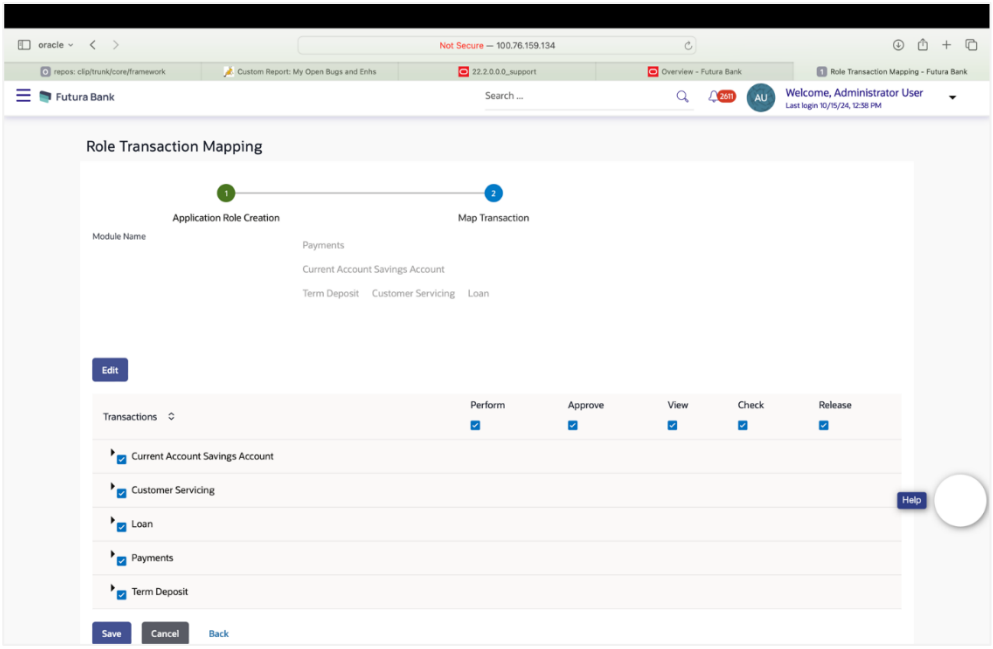
8. Go to Touch Point Maintenance Page and search for “Voice Assistant Banking” touch point name.
9. Add “accounts” scope to the list of scopes and add upload your bank logo.
10. Enable Touch point status.



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



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';

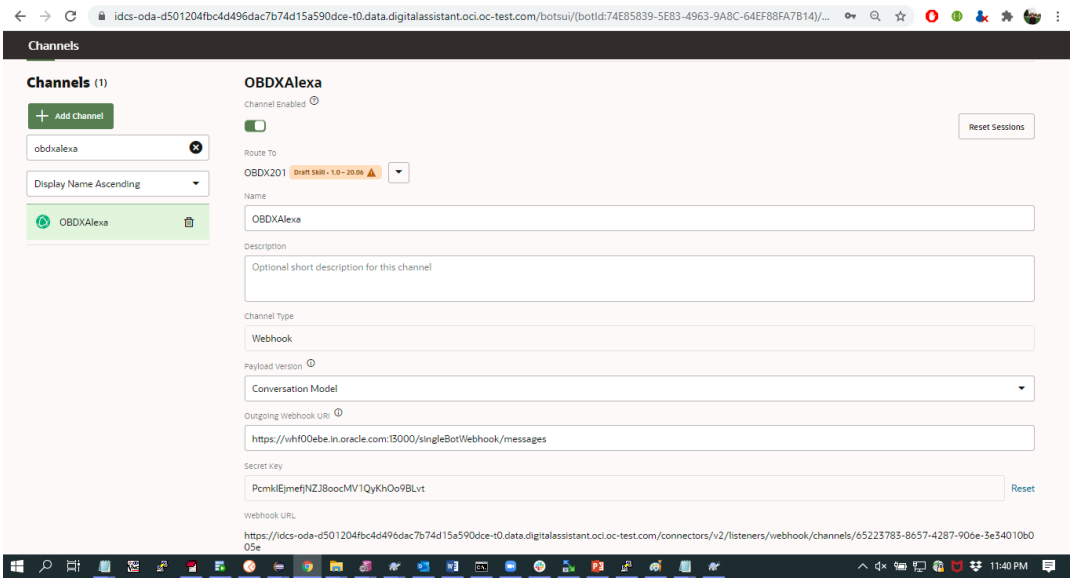
14. Update token_expiry in digx_au table for OBDXVoiceAstDomain domain_id

