DOCUBOT Integration with OFSLL
Overview and Developer Guide
Oracle Financial Services Lending and Leasing
Release 14.12.0.0.0
Part No. F53373-01
March 2022

ORACLE
Financial Services

Table of Contents

1. OFSLL DOCUBOT Overview and Developer Guide

OFSLL has an extended out of the box support for CHATBOT integration. This provides a new framework for direct user interaction with the system. However, since OFSLL is a back-office system there are additional external components required to be integrated to host and utilize the CHATBOT functionality.

For latest version of this document, refer to https://docs.oracle.com/cd/F53373_01/pdf/refdocs/ofsll_docubot_overview_and_developer_guide.pdf

This document consists of following chapters:

- OFSLL DOCUBOT Overview
- Developer Guide for BOT Customization

Following topics are discussed in OFSLL DOCUBOT Overview chapter:

- Introduction
- Indexing Workflow
- How does search work?
- Third Party Licenses
- Features of BOT
- Launch OFSLL BOT

1.1 Introduction

Currently, OFSLL integration with CHATBOT is supported with some of the functionalities such that end users can search for documentation and / or query and fetch the account related information and/or perform other actions on an account with options presented in CHATBOT menu.

This document outlines the integrated framework and procedures required to implement certain features, but it is not a general-purpose configuration manual.

1.1.1 DocuBot Overview

OFSLL integrated Docubot (Documentation searchable Chabot) is a functionality for guiding product end-users to navigate through release documentation with dynamic search capability along with other key features like access to latest release documents, module specific documentation, index glossary and so on.

This serves as knowledge repository and one point reference for information related to product usability, process workflow, installation, administration of all product release till date.

The Documentation ChatBot or DocuBot - hereafter is referred to as ‘BOT’ in the document.

1.1.2 About

The documentation search engine adapted in BOT facilitates for all types of data search including textual, alphanumeric, numeric, keyword, phrases, and sentences. This is an 'Elastic Search' and uses a data structure called 'Inverted Index which is designed for fast and full-text searches. An inverted index lists every unique word that appears in any document and identifies all of the documents in which the word is present.

The advantage of using elastic search is the speed, scalability and its ability to index most format of content.

1.1.3 Purpose

The purpose of this document is to demonstrate the capability of OFSLL BOT in handling documentation search requirements by integrating with Oracle Digital Assistant (ODA). This document is intended to detail the usability features and also to serve as a developer guide to understand the configuration procedures. However, the features and options presented are provided only as a sample and needs further customization based on requirements.

1.1.4 Audience

In general, this document is intended to all those parties and decision makers who are interested to know about OFSLL BOT integrated framework. The configuration sections are intended for system administrators, consulting and implementation teams who deploy customized solutions for customer.

1.1.5 Accessibility

The OFSLL BOT integrated framework is supported from OFSLL 14.12.0.0.0 release.

BOT is agnostic of which self-service site / portal is used to provide access and interface to the users for help documentation.

1.1.6 Access

Currently the framework supports basic authentication (not OAUTH). User Management and authentication needs to be handled as part of the implementation.

1.1.7 Pre-requisites

Following are the pre-requisites:

- The BOT is designed to work in ODA framework (platform version 21.02). The configuration is to be done as detailed in Developer Guide for BOT Customization section.
- Also the ODA Server Environment has to be licensed separately. For more information, refer to https://www.oracle.com/in/chatbots/digital-assistant-platform/
- Need to have release specific pre-indexed file for elastic search to work.
- Adequate space to store the indexed file directories in the respective folders.
- WebLogic server for deployment of war file (OracleFSLLChatBot.war).
- The parameters in 'Channel.Properties' file are to be configured before creating and deploying the .war file (OracleFSLLChatBot.war). For details, refer to 'BOT Configuration' section.

1.2 Indexing Workflow

The elastic search for OFSLL BOT requires pre-indexing of content. Hence, indexing is done for 14.12.0.0.0 release documents. The indexing process is done automatically using the third-party plugins such as Apache Lucene and Jsoup to identify unique keywords in HTML files. This generates indexed files which serves as common directory for searched keyword and the file instance where it exists.

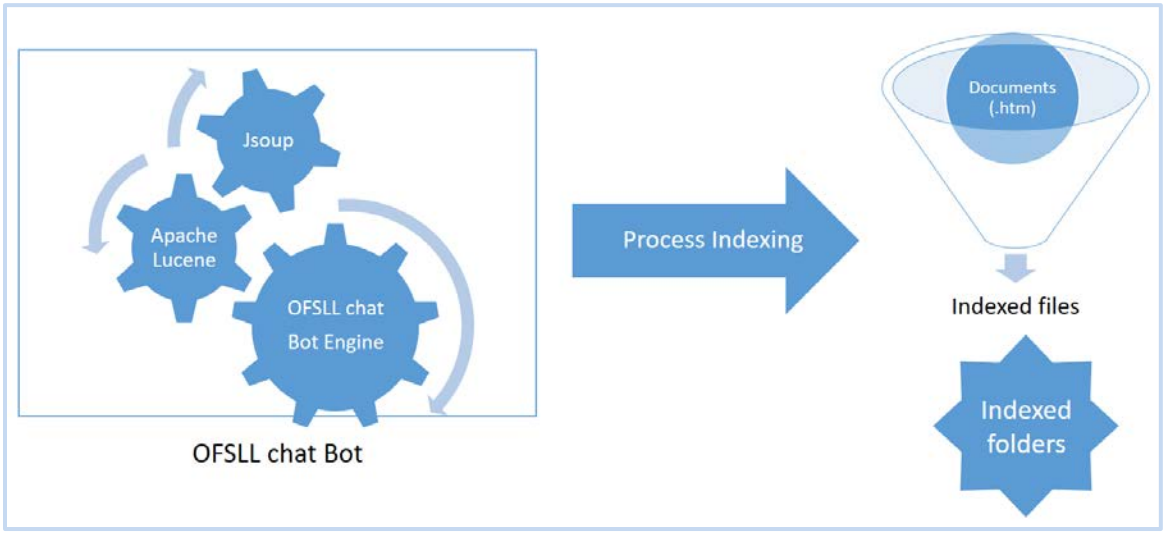
For more information on third-party plugins used, refer to 'Third Party Licenses' section.

1.2.1 How is Indexing Done?

Apache Lucene manages an index over a dynamic collection of documents and provides very rapid updates to the index as and when documents are either added or removed from the collection. An index may store a heterogeneous set of documents.

Index in OFSLL BOT is handled by feeding all the release related html files. These htmls files are indexed both as single page reference and as well as at topic level using the hash tag to which it is mapped.

The workflow indicated below illustrates on how 3rd party Apache Lucene engine creates indexed files and stores in respective release specific folder.



1.2.2 Release Specific Indexing

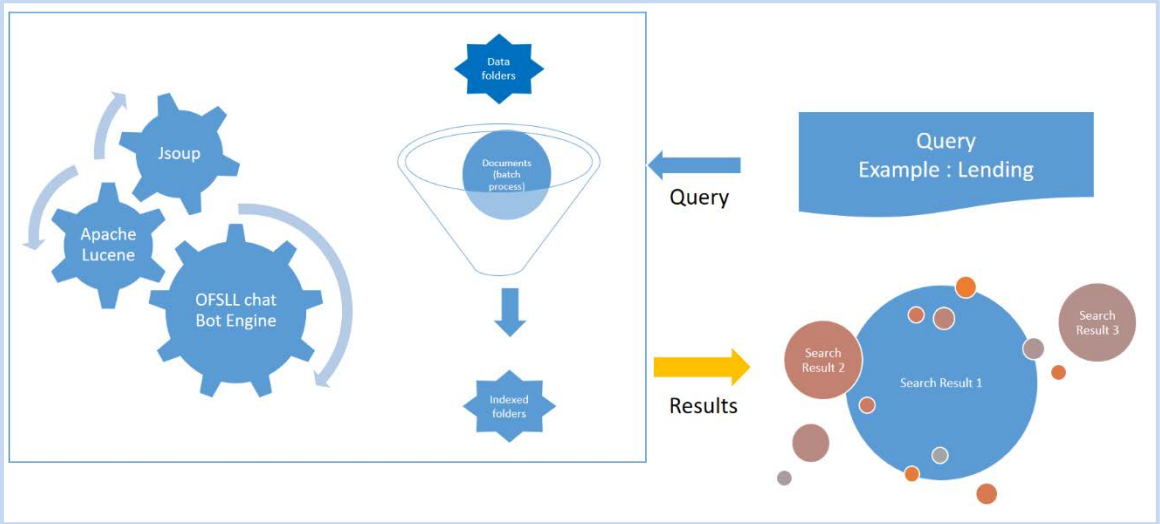
Indexing is done for the following release of OFSLL and indexed files are provided in respective folder. The mapping of Release number v/s Folder name and Part Number is indicated below:

Release No	Folder Name	Part Number
14.12.0.0.0	14.12	F53373_01

1.3 How does Search Work?

On initiating the Search, the keyword is searched in the index directory and a URL link is returned in two ways for the specific release number mapped. The search results are provided in both the below combinations:

- Page(s) link in which the Search keyword is present
- Direct heading link in which the Search keyword is present. This is fetched using hash tag reference for the topic.



1.4 Third Party Licenses

OFSLL BOT uses the following third party licenses:

- Apache Lucene, Version: 8.10.1

  The Apache Software Foundation, Technology: Lucene, Version: 8.10.1

  Files used (below are part of Apache Lucene 8.10.1)

  Lucene Core (8.10.1)

  Lucene query parser (8.10.1)

- JSOUP 1.14.3

  Jsoup is a Java library for working with real-world HTML.

  It provides a very convenient API for fetching URLs and extracting and manipulating data, using the best of HTML5 DOM methods and CSS selectors.

  jsoup implements the WHATWG HTML5 specification, and parses HTML to the same DOM as modern browsers do.

  scrape and parse HTML from a URL, file, or string

  find and extract data, using DOM traversal or CSS selectors

  manipulate the HTML elements, attributes, and text

  The purpose of using Jsoup in chatbot is to read the html elements <tags> <href> and use it as a added part of indexing

  Link : https://jsoup.org

For detailed information, refer to product licensing guide.

1.5 Features of BOT

Following are the unique features of OFSLL BOT:

- Readily available navigation links to the following:
  - Link to all Release documentation
  - Dynamic Document Search option
  - Link to currently mapped Product Release notes
  - Listing of Product Module / Classified Guides
  - Link to list of indexed Keywords
  - Link to Getting Started Video gallery
  - Link to Release Highlights
- Intuitive Menu options:
  - Option to clear chat data
  - Speech Conversion – Voice based Input
  - Personalization of BOT interface

1.5.1 Support of Text and Voice Based inputs

The BOT can support both Text and Voice based inputs to find information. This attempts to comply with multiple accessibility options.

The BOT is enabled with voice based inputs where in voice commands are accepted as input equivalent to typing or clicks. This option works on clicking the Mic button.

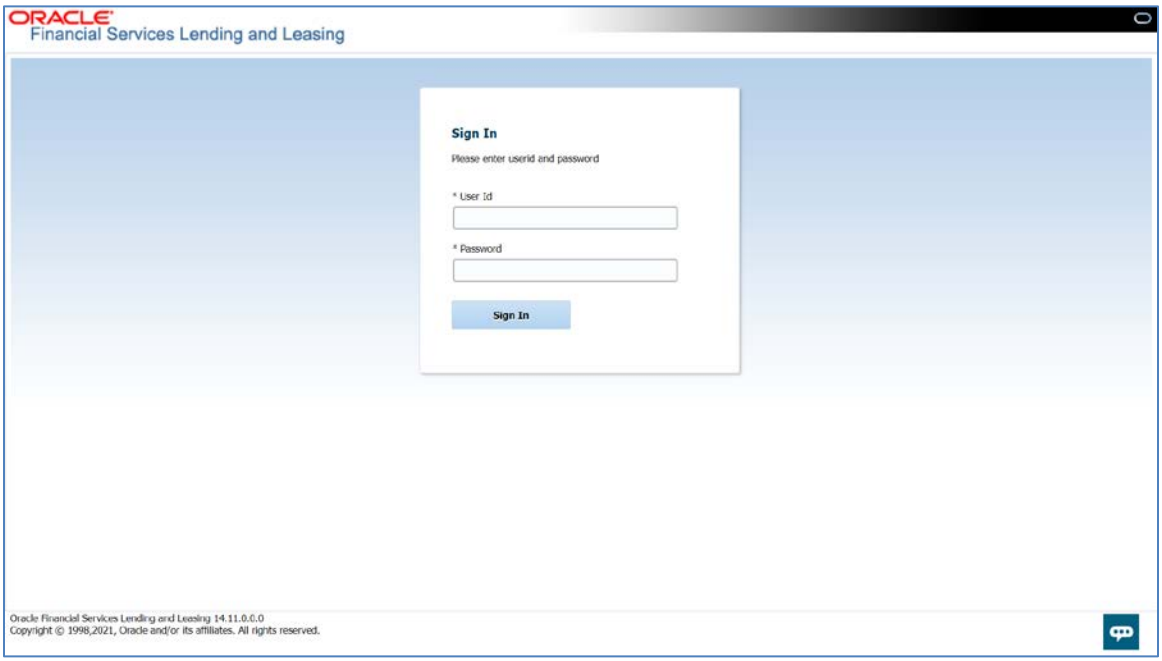
During text based input, the response is provided in the BOT interface. In a voice based input, the response is provided in both voice based response and BOT response simultaneously.

