

DOCUBOT Integration with OFSLL
Overview and Developer Guide
Oracle Financial Services Lending and Leasing
Release 14.10.0.0.0
Part No. F35490-01
December 2020

ORACLE[®]
Financial Services

Table of Contents

1.	OFSLL DOCUBOT OVERVIEW AND DEVELOPER GUIDE	1-1
1.1	INTRODUCTION	1-1
1.1.1	<i>DocuBot Overview</i>	1-1
1.1.2	<i>About</i>	1-1
1.1.3	<i>Purpose</i>	1-2
1.1.4	<i>Audience</i>	1-2
1.1.5	<i>Accessibility</i>	1-2
1.1.6	<i>Access</i>	1-2
1.1.7	<i>Pre-requisites</i>	1-2
1.2	INDEXING WORKFLOW	1-2
1.2.1	<i>How is Indexing Done?</i>	1-3
1.2.2	<i>Release Specific Indexing</i>	1-3
1.3	HOW DOES SEARCH WORK?	1-3
1.4	FEATURES OF BOT	1-4
1.4.1	<i>Support of Text and Voice Based inputs</i>	1-4
1.5	BOT CONFIGURATION	1-4
1.5.1	<i>BOT UI Elements</i>	1-7
1.5.2	<i>Launch OFSLL BOT</i>	1-8
1.5.3	<i>BOT Usability Workflow</i>	1-8
1.6	THIRD PARTY LICENSES	1-13
2.	DEVELOPER GUIDE FOR BOT CUSTOMIZATION	2-1
2.1	PRE-REQUISITES	2-1
2.2	OFSLL WRAPPER CUSTOMIZATION	2-1
2.3	ODA – DIALOG FLOW DEVELOPMENT	2-5
2.4	DEPLOYING WAR FILE ON WEBLOGIC SERVER	2-9
2.5	WEB APPLICATION UI FOR ACCESSING BOT	2-15
2.6	CONFIGURE CSF MAPPING IN WEBLOGIC	2-15

1. OFSLL DOCUBOT Overview and Developer Guide

1.1 Introduction

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.

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.

For latest version of this document, refer to https://docs.oracle.com/cd/F35490_01/pdf/refdocs/ofsl docubot overview and developer guide .pdf

Following topics are discussed in this section:

- [Indexing Workflow](#)
- [How does search work?](#)
- [Features of BOT](#)
- [BOT Configuration](#)
- [Third Party Licenses](#)
- [Developer Guide for BOT Customization](#)

1.1.1 DocuBot Overview

OFSLL integrated Docubot (Documentation searchable chatbot) 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 14.10.0.0.0 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. 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.
- The parameters in 'Channel.Properties' file are to be configured before creating and deploying the .war file. 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 14.10.0.0.0 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.

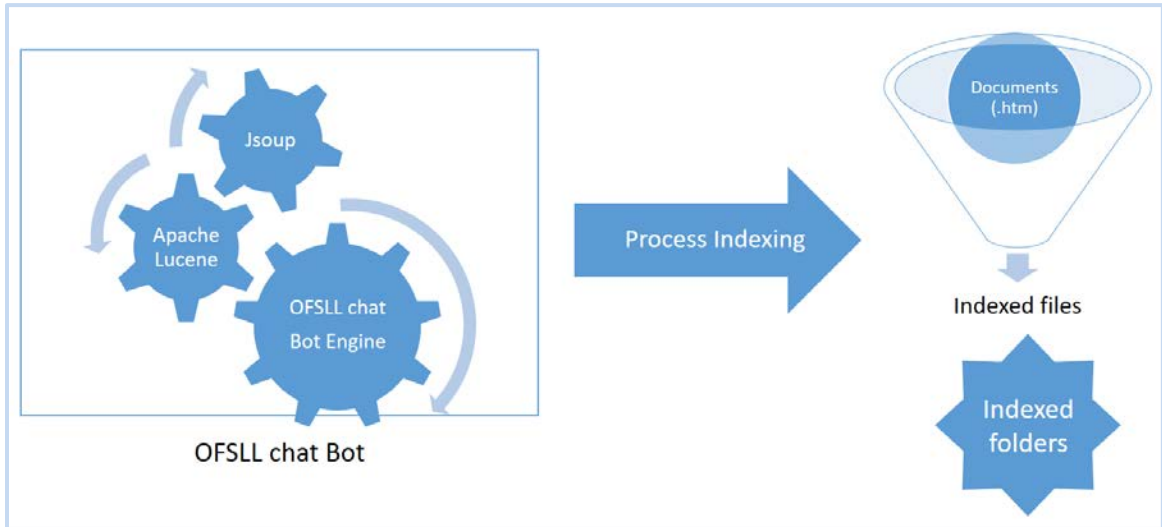
For more information on third-party plugins used, refer to '[Third Party Licenses](#)' section.

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

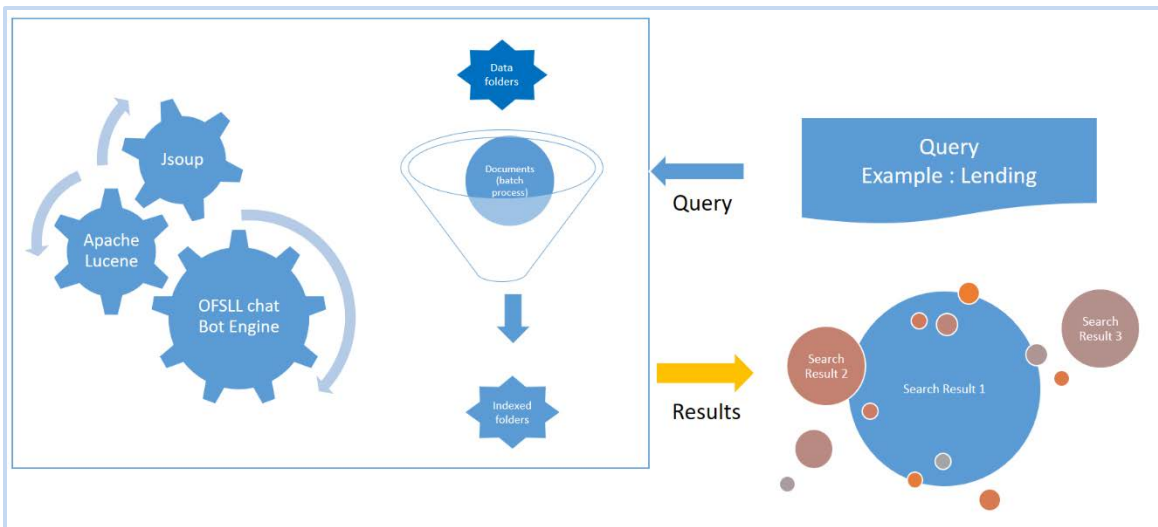
Indexing is done for the following release of OFSLL and indexed files are provided in respective folder. The mapping of Release number v/s Folder name and Part Number is indicated below:

Release No	Folder Name	Part Number
14.10.0.0.0	14.10	F53490_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 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
- Intuitive Menu options:
 - Option to clear chat data
 - Speech Conversion – Voice based Input
 - Network Status indication

1.4.1 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.

1.5 BOT Configuration

For the BOT to function, the following parameters are to be defined in the Properties file as indicated below. This file can be accessed from the location <OFSLL Installed Directory path>/web_interface/ofslbot/src.

This section consists of the following:

- [BOT UI Elements](#)
- [Launch OFSLL BOT](#)
- [BOT Usability Workflow](#)

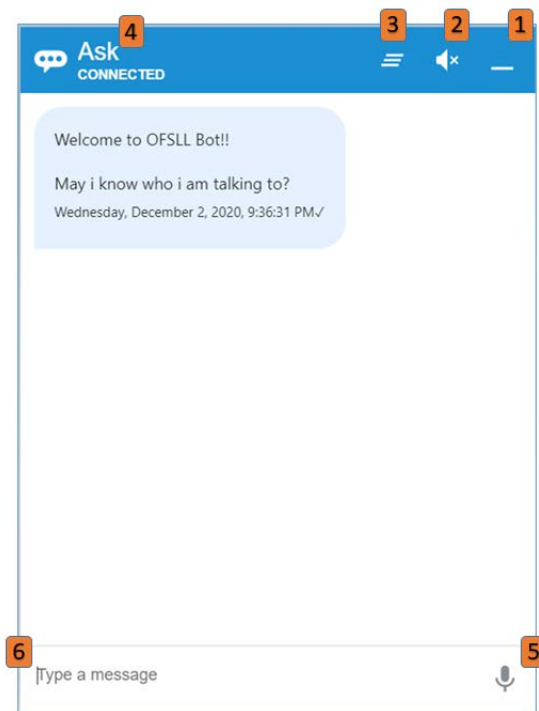
The below tables lists all the parameters of the properties file. However, only those fields marked as 'Y' in Update required (Y/N) column are to be updated.