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

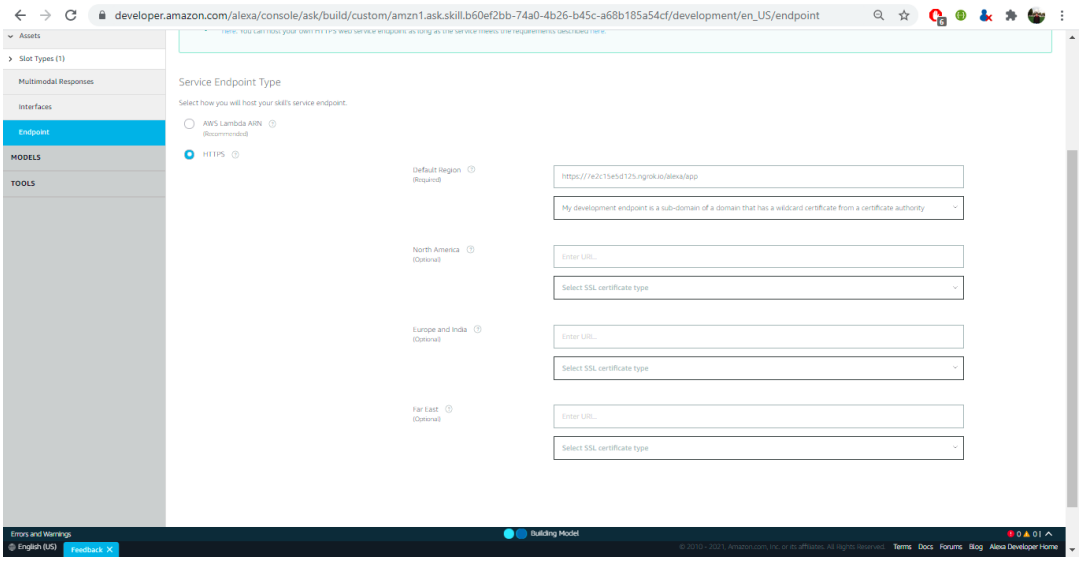
15. Restart the server.

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.