However, note that voice based input does not support to open a URL (link) reference.

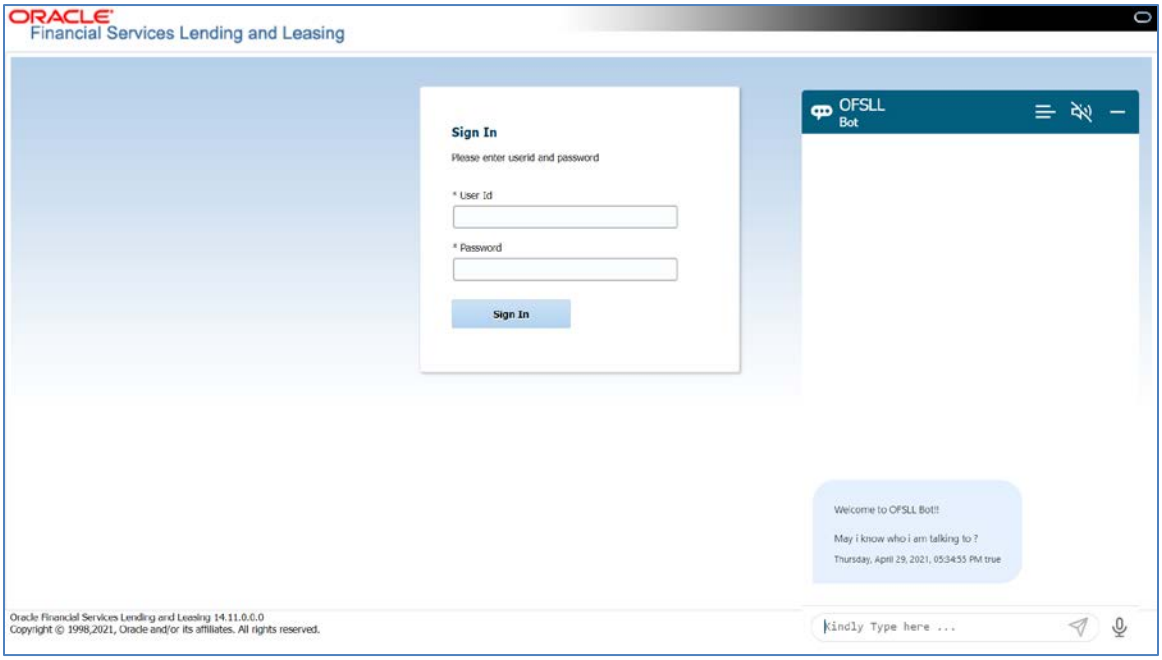
1.6 Launch OFSLL BOT

OFSLL BOT can either be in enabled or disabled status by default depending on the weblogic csf configuration (refer section 2.5 in this document). If enabled, on launch of OFSLL application the BOT is available at right bottom corner.

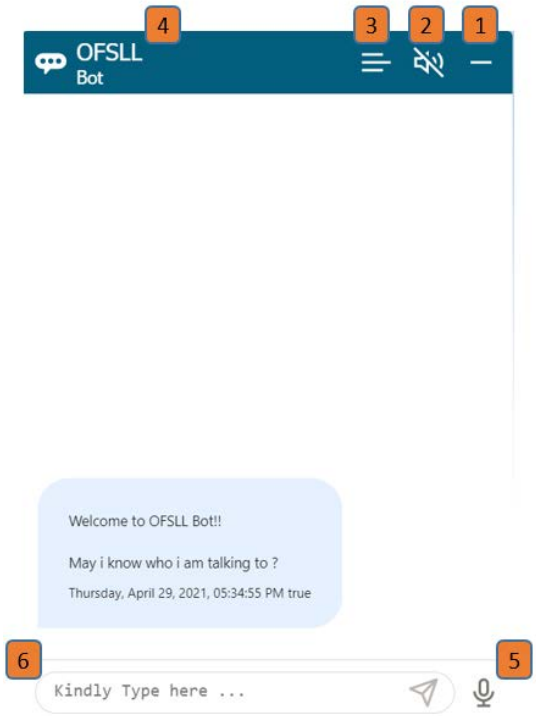
Note: Before you being, ensure to perform the required configuration as detailed in 'Developer Guide for BOT Customization' chapter.



On clicking the BOT icon, the welcome message is as displayed:



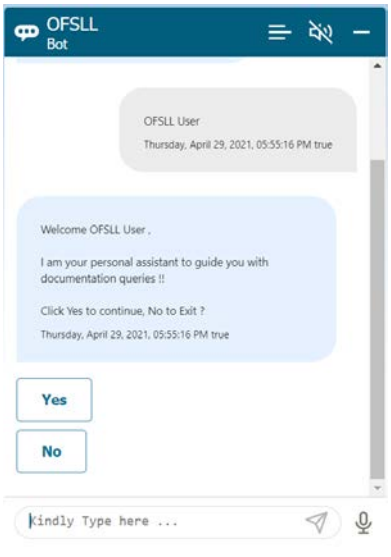
1.6.1 BOT UI Elements

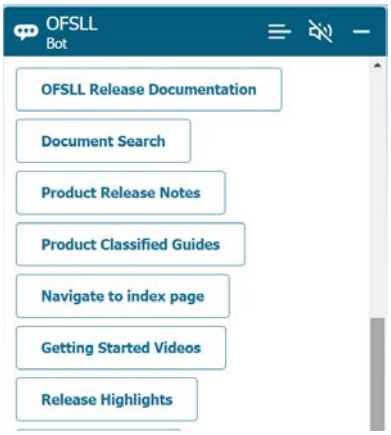
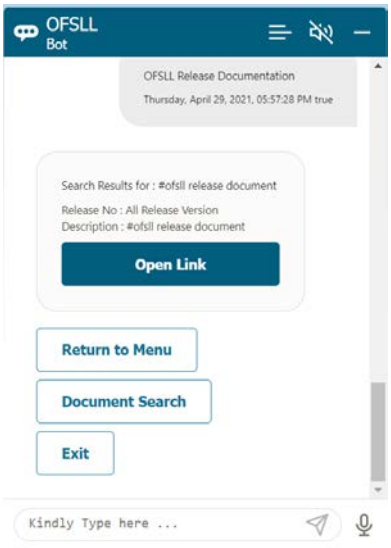


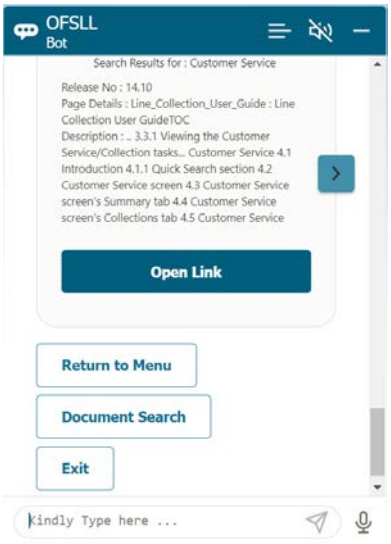
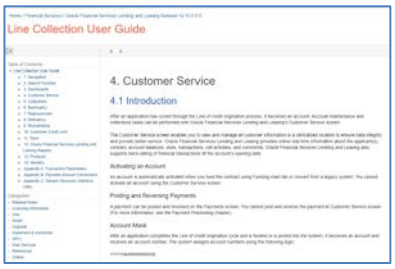
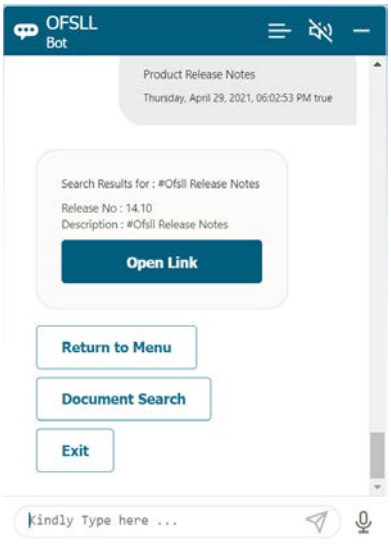
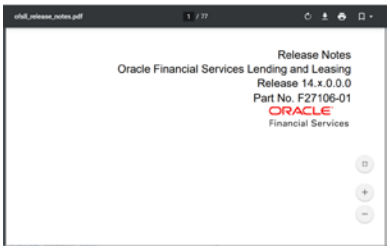
Sl.No	Option	View / Action
1	Minimize	Minimize BOT window
2	Speaker output	Enable BOT in speaker mode
3	Clear chat	Clear all messages in the BOT
4	Customized label	Customization for title label is detailed in Bot Customization section.
5	Mic Input	Enable Mic for voice based input
6	Text Input	Enter search string using keyboard

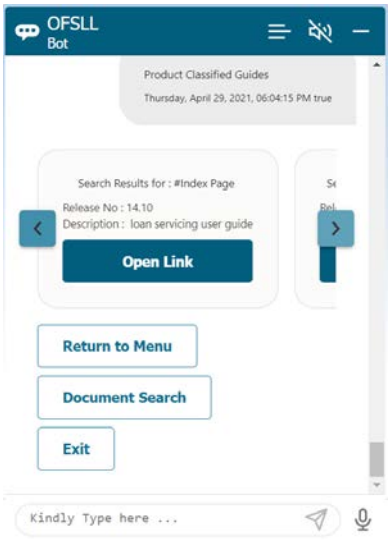
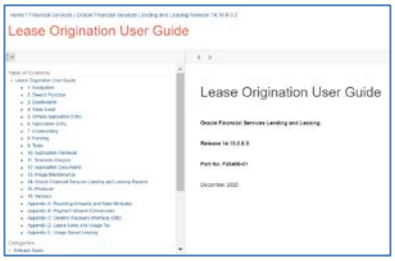
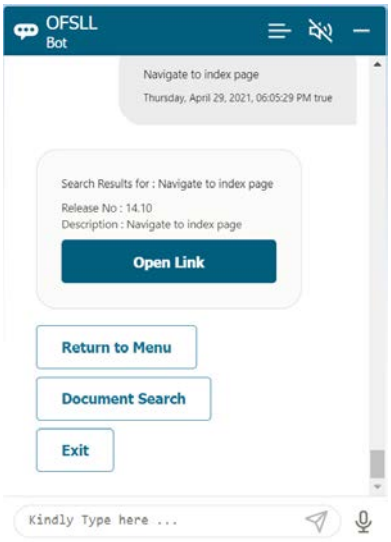
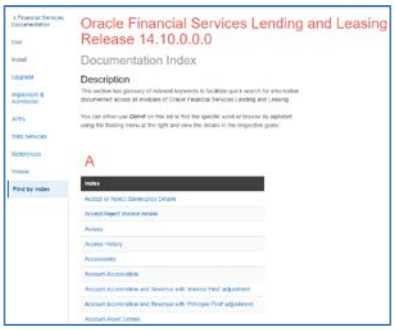
1.6.2 BOT Usability Workflow

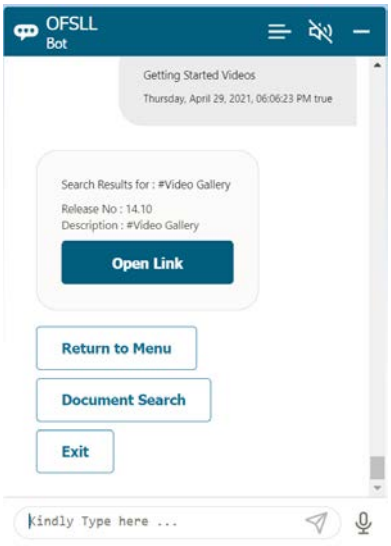
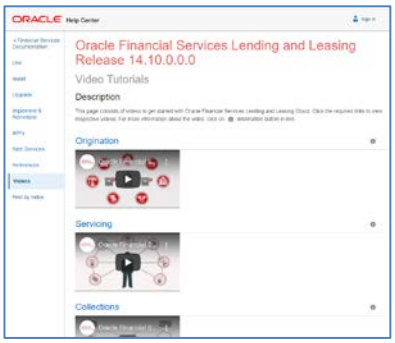
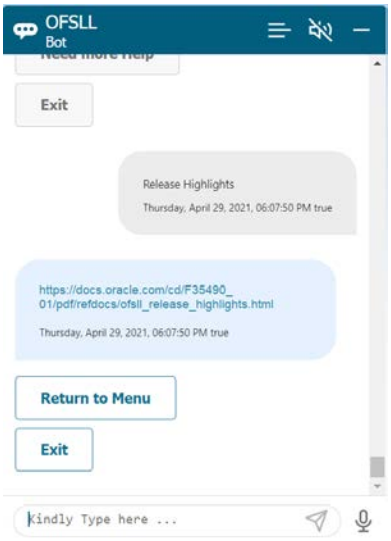
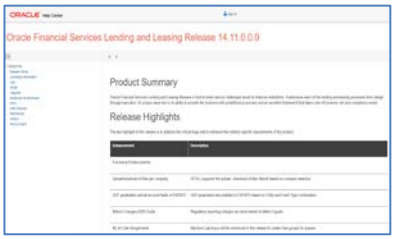
Below is a simple sequence of user interaction with OFSLL BOT.