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	
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	sessionExpiresInMinutes = 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

SI. No	Parameter Name	Fields	Description	Update required (Y/N)	Sample
15	ofssl.index=index.htm	String	Captures the index.htm	N	Keep as index.htm
16	ofssl.video=/videos.htm	String	Captures the video file path	N	Keep as /video.htm
17	ofssl.ofsslReleaseNotes=/pdf/refdocs/ofssl_release_notes.pdf	String	Captures the OFSLL release notes suffix	N	Do not change
18	ofssl.ofsslReleaseDoc=https://docs.oracle.com/en/industries/financial-services/financial-lending-leasing/index.html	String	Captures the OFSLL release doc URL	N	Do not change
19	ofssl.splitSeparator==	String	Captures the Split separator	N	Do not change
20	ofssl.maxHitsResults=100	String	Captures the Max no of its results of the document query	Y (optional)	Change depending upon search results
21	ofssl.baseURL =	String	Captures the Service API URL	Y	Keep this blank for documentation bot
22	ofssl.username =	String	Captures the username of weblogic server	Y	Keep this blank for documentation bot
23	ofssl.paswd =	String	Captures the Password of weblogic server	Y	Keep this blank for documentation bot
24	ofssl.indexDir =/folder path	Path	Captures the complete folder path where index files are placed	Y	Change as per server indexed folder

SI. No	Parameter Name	Fields	Description	Update required (Y/N)	Sample
			(In this location, copy the index files from respective release folder)		
25	ofslI.releaseVersionUrl=	Path	Captures the Part Number	Y	Refer Release Specific Indexing table.
26	ofslI.releaseNo=	Decimal	Captures the Release Number		Refer 'Folder Name' column Release Specific Indexing table.

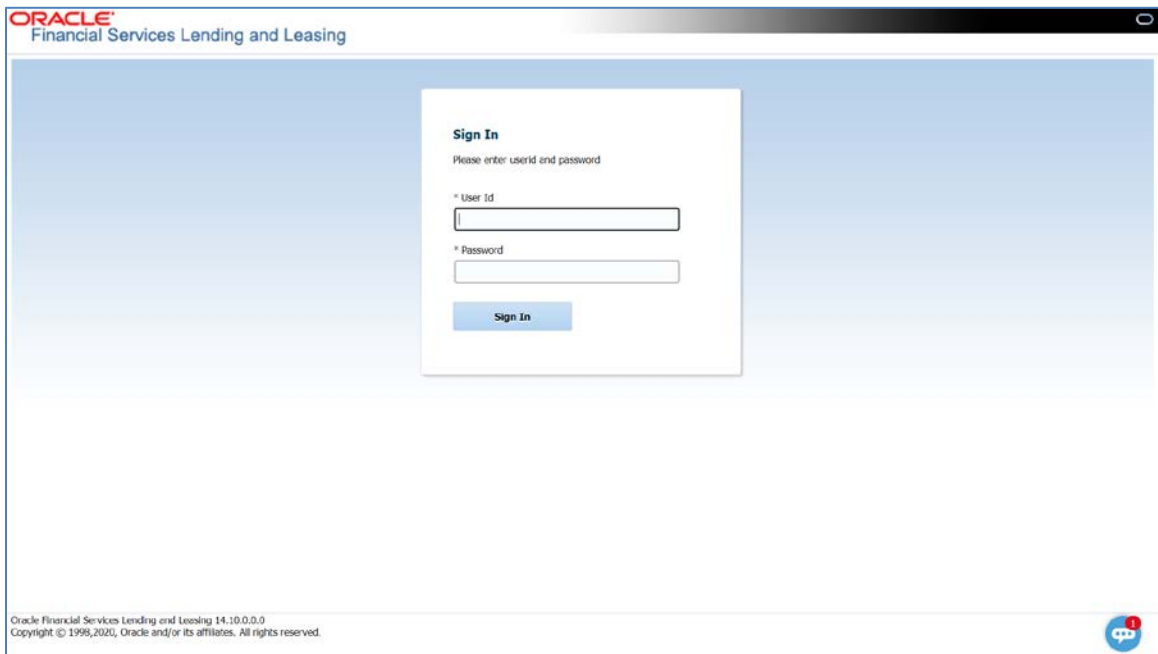
1.5.1 BOT UI Elements



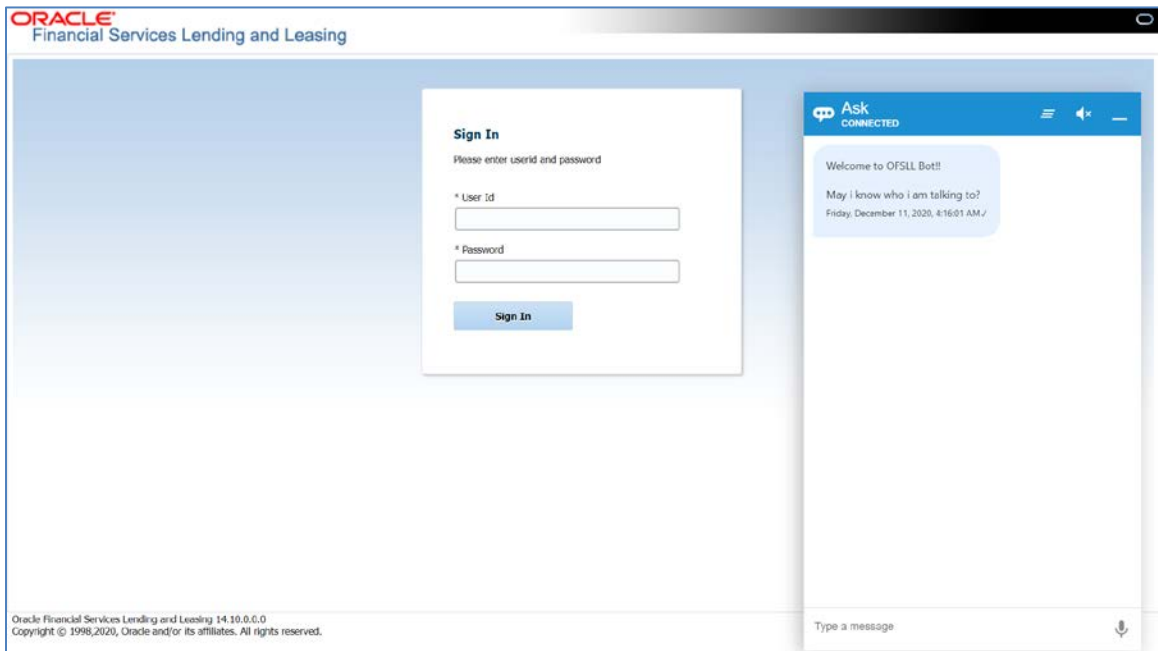
SI.No	Option	View / Action
1	Minimize	Minimize BOT window
2	Speaker output	Enable BOT in speaker mode
3	Clear chat	Clear all messages in the BOT
4	Network Status	Connected: BOT connected to server Connecting: BOT awaiting network connection
5	Mic Input	Enable Mic for voice based input
6	Text Input	Enter search string using keyboard

1.5.2 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.

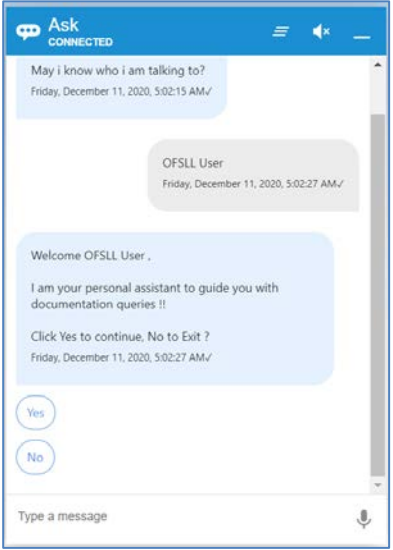
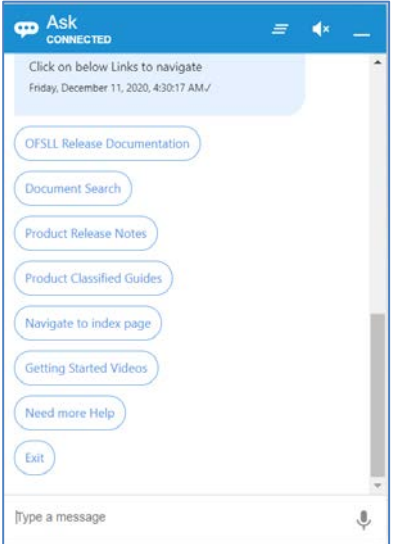


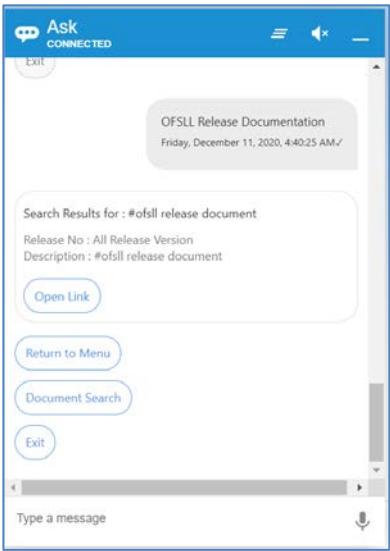

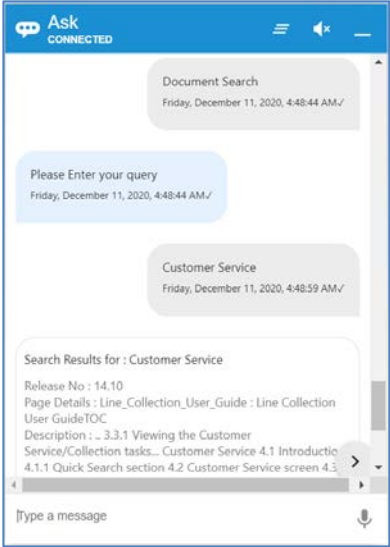
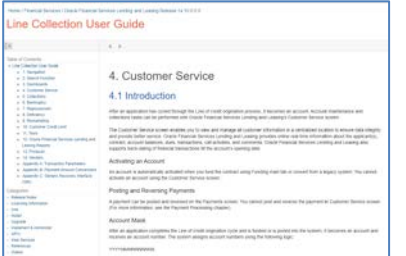
On clicking the BOT icon, the welcome message is as displayed:

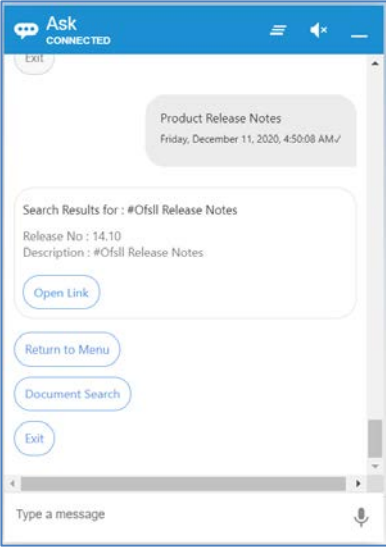
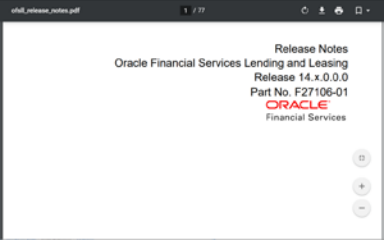
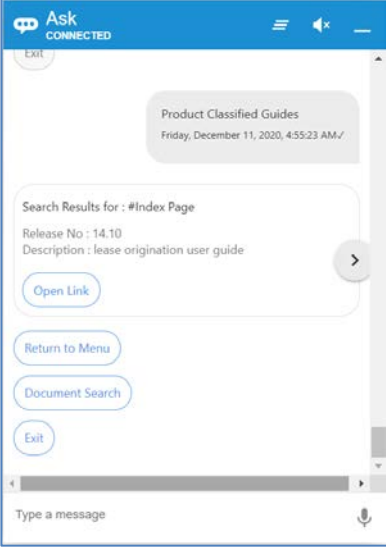
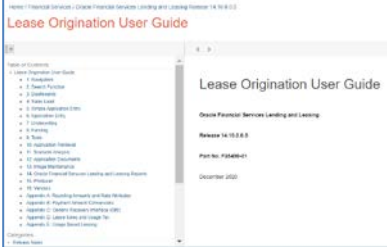


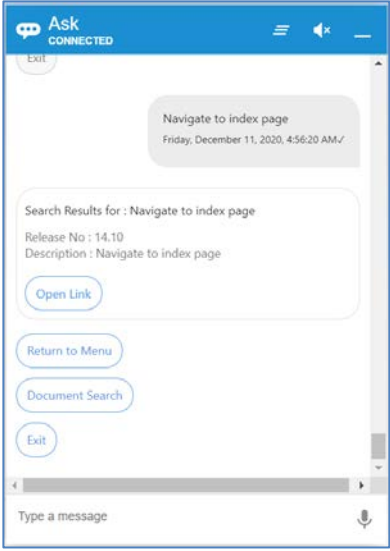

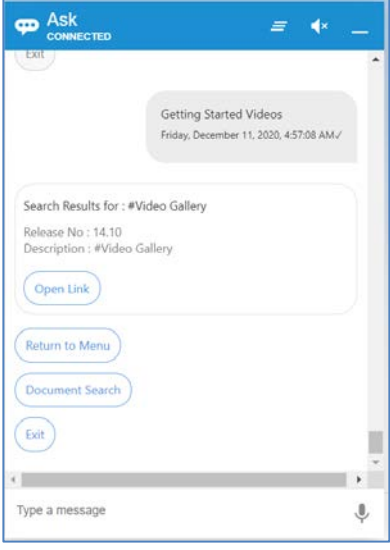
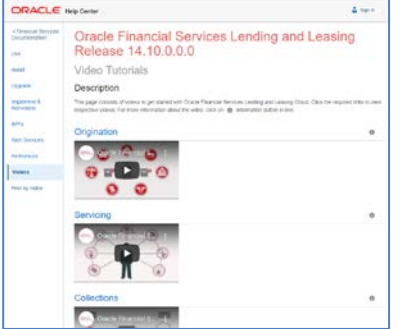
1.5.3 BOT Usability Workflow

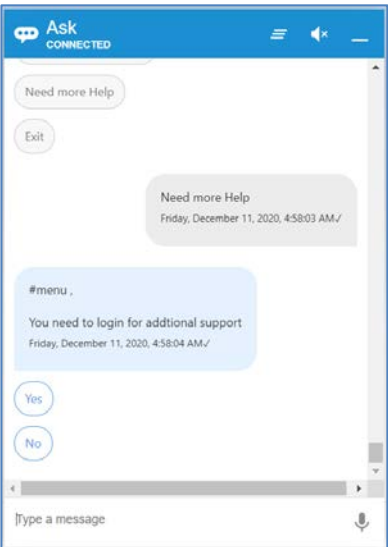
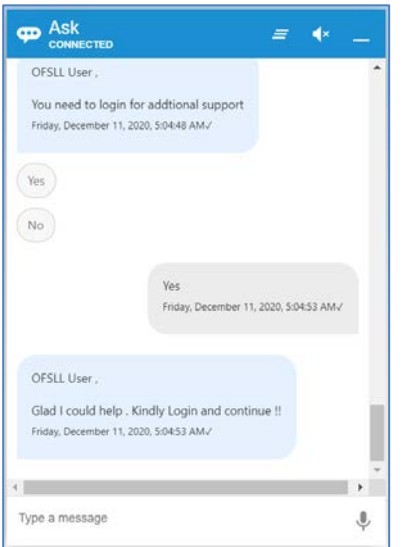
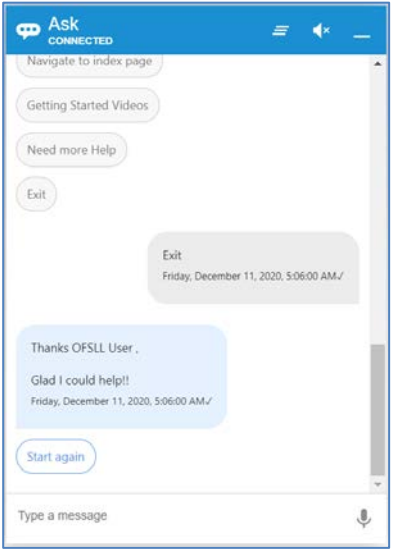
Below is a simple sequence of user interaction with OFSLL BOT.

SI.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 resources for selection.	
	Clicking on 'No'	Docubot displays an exit message. However, there is option to start again.	

<p>3</p>	<p>Clicking on OFSLL Release Documentation</p>	 <p>Clicking on the link, BOT displays the home screen with access to all release documentation libraries.</p>	 <table border="1"> <thead> <tr> <th>Release</th> <th>Part #</th> <th>Link</th> <th>Download</th> <th>Patch Release</th> </tr> </thead> <tbody> <tr> <td>14.0.0.0</td> <td>12700-01</td> <td>View</td> <td>Download</td> <td>NA</td> </tr> <tr> <td>14.0.0.0</td> <td>12200-01</td> <td>View</td> <td>Download</td> <td>14.0.0.0 Release</td> </tr> <tr> <td>14.0.0.0</td> <td>14000-01</td> <td>View</td> <td>Download</td> <td>14.0.0.0 Release</td> </tr> <tr> <td>14.0.0.0</td> <td>14000-01</td> <td>View</td> <td>Download</td> <td>14.0.0.0 Release</td> </tr> <tr> <td>14.0.0.0</td> <td>00000-01</td> <td>View</td> <td>Download</td> <td>14.0.0.0 Release</td> </tr> </tbody> </table>	Release	Part #	Link	Download	Patch Release	14.0.0.0	12700-01	View	Download	NA	14.0.0.0	12200-01	View	Download	14.0.0.0 Release	14.0.0.0	14000-01	View	Download	14.0.0.0 Release	14.0.0.0	14000-01	View	Download	14.0.0.0 Release	14.0.0.0	00000-01	View	Download	14.0.0.0 Release
Release	Part #	Link	Download	Patch Release																													
14.0.0.0	12700-01	View	Download	NA																													
14.0.0.0	12200-01	View	Download	14.0.0.0 Release																													
14.0.0.0	14000-01	View	Download	14.0.0.0 Release																													
14.0.0.0	14000-01	View	Download	14.0.0.0 Release																													
14.0.0.0	00000-01	View	Download	14.0.0.0 Release																													
<p>4</p>	<p>Clicking on Documentation Search</p>	 <p>BOT displays dynamic results in horizontal navigating cards based on ranks from the list of documents which were indexed.</p> <p>Clicking on the link, BOT displays the result page which contains the search keyword.</p>																															

<p>5</p>	<p>Clicking on Product Release Notes</p>	 <p>Clicking on the link, BOT displays the release notes of current release.</p>	
<p>6</p>	<p>Clicking on Product Classified Guides</p>	 <p>Clicking on the link, BOT displays the module specific guide.</p>	

<p>7</p>	<p>Clicking on Navigate to index page</p>	 <p>Clicking on the link, BOT displays the Find by Index page which consists of all indexed keywords in documentation.</p>	
<p>8</p>	<p>Clicking on Getting Started Videos</p>	 <p>Clicking on the link, BOT displays the video gallery link with getting started videos.</p>	

9	Clicking on Need More Help		
10	Clicking on Exit	<p>BOT displays an exit message. However, there is option to start again.</p>	

1.6 Third Party Licenses

OFSLL BOT uses the following third party licenses:

- Apache Lucene, Version: 8.5.1
The Apache Software Foundation, Technology: Lucene, Version: 8.5.1
Files used (below are part of Apache Lucene 8.5.1)
Lucene Core (8.5.1)
Lucene query parser (8.5.1)

- JSOUP 1.13.1

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.