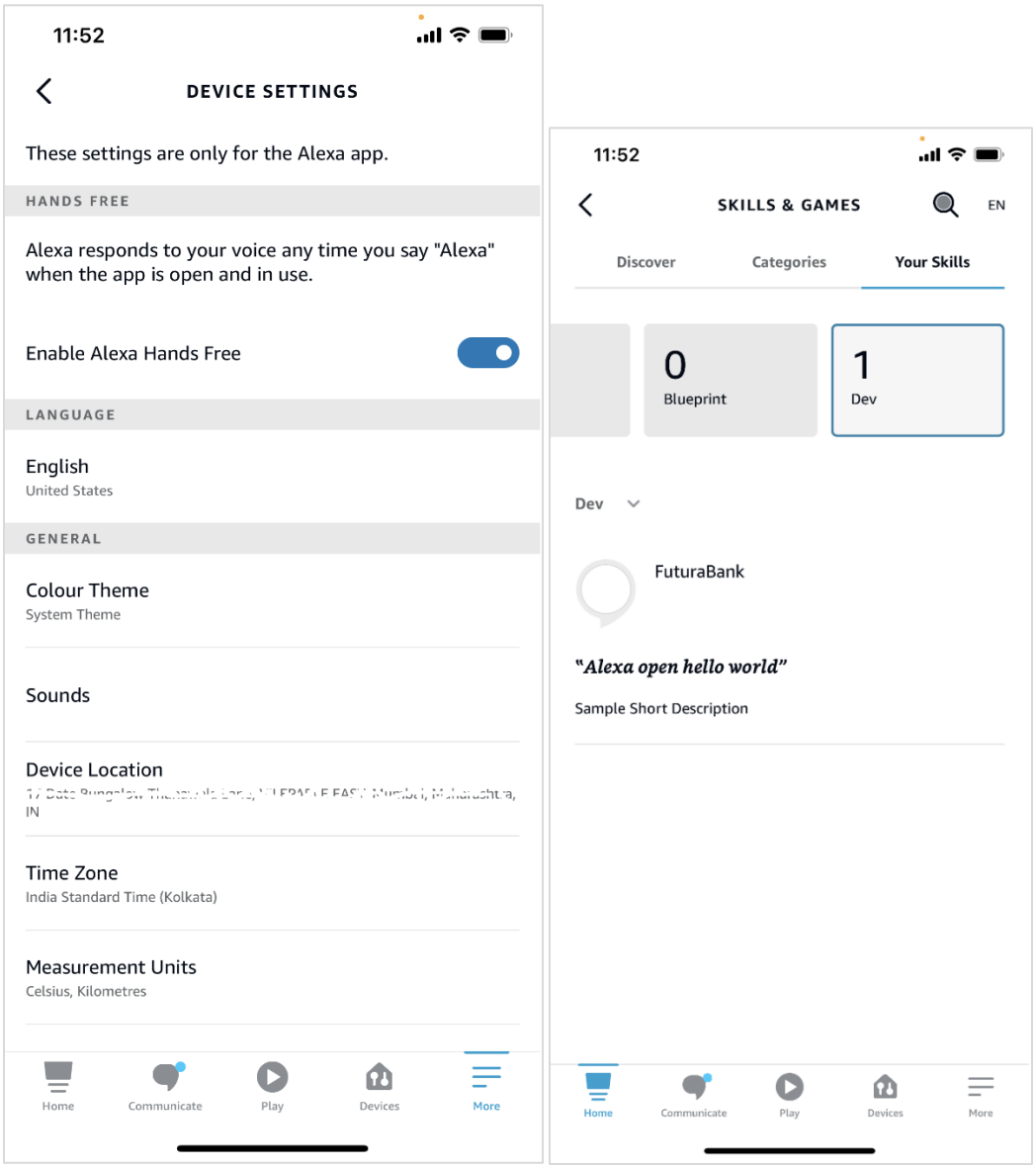


   a. Update amzn_appId, channelSecretKey & channelUrl in service.js
   b. Run nodeJS script
   c. npm install
   d. node index.js
   e. This must be publicly hosted in https URL.
   f. Enter this URL in below screen