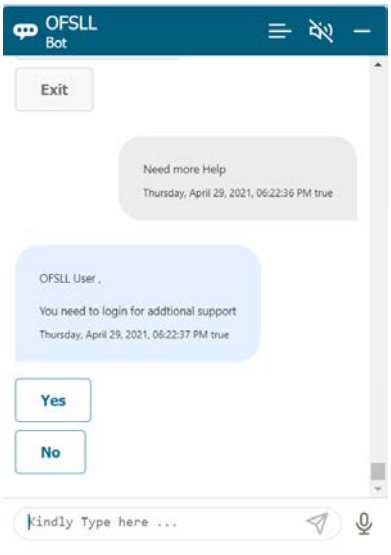
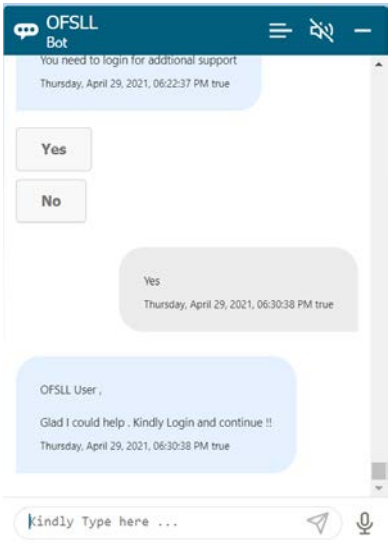
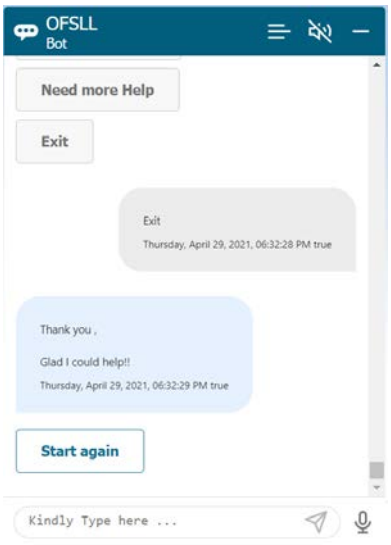
Sl.No	Action	BOT response	
1	Enter your name	BOT registers and provides a confirmation message to continue.	

2	Clicking on ‘Yes’	BOT presents list of documentation resorces for selection.	
	Clicking on ‘No’	Docubot displays an exit message. However, there is option to start again.	
3	Clicking on OFSLL Release Documentation	 Clicking on the Open Link, BOT displays the home screen with access to all release documenation libraries.	

4	Clicking on Documentation Search	 BOT displays dynamic results in horizontal navigating cards based on ranks from the list of documents which were indexed. Clicking on the Open Link, BOT displays the result page which contains the search keyword.	
5	Clicking on Product Release Notes	 Clicking on the Open Link, BOT displays the release notes of current release.	

6	Clicking on Product Classified Guides	 Clicking on the Open Link, BOT displays the module specific guide.	
7	Clicking on Navigate to index page	 Clicking on the Open Link, BOT displays the Find by Index page which consists of all indexed keywords in documentation.	

8	Clicking on Getting Started Videos	 Clicking on the Open Link, BOT displays the video gallery link with getting started videos.	
9	Clicking on Release Highlights	 Clicking on the link, BOT displays the Release Highlights page which consists of new developments made to the current version of the product.	

10	Clicking on Need More Help		
11	Clicking on Exit	BOT displays an exit message. However, there is option to start again.	

2. Developer Guide for BOT Customization

This section of the document intends to help you to set up and configure Oracle Digital Assistant (ODA) 'ASK' with the sample OFSLL wrapper. However, the instructions are provided in brief and for any additional information, contact Oracle Financial Services Lending and Leasing Product Engineering team.

Note: Currently this framework supports basic authentication provided by OFSLL REST service. OAUTH authentication is not supported. Additionally, OBDX (Oracle Banking Digital Experience) can be integrated for user authentication purpose. For more information, refer to documentation at https://docs.oracle.com/cd/E97825_01/webhelp/Content/obdx/core/authentn/authntctn.htm

Following topics are discussed in this section:

2.1 Pre-requisites

Following are the mandatory pre-requisites:

- OFSLL being a back-office system with limited capability, the following external components are to be integrated in a single framework:
  - ODA or Oracle Digital Assistant is a platform that allows to create and deploy digital assistants, which are AI-driven interfaces that help users accomplish a variety of tasks in natural language conversations.
  - OBDX or Oracle Banking Digital Experience as a Application Launching portal and for multi-factor authentication.
    --or--
  - Any 3rd party web application or customer self-service portal or lenders/financial services website to launch OFSLL BOT. In this case user authentication related integration needs to be handled as part of the implementation activity.
- Users need to have a capability to develop customized workflows using ODA development framework. A brief introduction is explained in 'ODA – Dialog Flow Development' section.
- User need to have a good understanding of OFSLL REST services and should be able to customize it accordingly.
- User needs to be well versed with OFSLL wrapper customization as explained in 'OFSLL Wrapper customization' section.

2.2 OFSLL Wrapper customization

Note: From the current release onwards, no additional jar file needs to be added since 'Maven – Pom.xml' based model has been implemented.

Follow the below steps for OFSLL wrapper customization:

1. Import project into eclipse and modify channel.Properties to update below properties

   ofsll.baseURL = <OFSLL REST service base URL <http://<host>:<port>/OfsllRestWS/service/api/resources>>

   ofsll.username = <OFSLL username>

   ofsll.password = <OFSLL pass>

   ofsll.suffix = htm

   ofsll.otmHttpUrl=https://docs.oracle.com/cd/

   ofsll.fIndex=/findex.htm

   ofsll.index=index.htm

   ofsll.video=/videos.htm

   ofsll.ofsllReleaseNotes=/pdf/refdocs/ofsll_release_notes.pdf

   ofsll.ofsllReleaseDoc=https://docs.oracle.com/en/industries/financial-services/financial-lending-leasing/index.html

   ofsll.splitSeperator==

   ofsll.maxHitsResults=<max number of results returned>

   ofsll.indexDir = <Release index directory path of server >

   ofsll.releaseVersionUrl= <Release Part number>

   ofsll.releaseNo=<Release No>

   ofsll.releaseHighlights=/pdf/refdocs/release_highlights.htm

2. To add any new service modify com.ofss.ofsll.chatbot.restclient.ChatRestClient.java file.
   - Inside ChatRestClient Class add a new method with required actions
   - Add supporting JAXB files
   - Use the available supporting methods -- readInputStream, setChatBotResponse, createConnection, stringToJaxb etc.

Example for document search functionality is indicated below:

```
@Consumes(MediaType.APPLICATION_JSON)
        @Produces(MediaType.APPLICATION_JSON)
        @POST
        @Path("/lucenesearch")
         public Response lucenesearch(ODARequestDTO ibcsRequest) throws
IOException {
            final IChatbotAssembler chatbotAssembler =
ChatbotAssemblerFactory.getInstance().getChatbotAssembler("ODA");
            final HashMap < String,
            Object > map = (HashMap < String, Object > ) ibcsRequest.getProperties();
            String searchQuery = "";
            Properties prop = new Properties();
            try (InputStream propertiesFile =
this.getClass().getClassLoader().getResourceAsStream("channel.properties")) {
              prop.load(propertiesFile);
            }
            if (map != null && map.containsKey("query")) {
              searchQuery = (String) map.get("query");
            }
            ResponseDTO ibcsResponse = null;
            try {

              ChatbotResponseDTO chatbotResponse = new ChatbotResponseDTO();
              String indexDirPath =
prop.getProperty("ofsll.indexDir")+prop.getProperty("ofsll.releaseNo");
              String releaseVersionUrl = prop.getProperty("ofsll.releaseVersionUrl");
              String urlPrefix = prop.getProperty("ofsll.otmHttpUrl");
              String splitSeperator = prop.getProperty("ofsll.splitSeperator");
              String releaseNo = prop.getProperty("ofsll.releaseNo");
              String urlPrefixPath = urlPrefix + releaseVersionUrl;
              String findexPath = prop.getProperty("ofsll.fIndex");
              String indexPath = prop.getProperty("ofsll.index");
              String videoPath = prop.getProperty("ofsll.video");
              String ofsllReleaseNotesPath = prop.getProperty("ofsll.ofsllReleaseNotes");
              String ofsllReleaseDocPath = prop.getProperty("ofsll.ofsllReleaseDoc");
              Integer maxHitsResults =
Integer.parseInt(prop.getProperty("ofsll.maxHitsResults"));
              File fileIndexDirPath = new File(indexDirPath);
              LuceneSearchHighlighter luceneSearchHighlighter = new
LuceneSearchHighlighter();
              List<String> fileList = new ArrayList <> ();

              if ((searchQuery.toLowerCase().trim().contains("#ofsll release document")) ||
(searchQuery.toLowerCase().trim().contains("navigate to index page")) ||
(searchQuery.toLowerCase().trim().contains("#video gallery")) ||
(searchQuery.toLowerCase().trim().contains("#ofsll release notes")) ||
(searchQuery.toLowerCase().trim().contains("#index page"))) {
```

```
                if ((searchQuery.toLowerCase().trim().contains("#ofsll release document")))
{
                    releaseNo="All Release Version";
                  fileList.add(searchQuery + splitSeperator + ofsllReleaseDocPath +
splitSeperator+searchQuery+ splitSeperator+releaseNo);
                }
                if ((searchQuery.toLowerCase().trim().contains("navigate to index page"))) {
                  fileList.add(searchQuery + splitSeperator + urlPrefixPath + findexPath +
splitSeperator+searchQuery+ splitSeperator+releaseNo);
                }
                if ((searchQuery.toLowerCase().trim().contains("#index page"))) {
                  searchQuery = indexPath;
                  fileList = luceneSearchHighlighter.searchsinglepage(fileIndexDirPath,
searchQuery, maxHitsResults, splitSeperator);
                }
                if ((searchQuery.toLowerCase().trim().contains("#video gallery"))) {
                  fileList.add(searchQuery + splitSeperator + urlPrefixPath + videoPath +
splitSeperator+searchQuery+ splitSeperator+releaseNo);
                }
                if ((searchQuery.toLowerCase().trim().contains("#ofsll release notes"))) {
                  fileList.add(searchQuery + splitSeperator + urlPrefixPath +
ofsllReleaseNotesPath + splitSeperator+searchQuery+ splitSeperator+releaseNo);
                }
              } else {
                searchQuery = searchQuery.replaceAll("#", "");
                fileList = luceneSearchHighlighter.search(fileIndexDirPath, searchQuery,
maxHitsResults, splitSeperator);
              }
              String serviceOutputForChatBot = "";
              for (String obj: fileList) {
                if (serviceOutputForChatBot == "") {
                  serviceOutputForChatBot = obj.replace("\\", "/");
                } else {
                  serviceOutputForChatBot = serviceOutputForChatBot + "\n---\n" +
obj.replace("\\", "/");
                }
              }
              if (fileList.isEmpty()) {
                String errorOutputForChatBot = "Search is not found for : " + searchQuery;
                setChatBotResponse("failure", errorOutputForChatBot, chatbotResponse,
"response", "request");
              } else {
                List < String > srhchoices = new ArrayList < >();
                for (String obj: fileList) {
                  srhchoices.add(obj.replace("\\", "/"));
                }
                setChatBotResponse("success", srhchoices, chatbotResponse, "acc_srh",
"acc_srh");
```

```
                }
                ibcsResponse =
chatbotAssembler.fromChatbotResponseDTO((RequestDTO) ibcsRequest,
chatbotResponse);
            } catch(Exception e) {
              LOGGER.log(Level.SEVERE, e.getMessage());
            }
            return Response.status(Response.Status.OK).entity((Object)
this.buildResponse((Object) ibcsResponse)).build();
        }
```

3. Export project as war file.
4. Deploy <WL_Home>/wlserver/common/deployable-libraries/jax-rs-2.0.war as Library on weblogic.
5. Deploy generated WAR (OracleFSLLChatBot.war) in step 3 onto weblogic server.
6. Note down base service URL that is required while publishing in ODA.
   Example : http://<host>:<port>/ofsll/v1/fulfillment

2.3 ODA – Dialog Flow Development

Each menu option displayed in BOT are configured as an 'Intent' which is configured to perform a specific function or otherwise call a REST service in OFSLL.