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:

- [OFSLL Wrapper customization](#)
- [ODA – Dialog Flow Development](#)
- [Deploying war file on WebLogic Server](#)
- [Web application UI for Accessing BOT](#)
- [Configure CSF Mapping in Weblogic](#)

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

Before starting OFSLL Wrapper customization, ensure that following files are present in jdeveloper/WebLogic installation with different versions which are required for compilation:

- eclipselink.jar
- jackson-annotations-2.9.5.jar
- jackson-core-2.9.5.jar

- jackson-databind-2.9.5.jar
- javax.ws.rs-api-2.0.1.jar
- jersey-client-2.28.jar
- jersey-common-2.29.jar
- servlet-api-2.5.jar
- jsoup-1.13.1.jar
- lucene-core-8.5.1.jar
- lucene-highlighter-8.5.1.jar
- lucene-memory-8.5.1.jar
- lucene-queries-8.5.1.jar
- lucene-queryparser-8.5.1.jar

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>

```

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(ODARRequestDTO 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("ofssl.indexDir")+prop.getProperty("ofssl.releaseNo");
            String releaseVersionUrl = prop.getProperty("ofssl.releaseVersionUrl");
            String urlPrefix = prop.getProperty("ofssl.otmHttpUrl");
            String splitSeperator = prop.getProperty("ofssl.splitSeperator");
            String releaseNo = prop.getProperty("ofssl.releaseNo");
            String urlPrefixPath = urlPrefix + releaseVersionUrl;
            String findIndexPath = prop.getProperty("ofssl.findIndex");
            String indexPath = prop.getProperty("ofssl.index");
            String videoPath = prop.getProperty("ofssl.video");
            String ofsslReleaseNotesPath = prop.getProperty("ofssl.ofsslReleaseNotes");
            String ofsslReleaseDocPath = prop.getProperty("ofssl.ofsslReleaseDoc");
            Integer maxHitsResults =
Integer.parseInt(prop.getProperty("ofssl.maxHitsResults"));
            File fileIndexDirPath = new File(indexDirPath);
            LuceneSearchHighlighter luceneSearchHighlighter = new
LuceneSearchHighlighter();
            List<String> fileList = new ArrayList <> ();

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