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

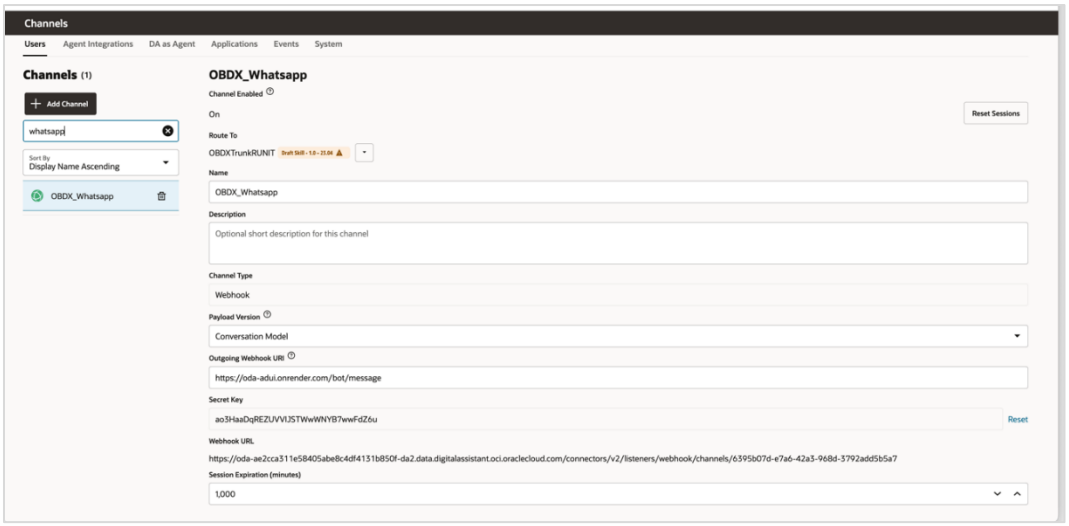
b. Change the language to English US



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

5.6 WhatsApp Channel

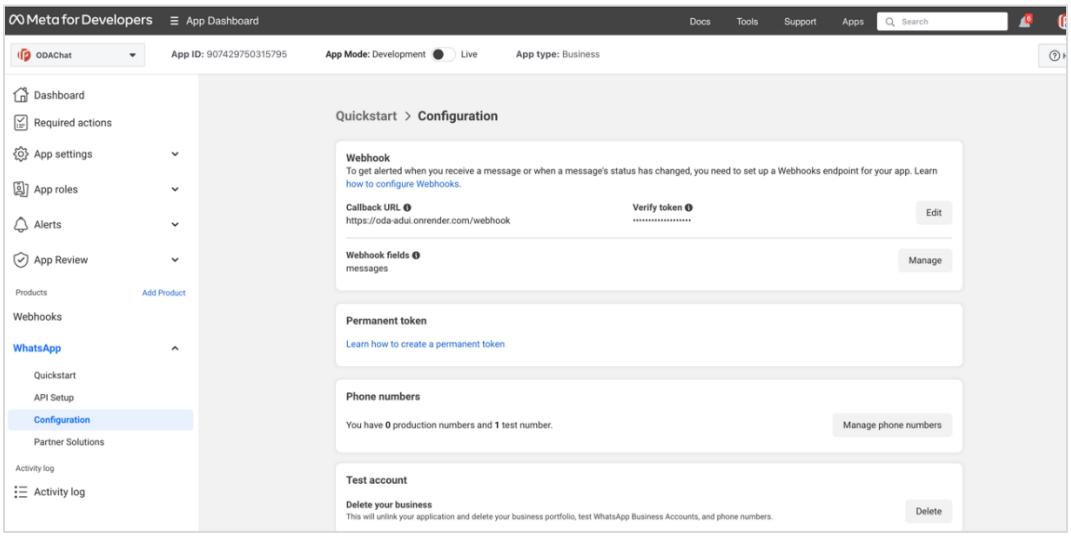
1. Create a channel of type Webhook.
2. Map the Skill to this channel & enable the same.



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.



6. Configuration in OBDX

1. This configuration is applicable for WEB/ANDOIRD/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.

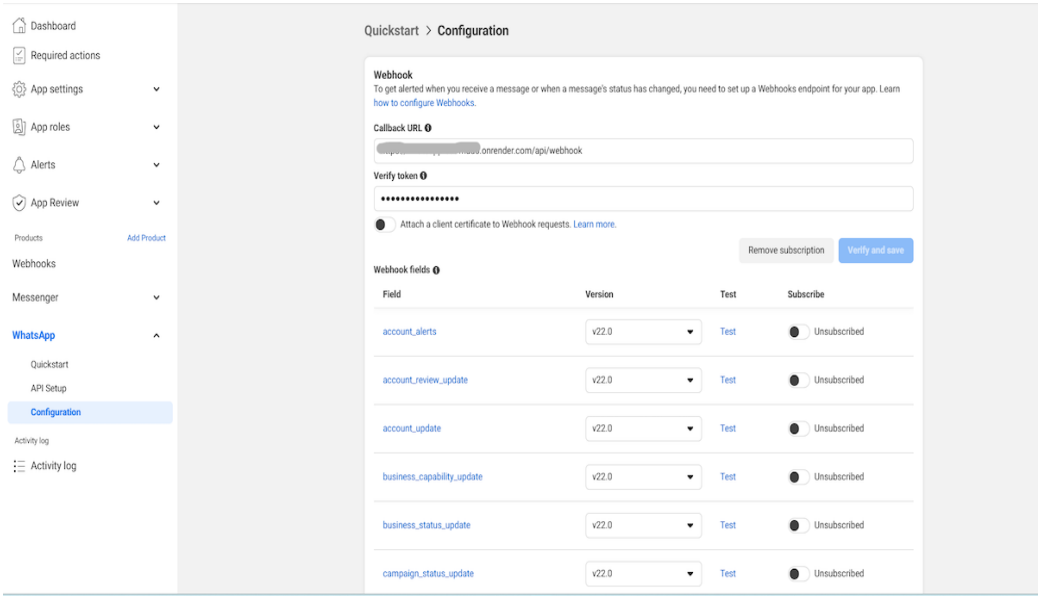
Login will be performed based on OTP.

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



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.