In-order to achieve a sequence of menu options, dialog flow development is required to be performed in ODA Oracle Digital Assistant. Following is a quick overview of steps involved:

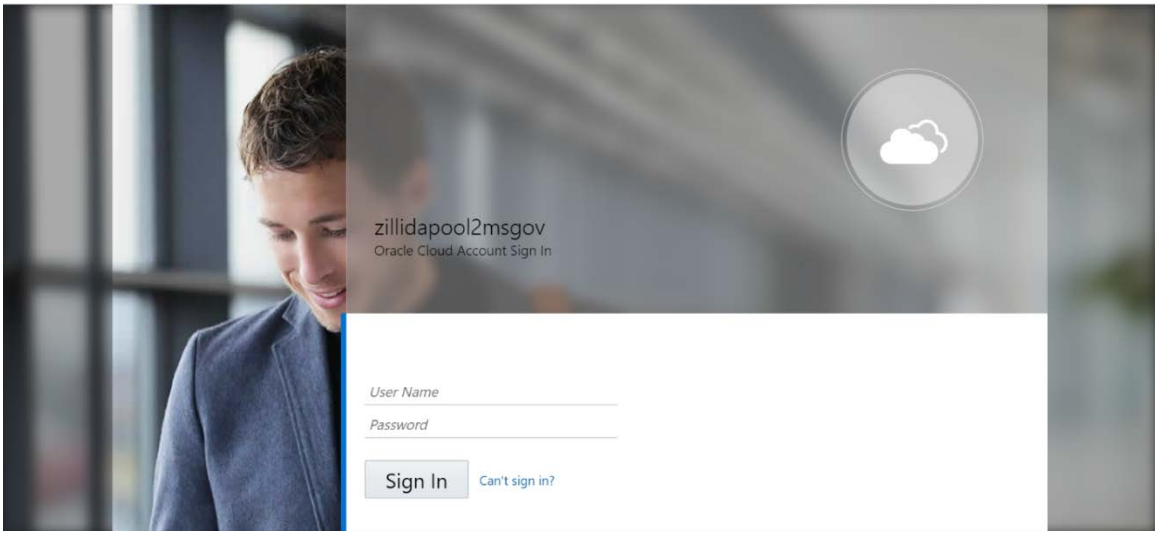
- Login
- Creating Skill / Digital Assistant
- Defining Entity
- Adding Intents
- Updating Bot flow using Yaml
- Adding OFSLL REST service
- Configuring Channel for Publishing
- Publishing

It is recommended to refer to ODA documentation for detailed information - https://docs.oracle.com/en/cloud/paas/digital-assistant/index.html

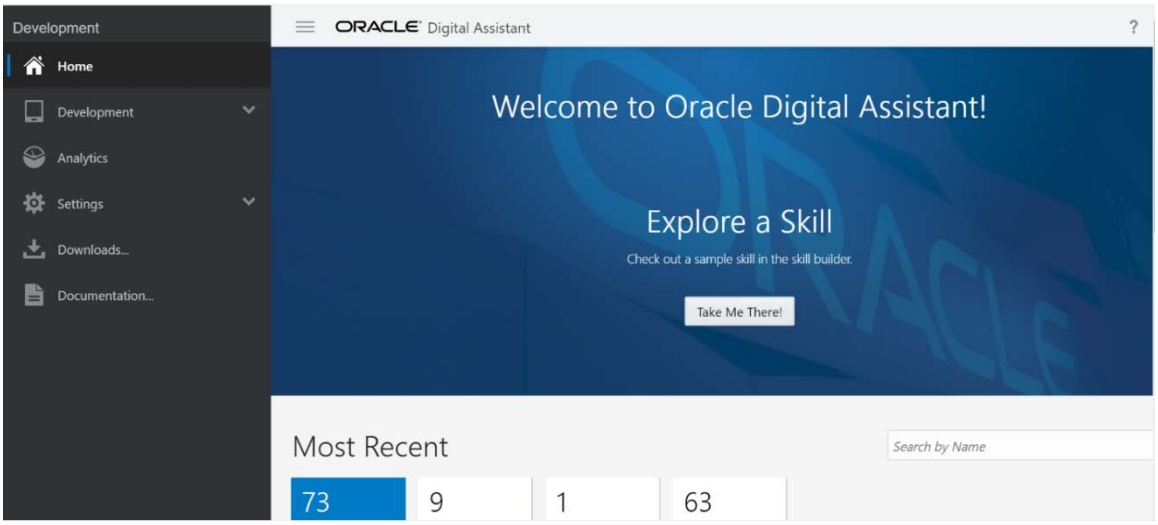
In the ODA - dialog flow development, you can either create new or import the given sample available in path – <release.zip>\LL\release\14_x_0_0_0\ws_as\ChatBot\documentation-bot

The sequence of flow in creating a sample BOT in ODA is indicated below with illustration:

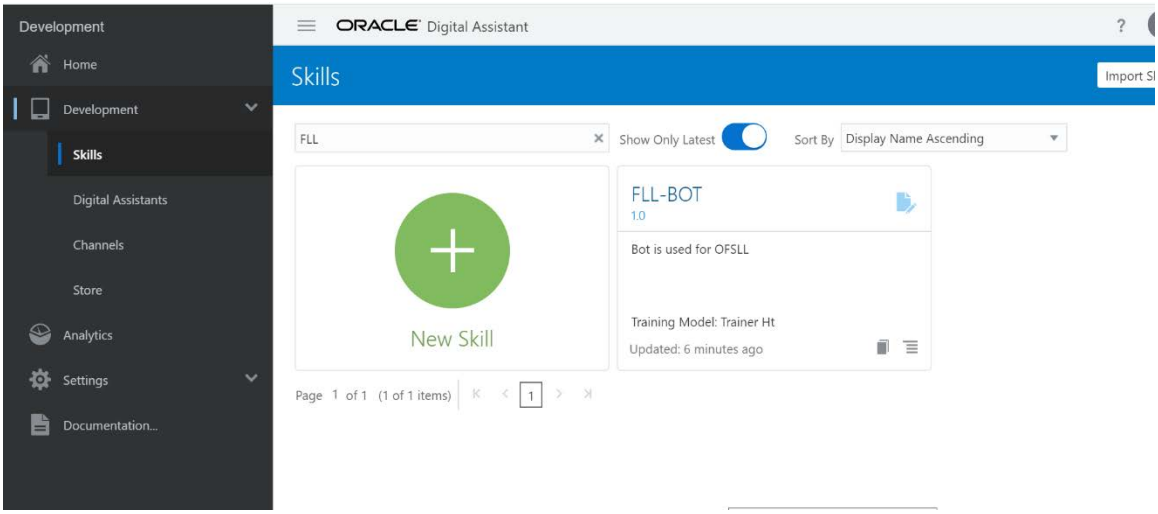
1. Login to ODA UI



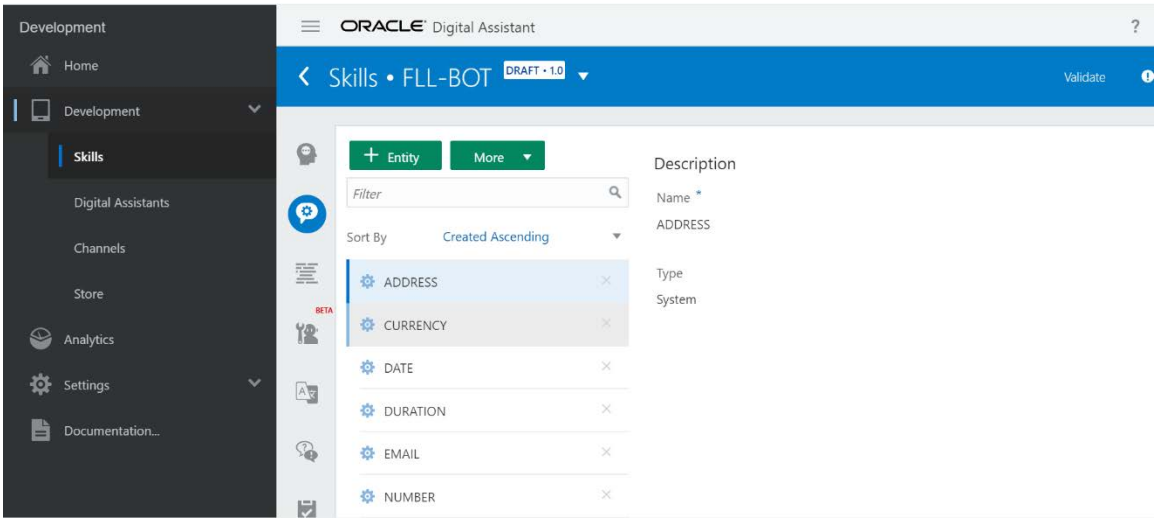
2. Go to Home



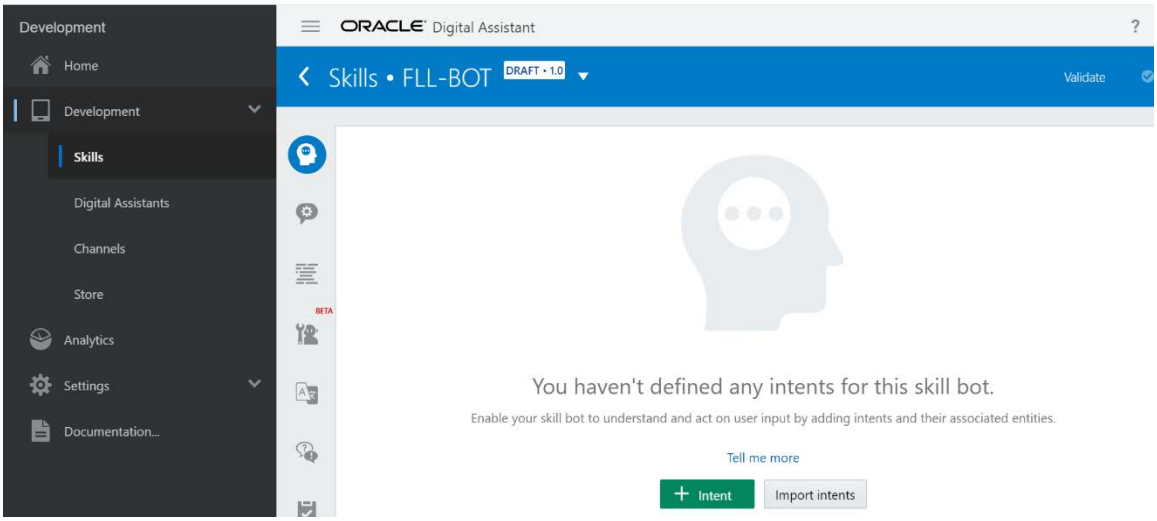
3. Create Skill/Digital Assistant.



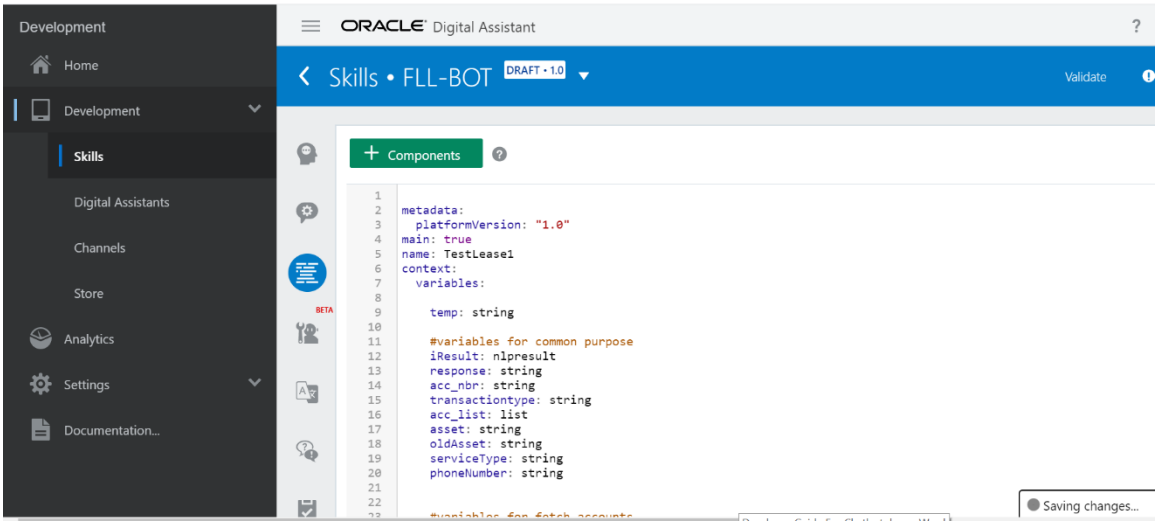
4. Add Entities



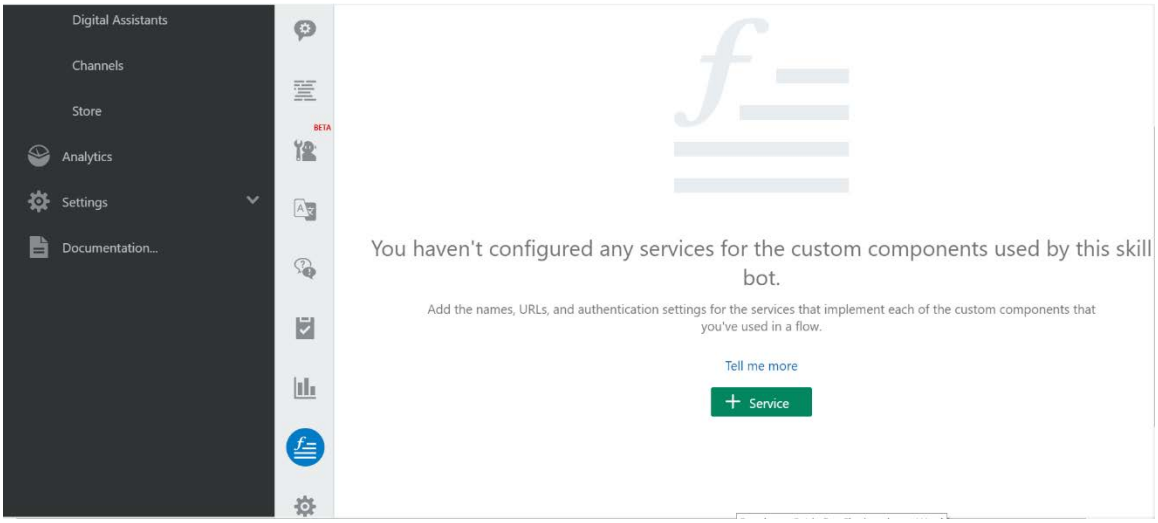
5. Add Intents. This involves defining Activity, Available option, Next level, Breakpoint, intermediate steps.