```

```

        if ((searchQuery.toLowerCase().trim().contains("#ofsl release document")))
    {
        releaseNo="All Release Version";
        fileList.add(searchQuery + splitSeperator + ofslReleaseDocPath +
splitSeperator+searchQuery+ splitSeperator+releaseNo);
    }
    if ((searchQuery.toLowerCase().trim().contains("navigate to index page"))) {
        fileList.add(searchQuery + splitSeperator + urlPrefixPath + indexPath +
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("#ofsl release notes"))) {
        fileList.add(searchQuery + splitSeperator + urlPrefixPath +
ofslReleaseNotesPath + 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 in step 3 onto weblogic server.
6. Note down base service URL that is required while publishing in ODA.
Example : http://<host>:<port>/ofssl/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 OFSSL.

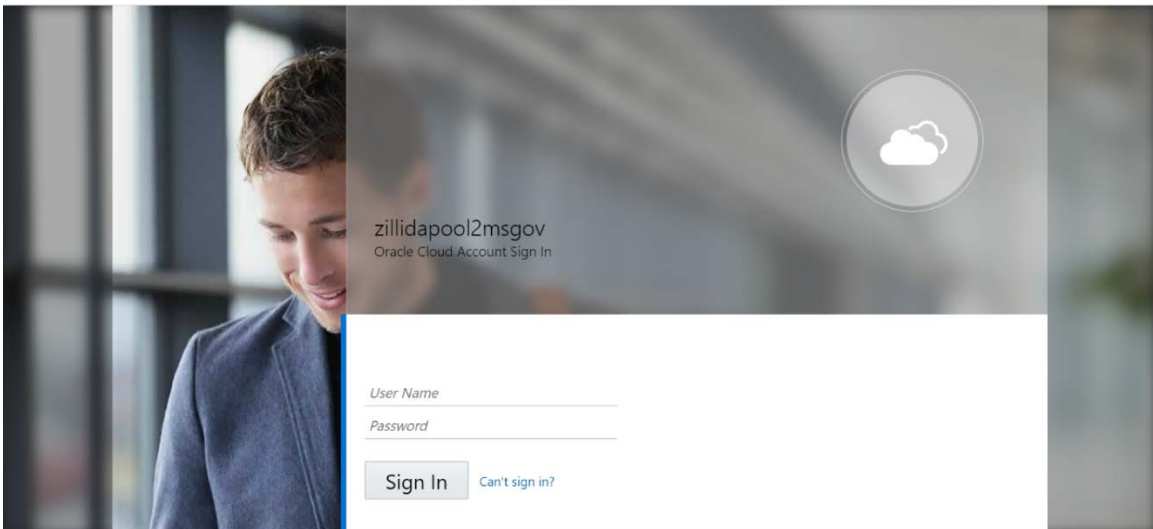
In-order to achieve a sequence of menu options, dialog flow development is required to be performed in ODA Oracle Digital Assistant. Following is a quick overview of steps involved:

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

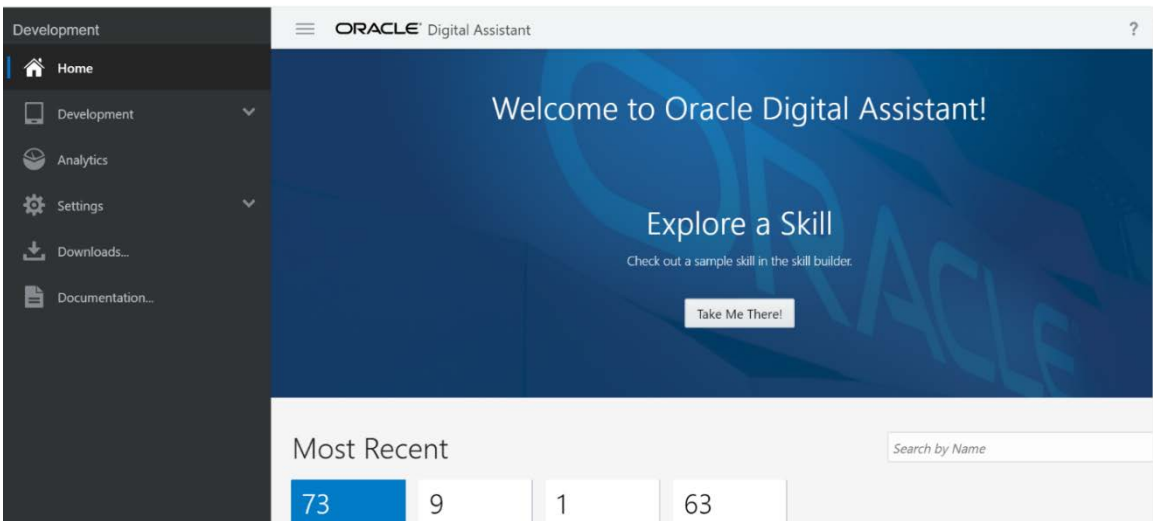
It is recommended to refer to ODA documentation for detailed information - <https://docs.oracle.com/en/cloud/paas/digital-assistant/index.html>

In the ODA - dialog flow development, you can either create new / import the given sample. The sequence of flow in creating a sample BOT in ODA is indicated below with illustration:

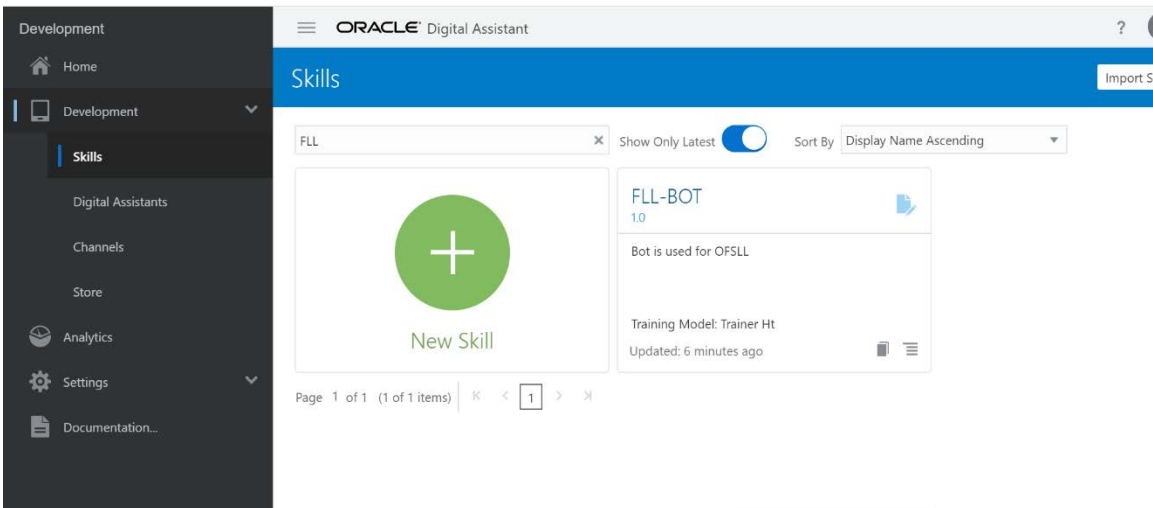
1. Login to ODA UI



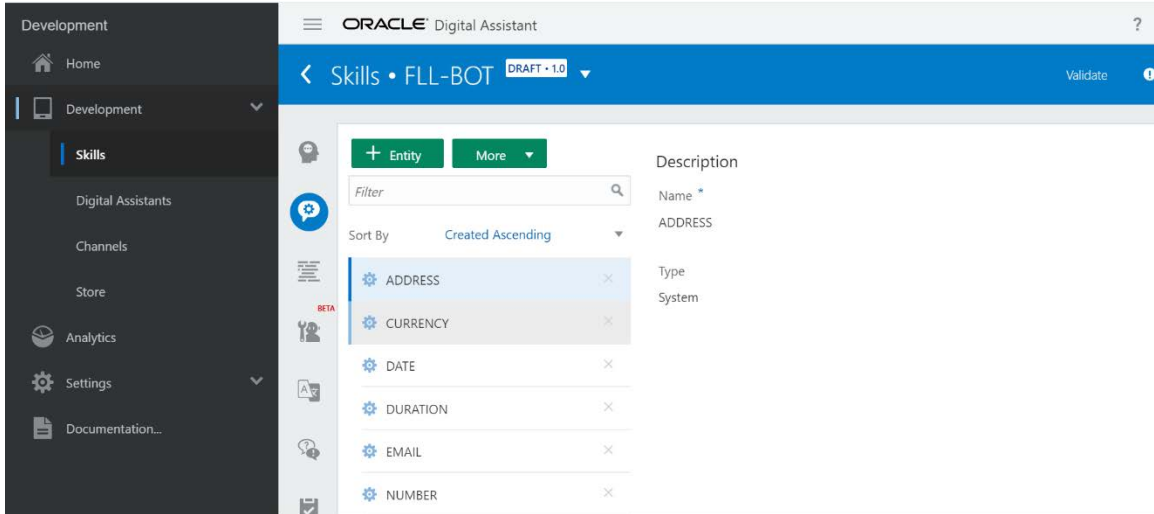
2. Go to Home



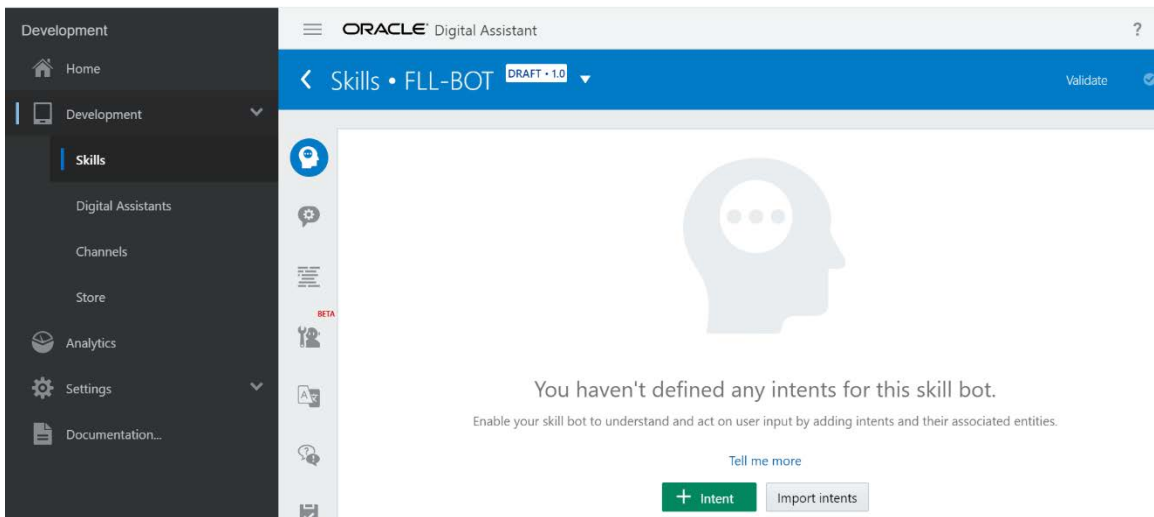
3. Create Skill/Digital Assistant.



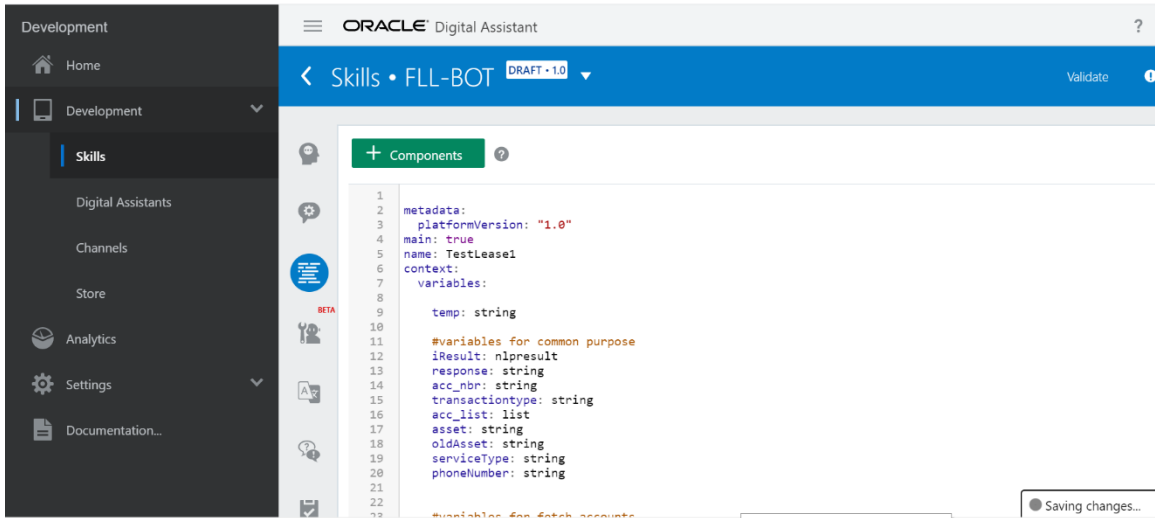
4. Add Entities



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



6. Add Bot flow using Yaml

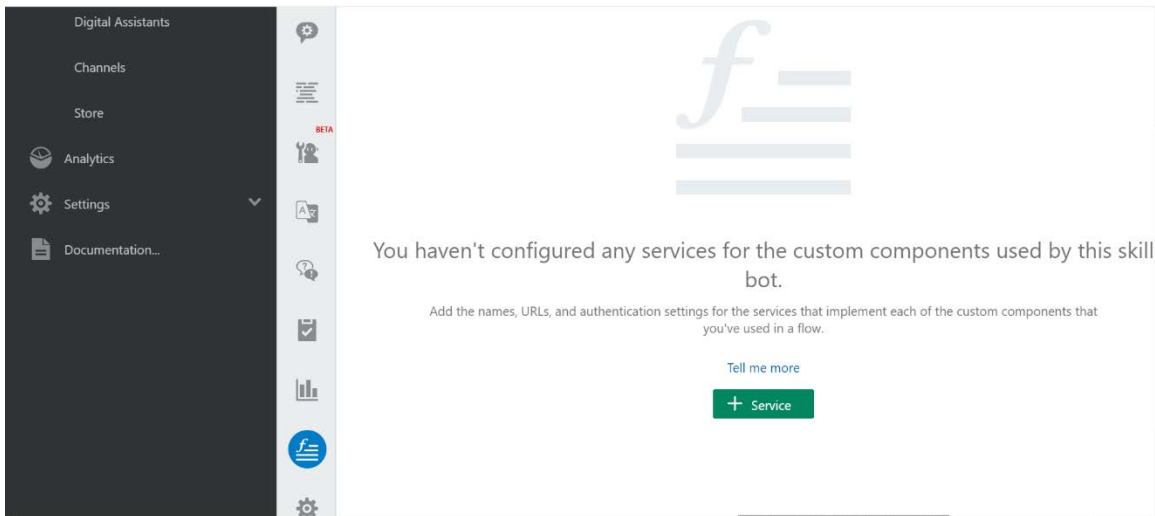


The screenshot shows the Oracle Digital Assistant interface. On the left is a navigation menu with options like Home, Development, Skills, Digital Assistants, Channels, Store, Analytics, Settings, and Documentation. The main area displays the configuration for a skill named 'FLL-BOT' in 'DRAFT' mode. A '+ Components' button is visible at the top. The configuration is shown in a code editor with the following YAML content:

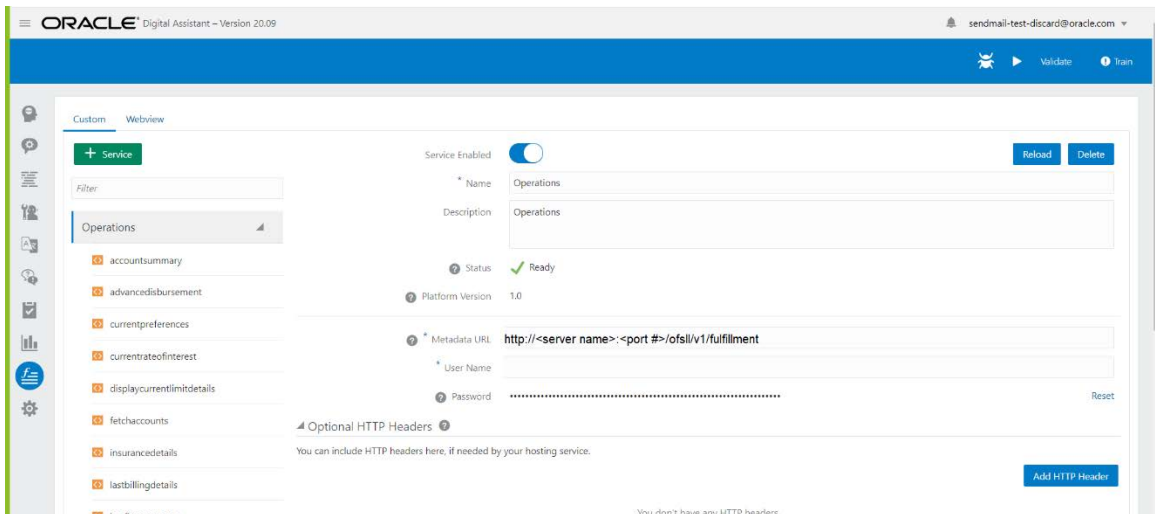
```
1
2 metadata:
3   platformVersion: "1.0"
4   main: true
5   name: TestLease1
6   context:
7     variables:
8       temp: string
9
10 #variables for common purpose
11 iResult: nIresult
12 response: string
13 acc_nbr: string
14 transactiontype: string
15 acc_list: list
16 asset: string
17 oldAsset: string