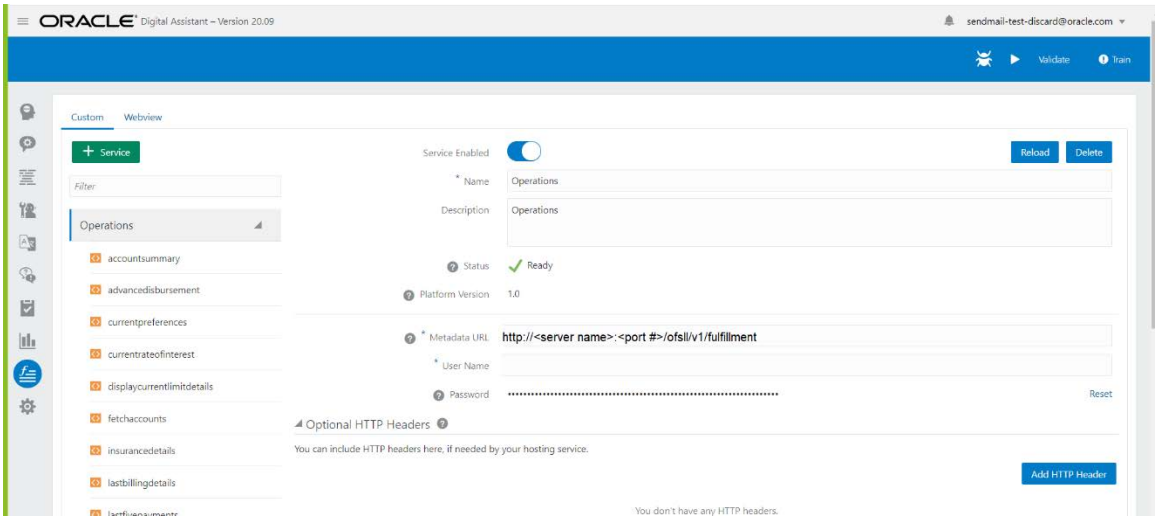


6. Add Bot flow using Yaml

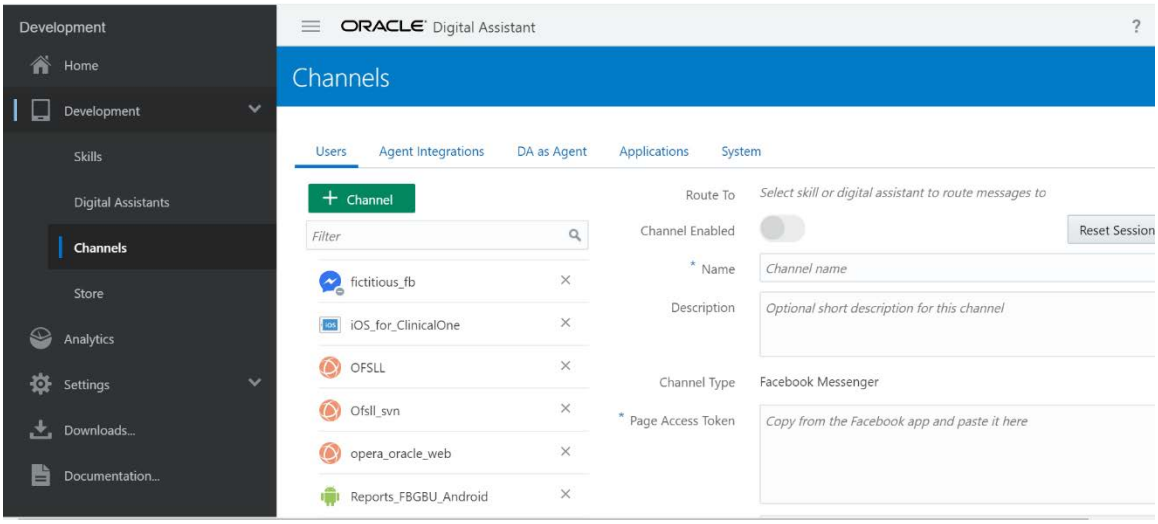


7. Add OFSLL REST Service





8. Add Channel. This indicates where it has to be published and in this sample application, only web channel is supported.
9. Enter the published URL as generated in step 2.6



10. After completion of Skill, publish. On publishing, the draft is converted to final non-editable version and only final published version is accessible in bot.
11. Additional security layer is available to allow chatbot to work for specific registered domains. To do so, select the channel, navigate to 'Allowed Domains' and add the domain name in the field. For example, adding *in.oracle.com* allows chatbot to work only from Oracle domain.
12. There is also an option to define the session time-out for chatbot which by default is set to maximum of 1440 minutes. You can enter the required time in minutes.

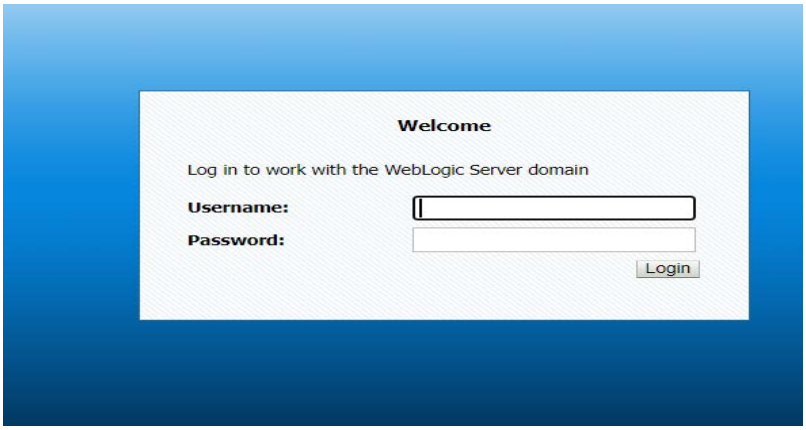
Note: The 'ofsll-documentation-bot' is the sample ODA FLL application designed for the demo purpose. The same can be imported in any ODA environment tested, modified for new features.

2.4 Deploying war file on WebLogic Server

Before you begin, ensure to use the war file for deployment of OFSLL BOT available in the path – release\<14_x.0.0.0>\ws_as\ChatBot\OracleFSLLChatBot.war.

1. Login to Web Logic application server enterprise manager (e.g.:http://hostname:port/em). For example, http://host01.example.com:8001/console

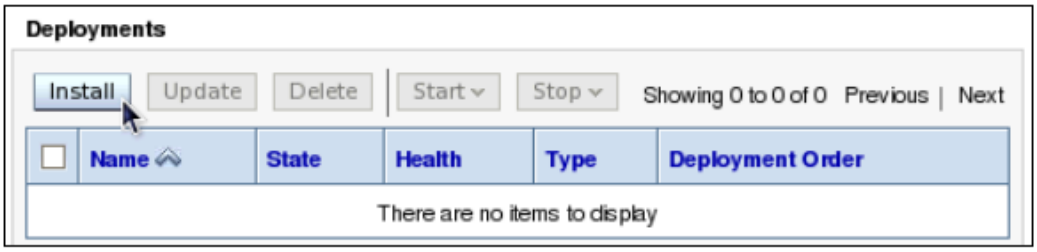
Note: Use the host name and port of the administration server of your domain.



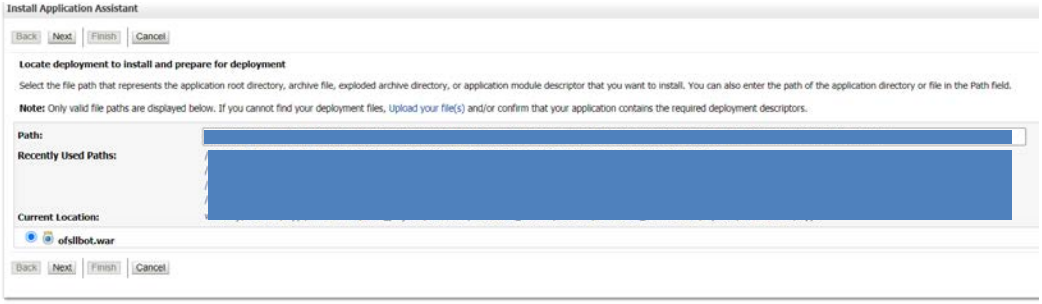
2. Enter valid login credentials.
3. Deploying an application is a change to the domain's configuration, so it must first be locked. In the Change Center. Click ‘Lock & Edit’.

4. Under Domain Structure, click ‘Deployments’.

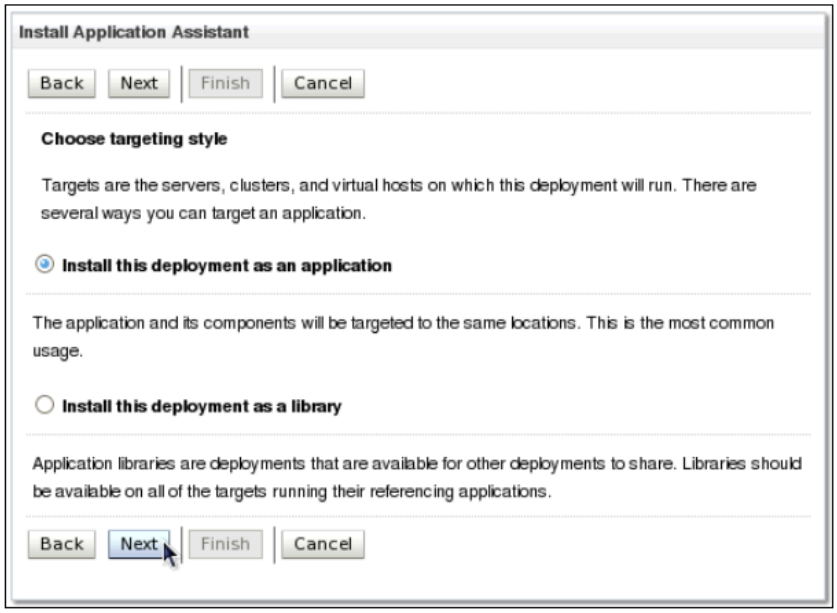
5. On the right, under Deployments, click ‘Install’.



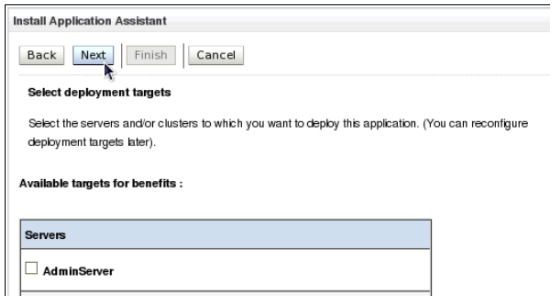
6. Find the Current Location field. Use the links to browse to the location in which you placed the downloaded OracleFSLLChatBot.war file.
7. The .war file is available in the path - release\<14_x.0.0.0>\ws_as\ChatBot\OracleFSLLChatBot.war. Select the .war file from the given path and click the radio button next to it. Using the links and the radio button, the console auto populates the Path fields. Alternatively, you can type in the path and file name in the Path field yourself. Click ‘Next’.



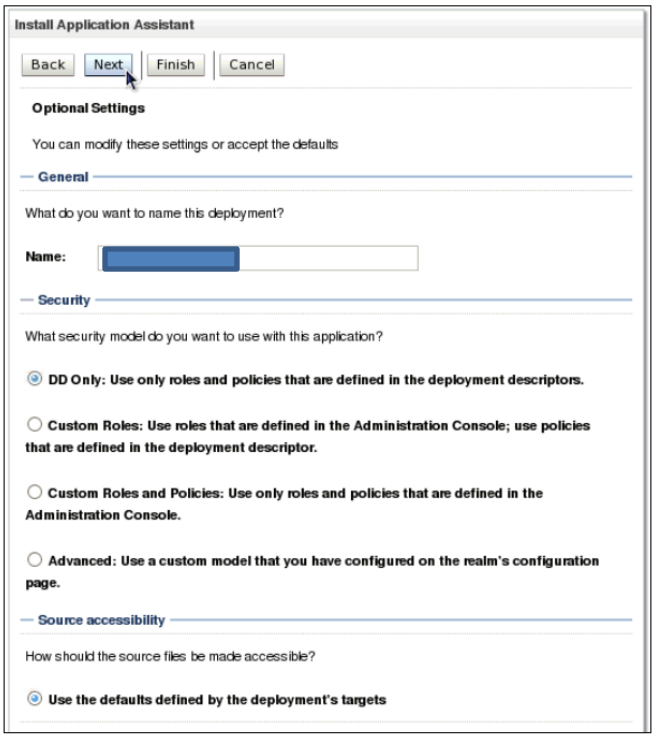
8. Ensure that ‘Install this deployment as an application’ option is selected. Click ‘Next’.



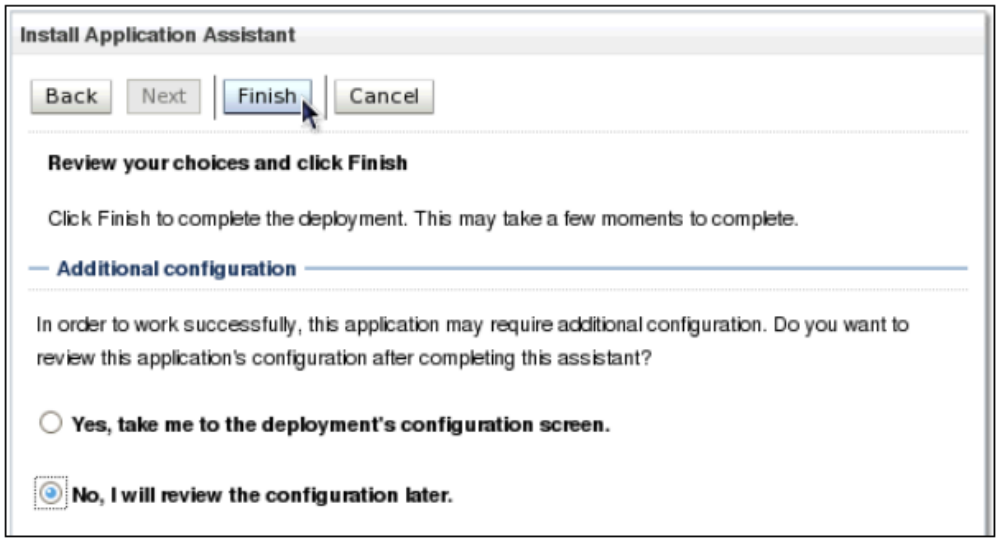
9. In the below window, click ‘Next’.



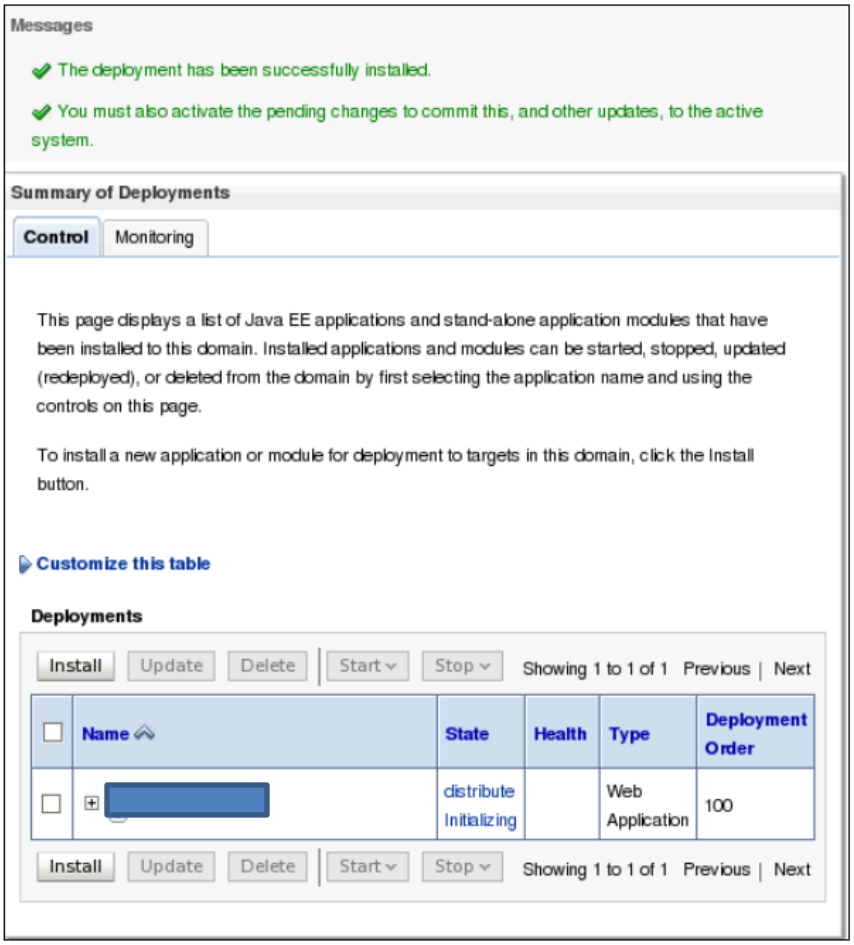
10. Retain the default values and click ‘Next’.



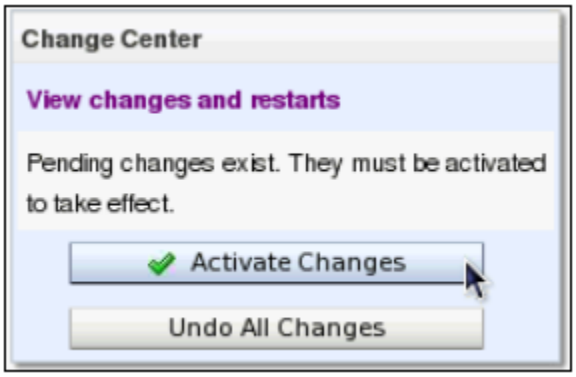
11. In the below window, select the option 'No, I will review the configuration later' and click 'Finish'.



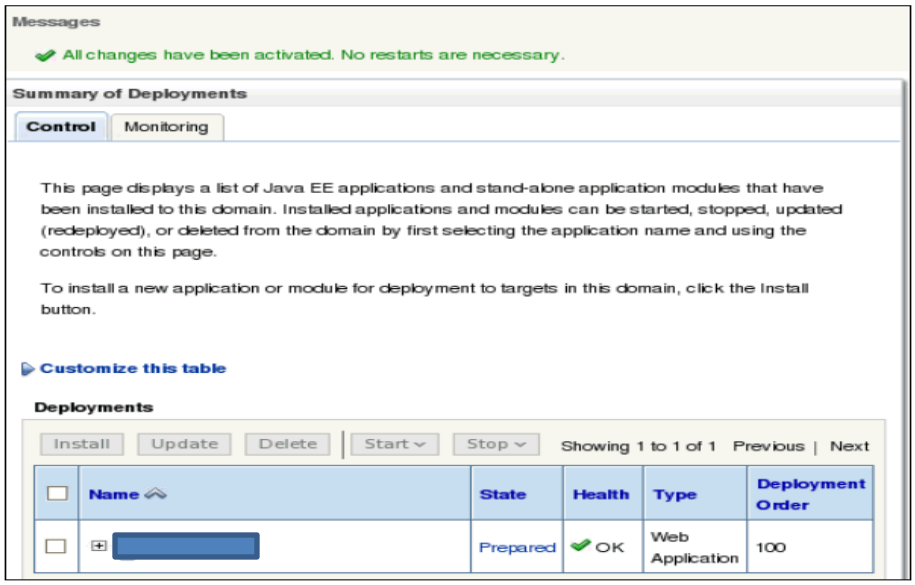
Once done view the messages indicating that the deployment was installed, but changes must be activated. In addition, notice the benefits application listed in the Deployments table.



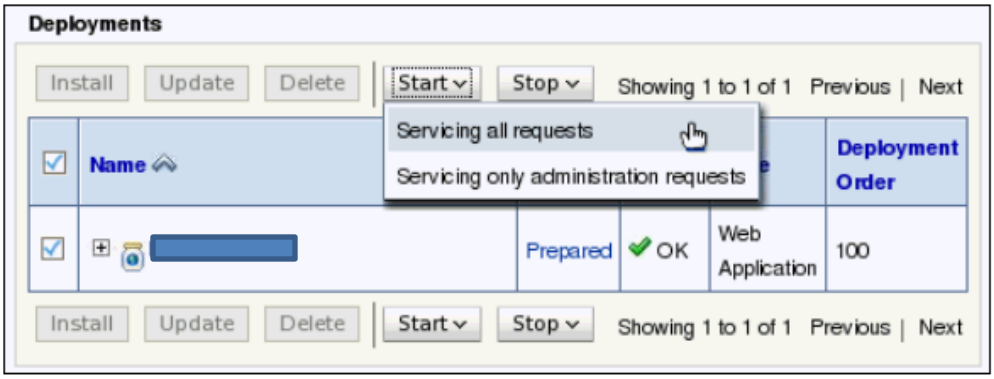
12. In the Change Center, click the Activate Changes button.



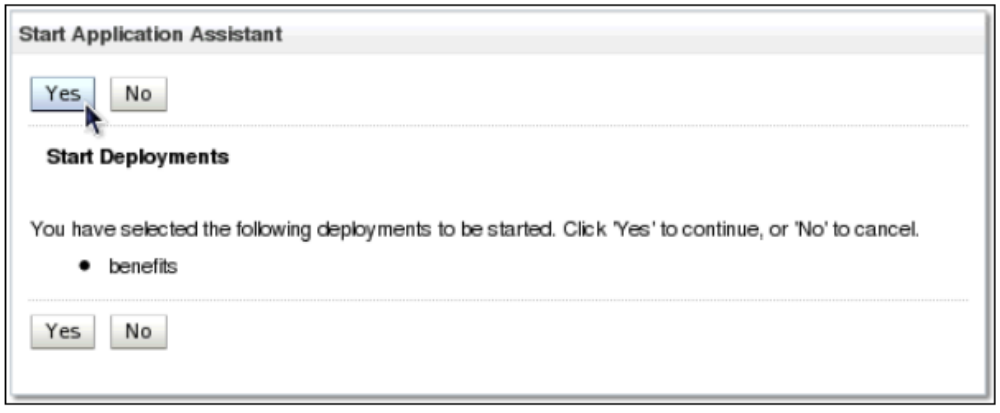
Notice the message indicating that the changes have been activated. In addition, notice the benefits application listed in the Deployments table is now in the "Prepared" state.



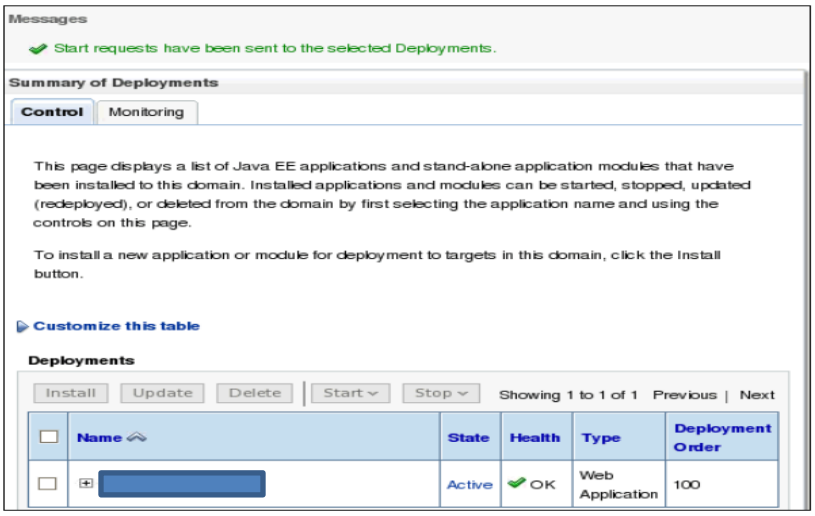
13. Select the checkbox against the left of the benefits application in the Deployments table. In the Start drop-down list, select ‘Servicing all requests’ option.



14. Click ‘Yes’ to continue.



15. A message is displayed indicating a start request was sent. Subsequently Notice that the state of application is ‘Active’ indicating that the application is accessible.



2.5 Web application UI for Accessing BOT

Web Application is User Interface where you can access the BOT functionality. The same can be integrated with OFSLL UI or any other front-end application such as customer support portal or financial institution website.

To configure WebApp, do one of the following:

- In case you wish to launch BOT as separate application, Modify **index.html** in OracleFSLLChatBot.war (or OracleFSLLChatBot.war) and update the following 2 fields with required details:
  - URI: '<ODA host>',
  - channelId: 'published bot channel ID'
- In case you wish to integrate BOT in an existing front-end application, use the provided **index.html** with the modified value and **web-sdk.js**

The BOT needs to be published on the login page and the only way it come be done is by adding the above properties in the Weblogic

For additional information, contact Oracle Financial Services Lending and Leasing Product Engineering team.

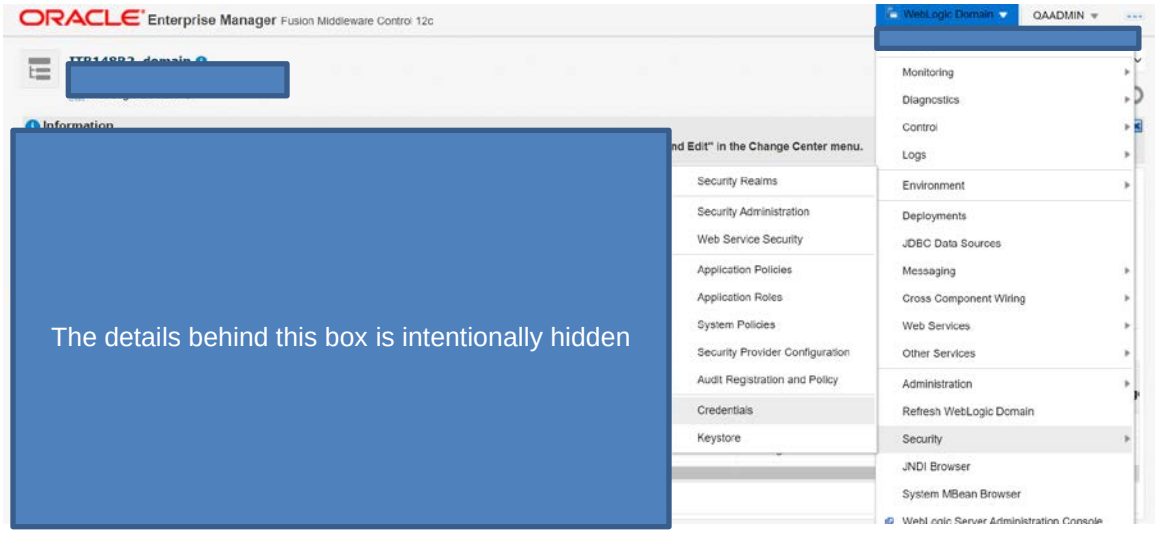
2.6 Configure CSF Mapping in Weblogic

The below section details the process of CSF configuration for BOT to appear on OFSLL home page.