18 serviceType: string
19 phoneNumber: string
20
21
22 #variables for fetch accounts
```

A 'Saving changes...' indicator is visible at the bottom right of the code editor.

7. Add OFSLL REST Service



The screenshot shows the Oracle Digital Assistant interface with a message indicating that no services are configured for the custom components used by the skill bot. The message reads: "You haven't configured any services for the custom components used by this skill bot." Below the message, it says: "Add the names, URLs, and authentication settings for the services that implement each of the custom components that you've used in a flow." There is a "Tell me more" link and a green "+ Service" button.

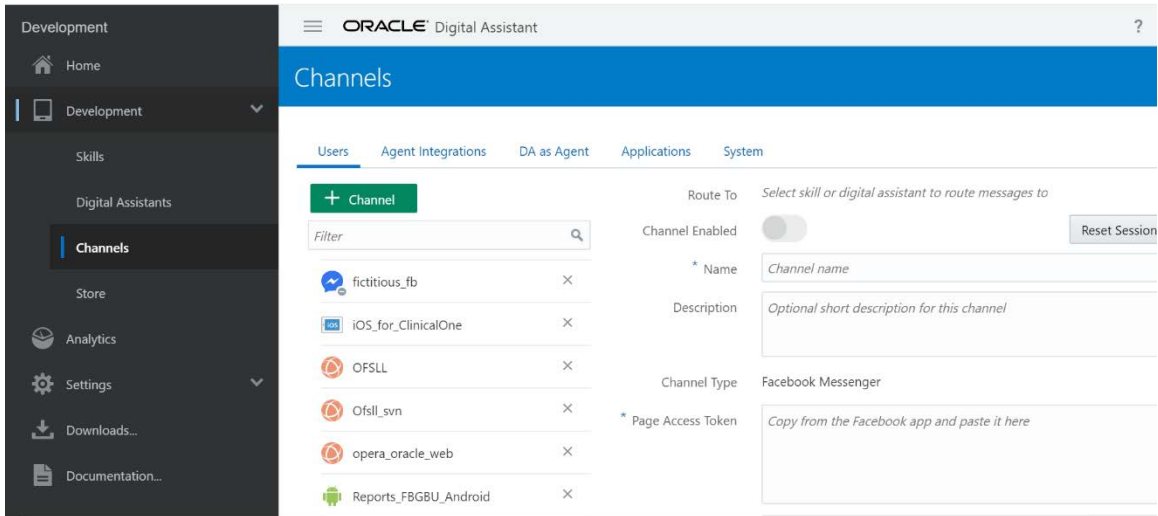


The screenshot shows the Oracle Digital Assistant interface with the configuration for a REST service named 'Operations'. The service is enabled and ready. The configuration details are as follows:

- Service Enabled:
- Name: Operations
- Description: Operations
- Status: Ready
- Platform Version: 1.0
- Metadata URL: `http://<server name>:<port #>/ofsl/v1/fulfillment`
- User Name: [Empty field]
- Password: [Empty field]

There is an "Optional HTTP Headers" section with a note: "You can include HTTP headers here, if needed by your hosting service." and an "Add HTTP Header" button.

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



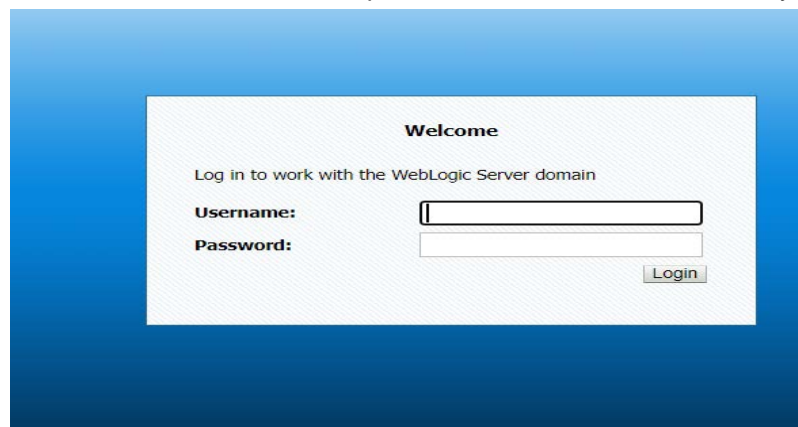
10. After completion of Skill, publish. On publishing, the draft is converted to final non-editable version and only final published version is accessible in bot.

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

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

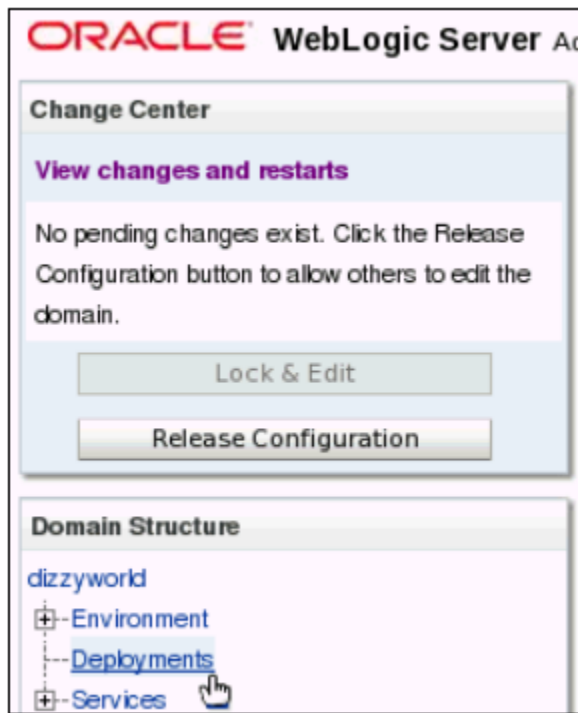
Note: Use the host name and port of the administration server of your domain.



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'.



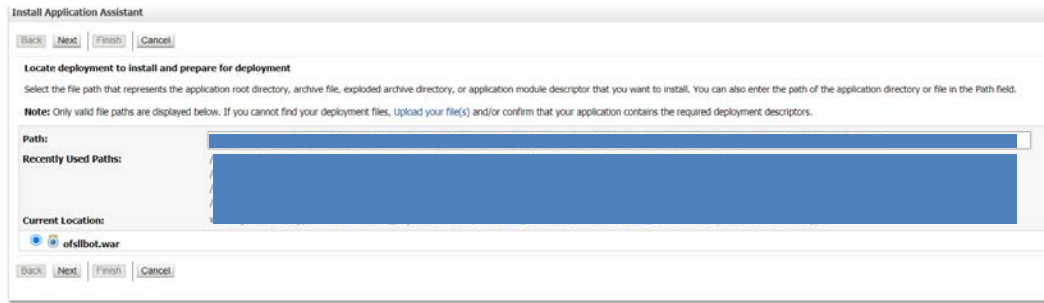
4. Under Domain Structure, click 'Deployments'.



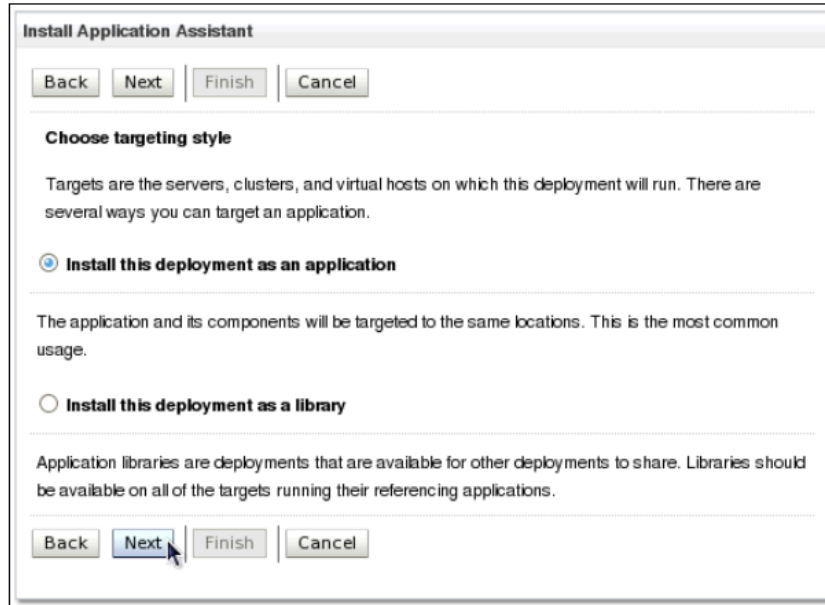
5. On the right, under Deployments, click 'Install'.



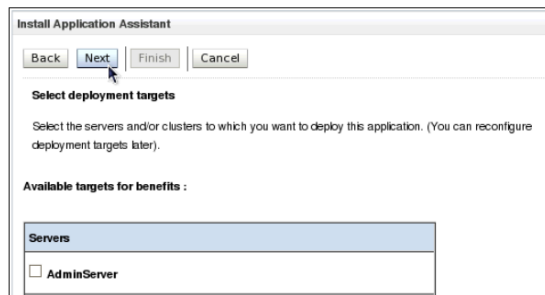
6. Find the Current Location field. Use the links to browse to the location in which you placed the downloaded ofslbot.war. War file.
7. On locating ofslbot.war, click the radio button next to it. Using the links and the radio button, the console auto populates the Path fields. Alternatively, you can 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'.