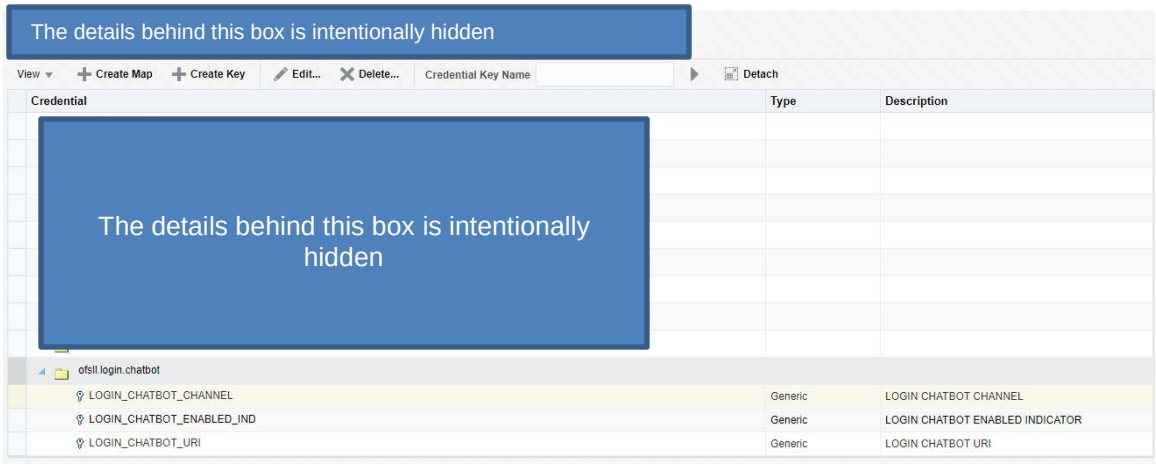
Following are the Parameters:

- Create a map called LOGIN_CHATBOT_MAP_NAME = ofsll.login.chatbot
- Channel ID
- URI
- BOT enabled – Y/N

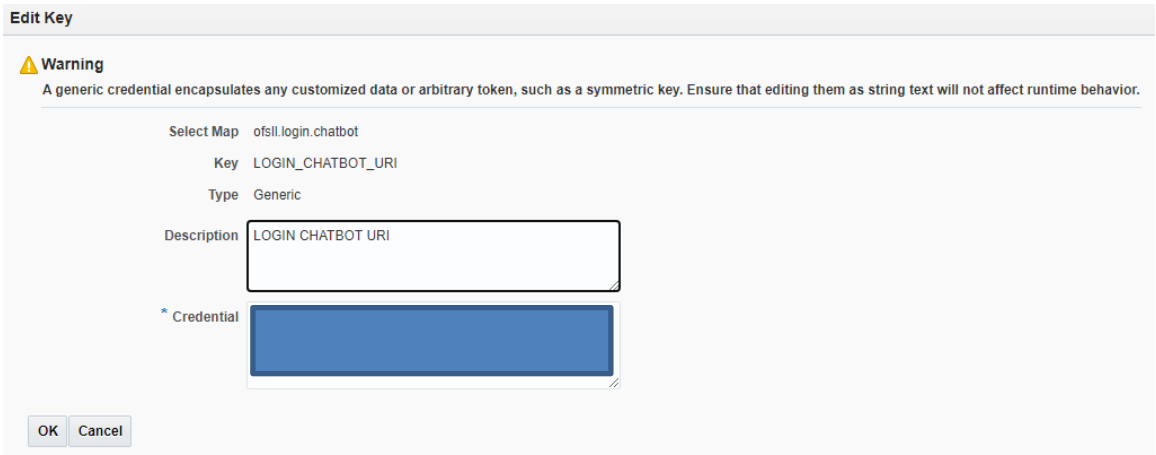
1. Login to the Weblogic server.



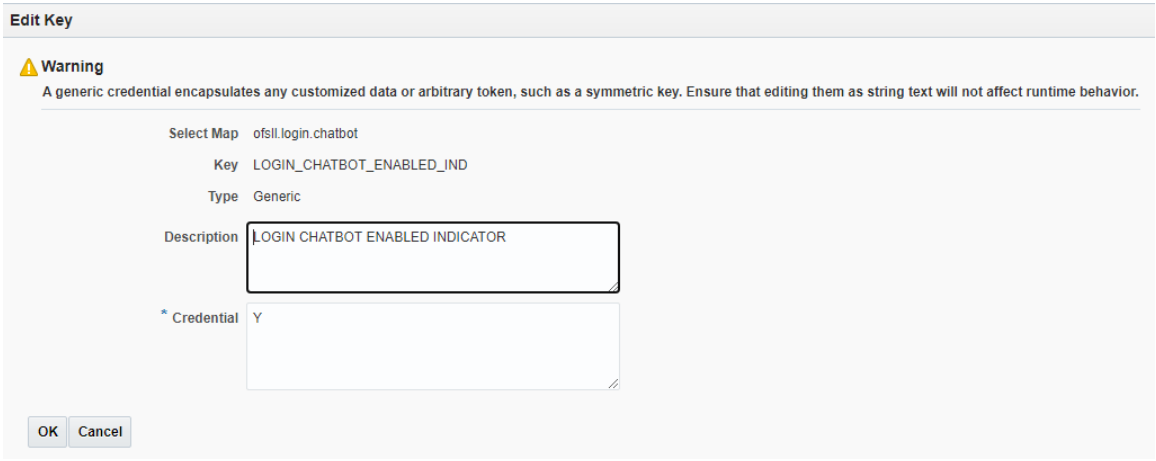
2. In the following screen, click ‘NEW MAP;



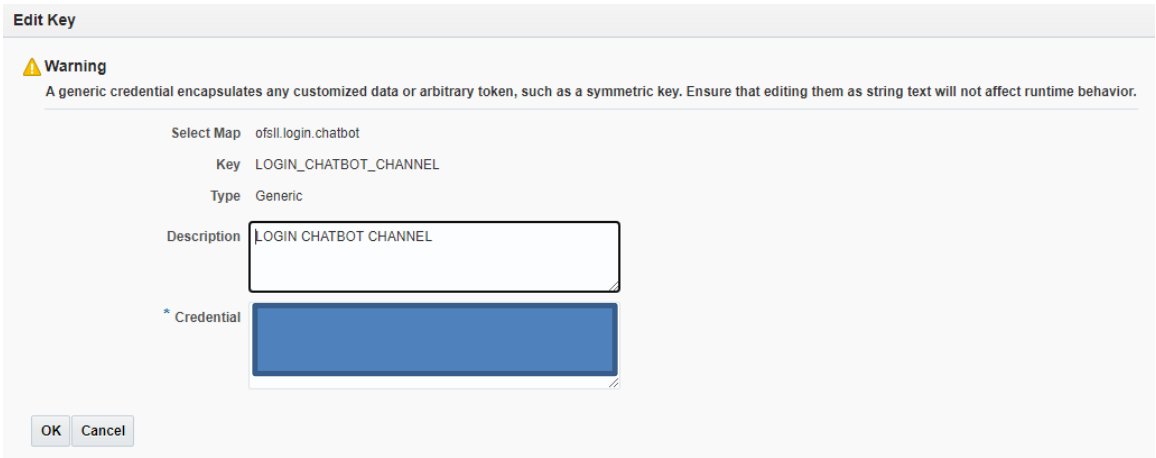
3. Add URI



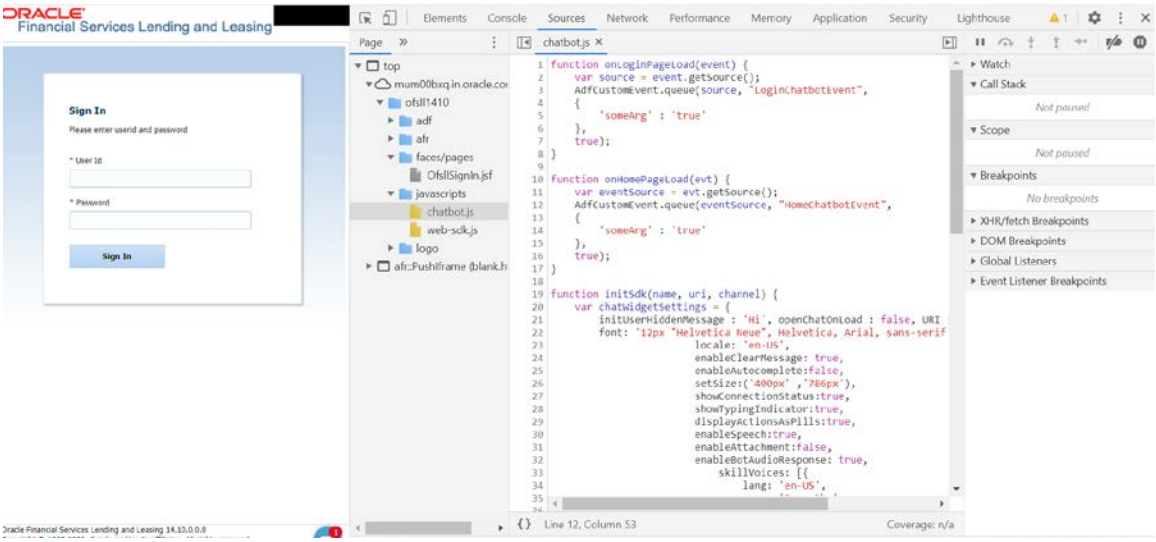
4. Enter BOT INDICATOR.



5. Enter BOT CHANNEL ID



6. Configure the chatbot.js on the login page. Refer to the below image and .js code for reference.



Ensure that no changes are done to the following js code:

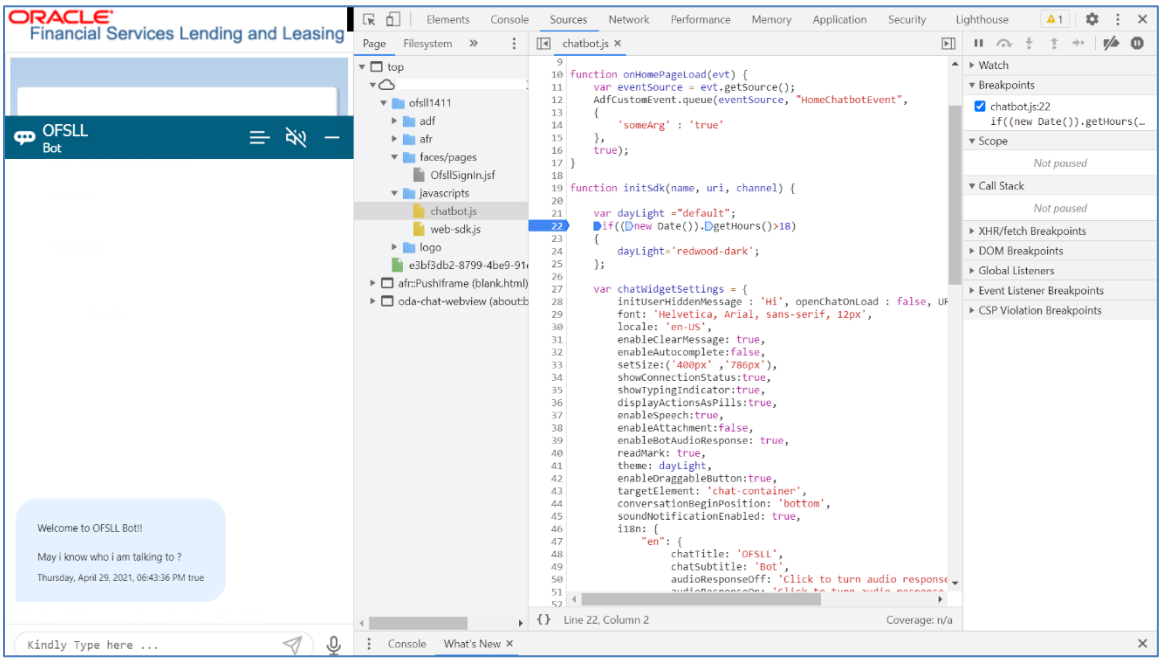
```
function onLoginPageLoad(event) {
   var source = event.getSource();
   AdfCustomEvent.queue(source, "LoginChatbotEvent",
   {
      'someArg' : 'true'
   },
   true);
}

function onHomePageLoad(evt) {
   var eventSource = evt.getSource();
   AdfCustomEvent.queue(eventSource, "HomeChatbotEvent",
   {
      'someArg' : 'true'
   },
   true);
}

function initSdk(name, uri, channel) {
   var chatWidgetSettings = {
      initUserHiddenMessage : 'Hi', openChatOnLoad : false, URI : uri, channelId :
channel,
      font: '12px "Helvetica Neue", Helvetica, Arial, sans-serif',
                  locale: 'en-US',
                  enableClearMessage: true,
                  enableAutocomplete:false,
                  setSize:('400px' ,'786px'),
                  showConnectionStatus:true,
                  showTypingIndicator:true,
                  displayActionsAsPills:true,
                  enableSpeech:true,
                  enableAttachment:false,
                  enableBotAudioResponse: true,
                    skillVoices: [{
                      lang: 'en-US',
                      name: 'Samantha'
                    }, {
                      lang: 'en-US',
                      name: 'Alex'
                    }, {
                      lang: 'en-UK'
                    }]
   };
   if (!name) {
```

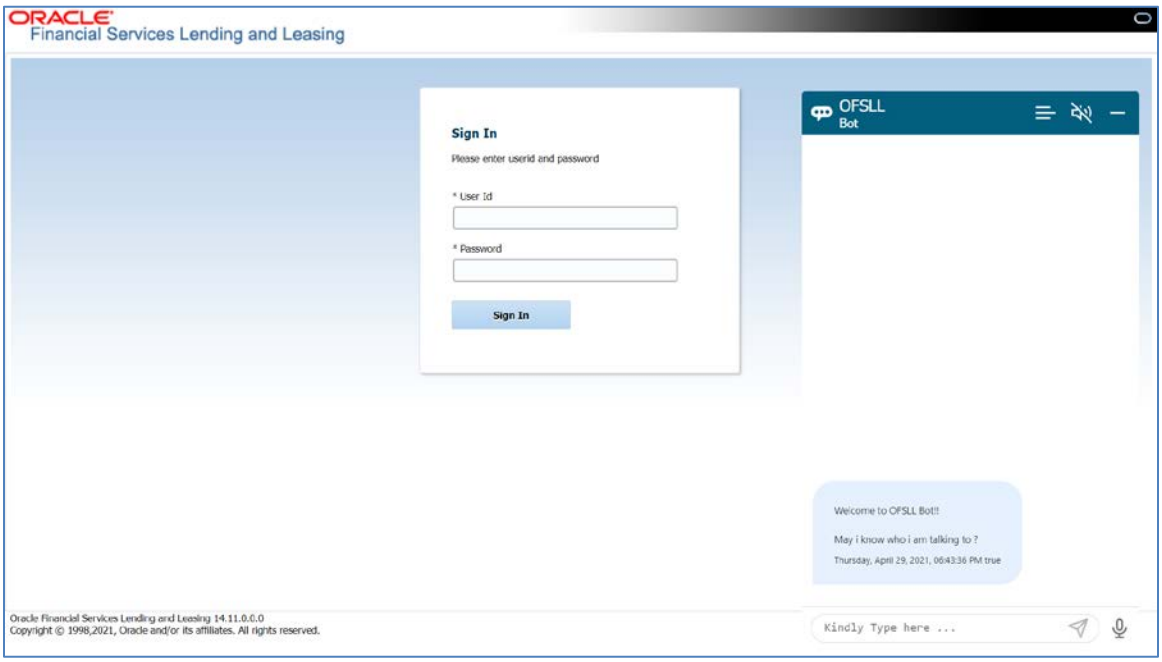
```
        name = 'Bots';
    }
    setTimeout(function () {
        const Bots = new WebSDK(chatWidgetSettings);// Initiate library with configuration
        Bots.connect()// Connect to server
.then(function () {
        })
        window[name] = Bots;
    });
}
```

The above code needs to be implemented in the chatbot.js file as shown below:



7. Web-sdk.js needs to be added from the << OFSLL Installed Directory >>/ /web_interface/ofsllbot/WebApp/scripts.

The BOT when launched form Web Application is as shown below:



2.7 BOT Configuration

For the BOT to function, the following parameters are to be defined in the application.properties file available in the .war in the path indicated below.

<OFSLL Installed Directory path>LL\release\<release version>\ws_as\ChatBot\OracleFSLLChatBot.war\WEB-INF\classes\

The below tables lists all the parameters of the properties file. However, only those fields marked as 'Y' in Update required (Y/N) column are to be updated.

Sl. No	Parameter Name	Fields	Description	Update required (Y/N)	Sample
1	paymentPurposeRequired=Y	Boolean	Captures the Payment purpose Required	N	Y
2	accessToken=	String	Captures the access token	N	
3	proxyIP=	String	Captures the Proxy	N	
4	proxyPort=	Integer	Captures the Proxy Port	N	
5	googleAPIKey=	String	Captures the Google API key	N	
6	imageUrl=	Path	Captures the Image URL	N	

Sl. No	Parameter Name	Fields	Description	Update required (Y/N)	Sample
7	defaultHomeEntity=	String	Captures the home entity	N	
8	stockCode=	String	Captures the Stock Code	N	
9	moneyTransferPay=	String	Captures the Money Transfer Pay	N	
10	defaultBaseContext=	String	Captures the default base content	N	
11	sessionExpiryInMinutes = 15	Integer	Captures the Session timeout value	N	
12	ofsll.suffix = htm	String	Suffix of the files	N	Keep as .htm
13	ofsll.otmHttpUrl=https://docs.oracle.com/cd/	String	Captures the suffix for OTM Url	N	Keep as https://docs.oracle.com/cd/
14	ofsll.fIndex=/findex.htm	String	Captures the Findex path	N	Keep as /findex.htm
15	ofsll.index=index.htm	String	Captures the index.htm	N	Keep as index.htm
16	ofsll.video=/videos.htm	String	Captures the video file path	N	Keep as /video.htm
17	ofsll.ofsllReleaseNotes=/pdf/refdocs/ofsll_release_notes.pdf	String	Captures the OFSLL release notes suffix	N	Do not change
18	ofsll.ofsllReleaseDoc=https://docs.oracle.com/en/industries/financial-services/financial-lending-leasing/index.html	String	Captures the OFSLL release doc URL	N	Do not change

Sl. No	Parameter Name	Fields	Description	Update required (Y/N)	Sample
19	ofsll.splitSeperator==	String	Captures the Split separator	N	Do not change
20	ofsll.maxHitsResults=100	String	Captures the Max no of its results of the document query	Y (optional)	Change depending upon search results
21	ofsll.baseURL =	String	Captures the Service API URL	Y	Keep this blank for documentation bot
22	ofsll.username =	String	Captures the username of weblogic server	Y	Keep this blank for documentation bot
23	ofsll.pasd =	String	Captures the Password of weblogic server	Y	Keep this blank for documentation bot
24	ofsll.indexDir =/folder path	Path	Captures the complete folder path where index files are placed (In this location, copy the index files from respective release folder. The index dir specific files are available in the below location: LL\release\14_x_0_0_0\ws_as\ChatBot\14.x)	Y	Change as per server indexed folder. **Note**: Ensure to use the same dir file indicated the path.
25	ofsll.releaseVersionUrl =	Path	Captures the Part Number	Y	Refer Release Specific Indexing table.
26	ofsll.releaseNo=	Decimal	Captures the Release Number	Y	Refer 'Folder Name' column Release Specific Indexing table.
27	ofsll.releaseHighlights =/pdf/refdocs/release_highlights.htm	String	Captures the release highlights file path	N	Keep as /pdf/refdocs/release_highlights.htm

2.8 Bot Customization

The Bot interface supports some of quick customization options as detailed below:

2.8.1.1 Label Customization

To have a customized label for BOT interface do the following:

1. Open the chatbot.js file available in below location.

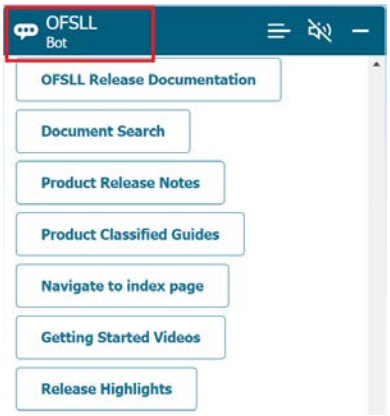
   <Installation Director >\javascripts\chatbots.js

2. Locate ‘OFSLL’ and edit the following code property as required.

```
i18n: {
    "en": {
     chatTitle: 'OFSLL', // Tool tip for title
     chatSubtitle: 'Documentation Bot',  // Tool tip for Sub Title
     audioResponseOff: 'Click to turn audio response on', // Tool tip for the speaker off
        button
     audioResponseOn: 'Click to turn audio response off',  // Tool tip for the speaker on
        button
     inputPlaceholder: 'Kindly Type here ... ', // Replaces Type a message
     send: 'Send (Enter)'          // Replaces Send tool tip
        }
}
```

3. Save the chatbots.js file and redeploy the application.

Once done, the BOT interface displays the customized label.



2.8.1.2 Bot Day/Night Theme Settings

The Bot interface can be customized to display in dark theme based on Day/Night time defined. Do the following:

1. Open the chatbot.js file available in below location.

   <Installation Director >\javascripts\chatbots.js

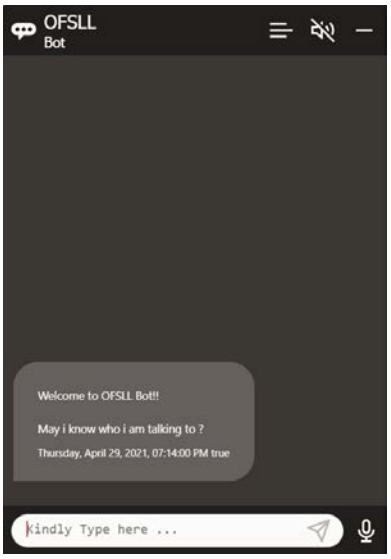
2. Locate and edit the pre-defined variable ‘daylight’ property as required.

```
var dayLight ="default";
if((new Date()).getHours()>18)
{
        dayLight='redwood-dark';
};
```

Note the following:

- Here, ‘default’ is the generic theme provided by ODA based Chabot.
- The number 18 represents 6PM, and Chabot theme changes after this time has elapsed in hours. The same can be changed as required.
- Ensure not to change the variable since the same is called in the chat Widget Settings assigned to attribute theme - Widget Settings : theme: dayLight,

Once done, the daylight theme is as displayed based on time zone:



2.8.1.3 Floating Icon

The Bot icon when minimized can be used as floating icon and moved over the screen area. However, on maximizing the Bot window is displayed in its native position.

To have the Bot icon floating, do the following:

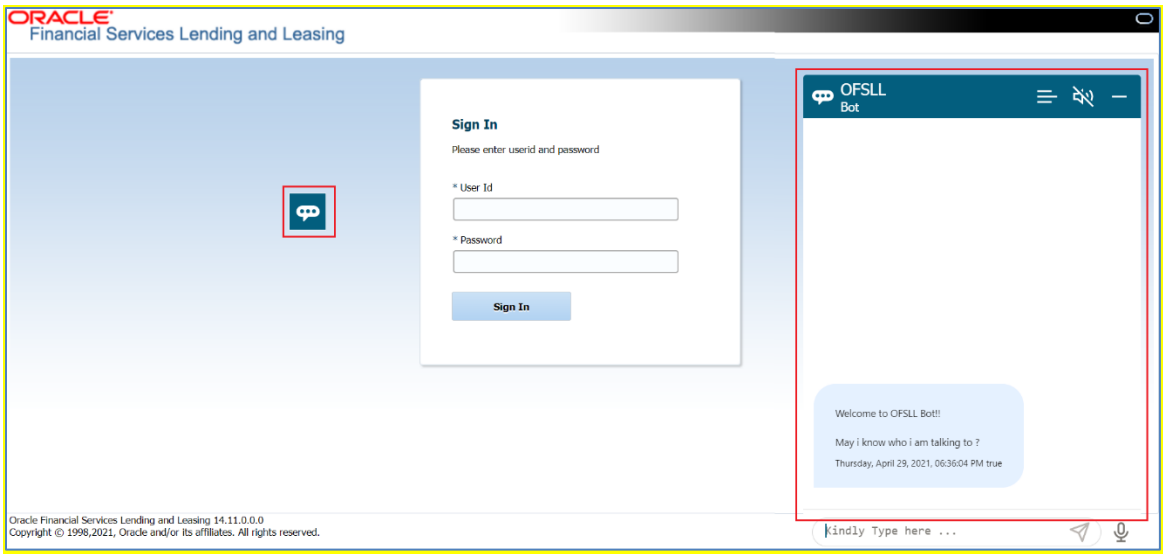
1. Open the chatbot.js file available in below location.

   <Installation Director >\javascripts\chatbots.js

2. Locate the below attribute and change the value to true.

   enableDraggableButton:true,

Note: When set to ‘true’, floating icon property is enabled and if set to ‘false’, floating icon property is disabled.


ORACLE
Financial Services Lending and Leasing
Sign In
Please enter userid and password
* User Id
* Password
Sign In
OFSLL
Bot
Welcome to OFSLL Bot!!
May i know who i am talking to ?
Thursday, April 29, 2021, 06:36:04 PM true
Kindly Type here ...
Oracle Financial Services Lending and Leasing 14.11.0.0.0
Copyright © 1998,2021, Oracle and/or its affiliates. All rights reserved.

ORACLE
Financial Services

CHATBOT Integration with OFSLL
Oracle Financial Services Lending and Leasing Release 14.12.0.0.0
March 2022

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
https://www.oracle.com/industries/financial-services/index.html