The screenshot shows the 'Install Application Assistant' dialog box. At the top, there are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'. Below the buttons is the 'Optional Settings' section, which includes a sub-section for 'General' with a text prompt 'What do you want to name this deployment?' and a text input field labeled 'Name:'. Below that is a 'Security' section with the prompt 'What security model do you want to use with this application?' and three radio button options: 'DD Only: Use only roles and policies that are defined in the deployment descriptors.' (which is selected), 'Custom Roles: Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor.', and 'Custom Roles and Policies: Use only roles and policies that are defined in the Administration Console.'. Below the security section is a 'Source accessibility' section with the prompt 'How should the source files be made accessible?' and one selected radio button option: 'Use the defaults defined by the deployment's targets'.

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

The screenshot shows the 'Install Application Assistant' dialog box. At the top, there are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'. Below the buttons is the 'Review your choices and click Finish' section, which includes a text prompt 'Click Finish to complete the deployment. This may take a few moments to complete.'. Below that is an 'Additional configuration' section with the prompt 'In order to work successfully, this application may require additional configuration. Do you want to review this application's configuration after completing this assistant?' and two radio button options: 'Yes, take me to the deployment's configuration screen.' and 'No, I will review the configuration later.' (which is selected).

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.

The screenshot shows the Oracle J2EE Administration Console interface. At the top, under the 'Messages' section, there are two green checkmark messages: 'The deployment has been successfully installed.' and 'You must also activate the pending changes to commit this, and other updates, to the active system.'

Below the messages is the 'Summary of Deployments' section, which has two tabs: 'Control' (selected) and 'Monitoring'. A text block explains that this page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. It notes that installed applications and modules can be started, stopped, updated (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page. It also states that to install a new application or module for deployment to targets in this domain, the user should click the Install button.

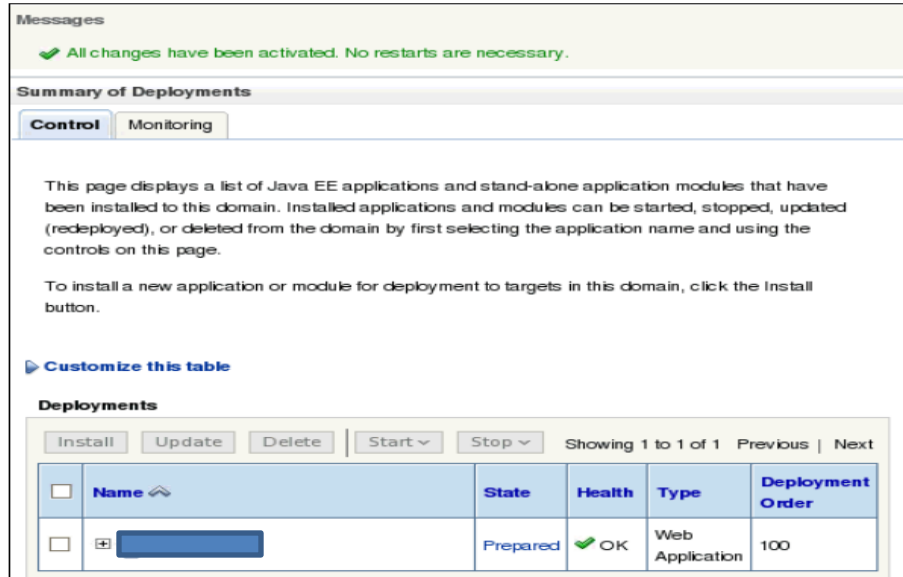
There is a link to 'Customize this table'.

The 'Deployments' section features a table with columns for Name, State, Health, Type, and Deployment Order. Above the table are buttons for 'Install', 'Update', 'Delete', 'Start', and 'Stop', along with 'Showing 1 to 1 of 1' and 'Previous | Next' navigation options. The table contains one entry with a checkbox, a plus sign, and a redacted name. The 'State' is 'distribute', 'Health' is 'Initializing', 'Type' is 'Web Application', and 'Deployment Order' is '100'. Below the table, there are identical buttons and navigation options as above.

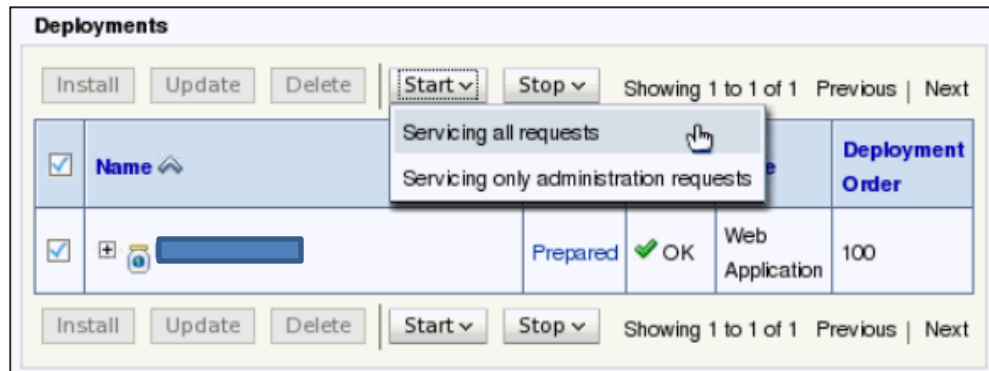
12. In the Change Center, click the Activate Changes button.

The screenshot shows the Oracle Change Center interface. It has a title bar 'Change Center' and a sub-header 'View changes and restarts'. Below this, a text block states: 'Pending changes exist. They must be activated to take effect.' At the bottom, there are two buttons: 'Activate Changes' (with a green checkmark icon) and 'Undo All Changes'. A mouse cursor is pointing at the 'Activate Changes' button.

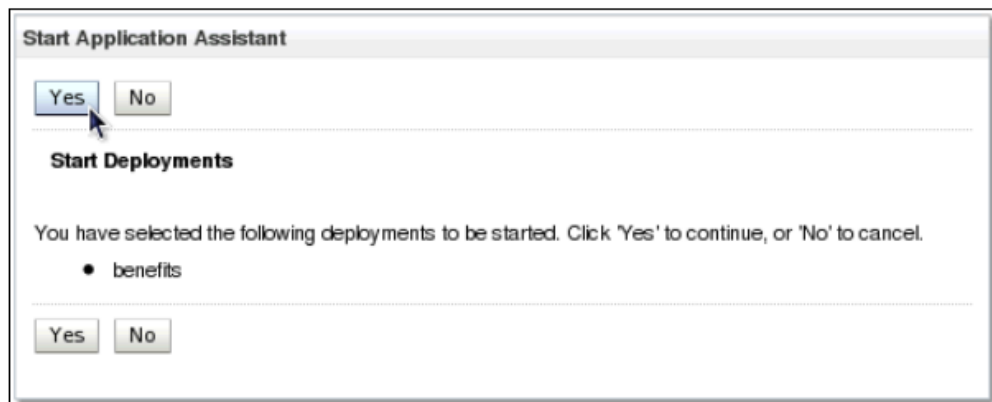
Notice the message indicating that the changes have been activated. In addition, notice the benefits application listed in the Deployments table is now in the "Prepared" state.



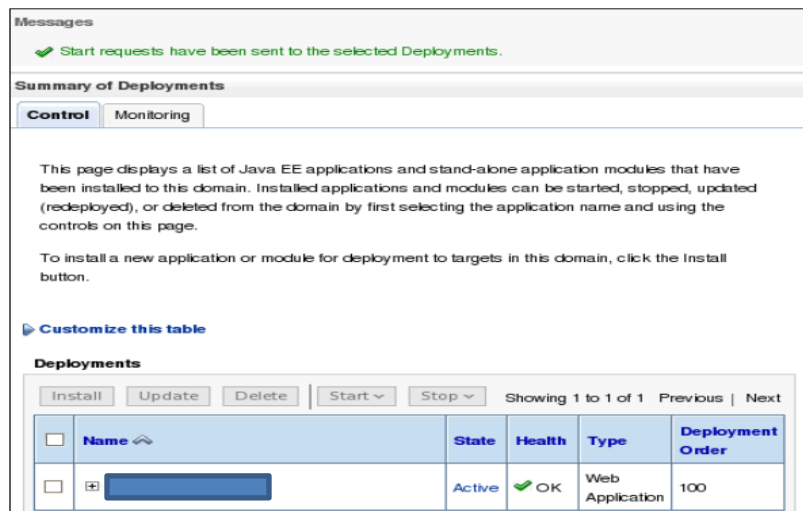
13. Select the checkbox against the left of the benefits application in the Deployments table. In the Start drop-down list, select 'Servicing all requests' option.



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 WebApp(or WebApp.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.

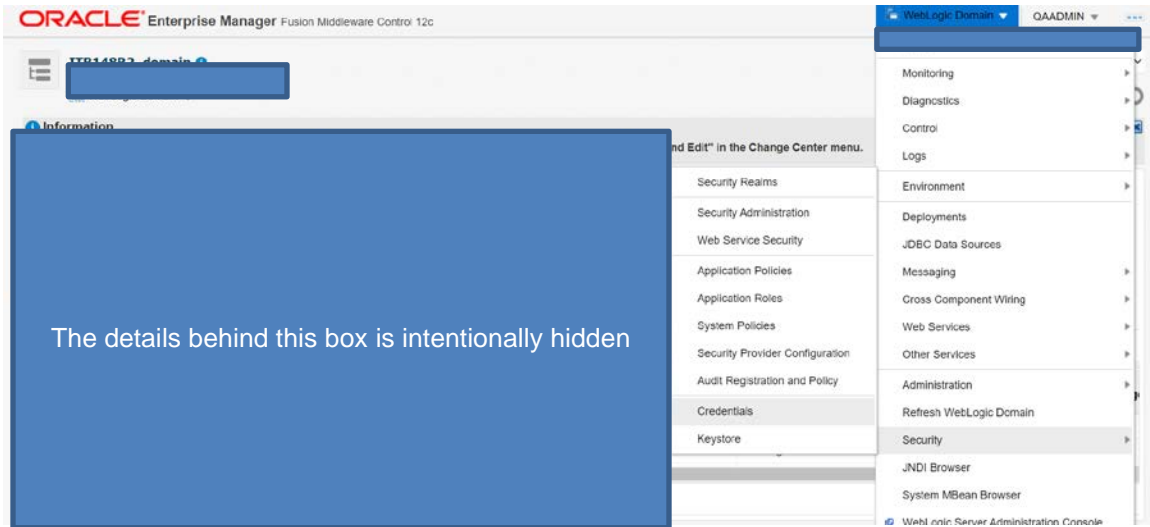
2.6 Configure CSF Mapping in Weblogic

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

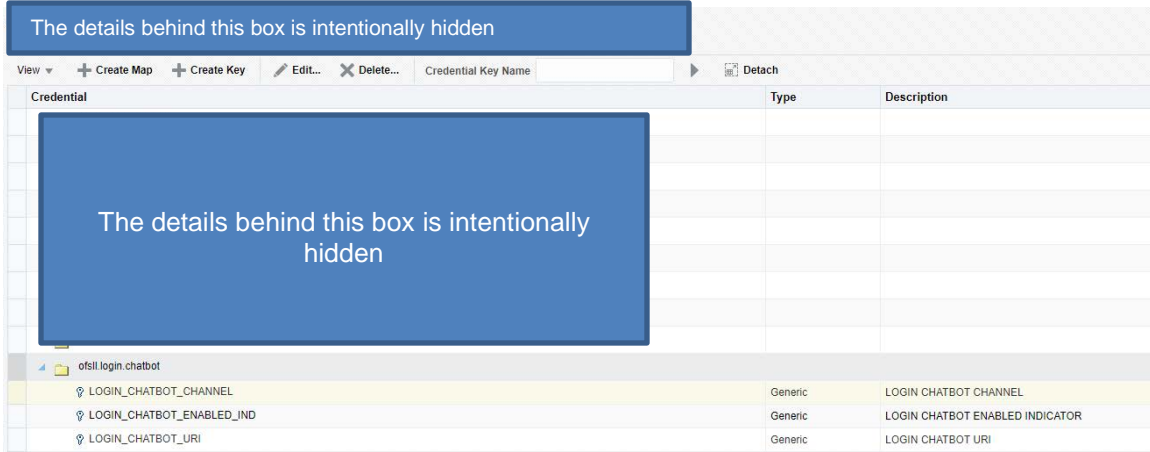
Following are the Parameters:

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

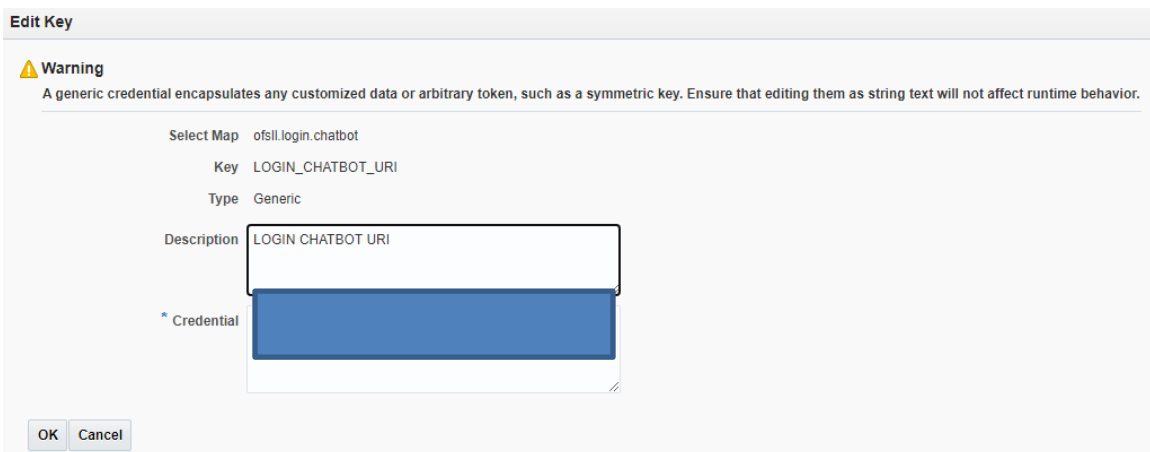
1. Login to the Weblogic server.



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



3. Add URI



4. Enter BOT INDICATOR.

Edit Key

Warning
A generic credential encapsulates any customized data or arbitrary token, such as a symmetric key. Ensure that editing them as string text will not affect runtime behavior.

Select Map ofssl.login.chatbot
Key LOGIN_CHATBOT_ENABLED_IND
Type Generic

Description LOGIN CHATBOT ENABLED INDICATOR

* Credential Y

OK Cancel

5. Enter BOT CHANNEL ID

Edit Key

Warning
A generic credential encapsulates any customized data or arbitrary token, such as a symmetric key. Ensure that editing them as string text will not affect runtime behavior.

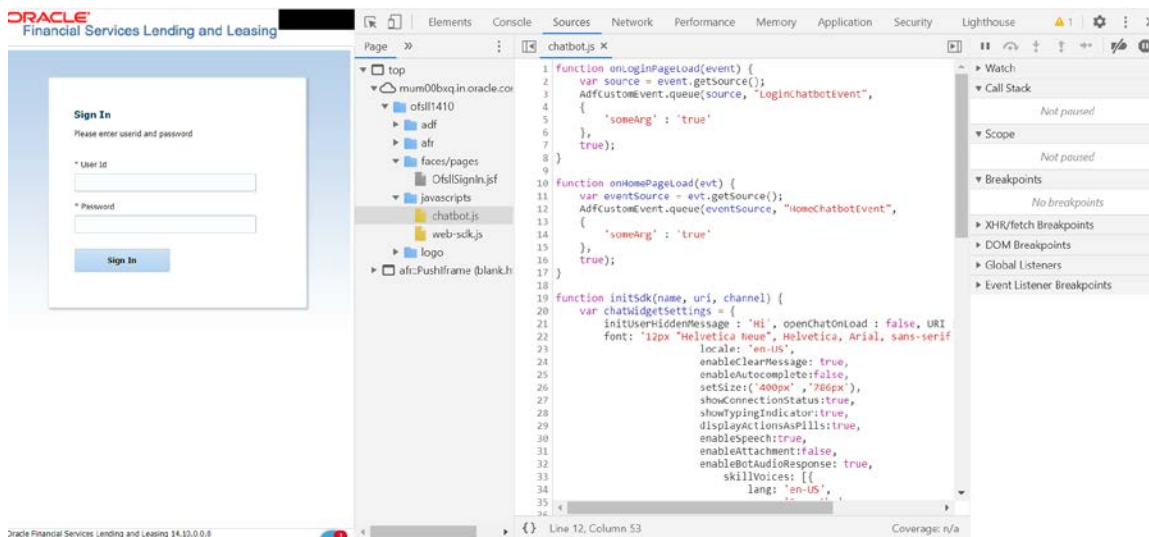
Select Map ofssl.login.chatbot
Key LOGIN_CHATBOT_CHANNEL
Type Generic

Description LOGIN CHATBOT CHANNEL

* Credential

OK Cancel

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';
    }
}

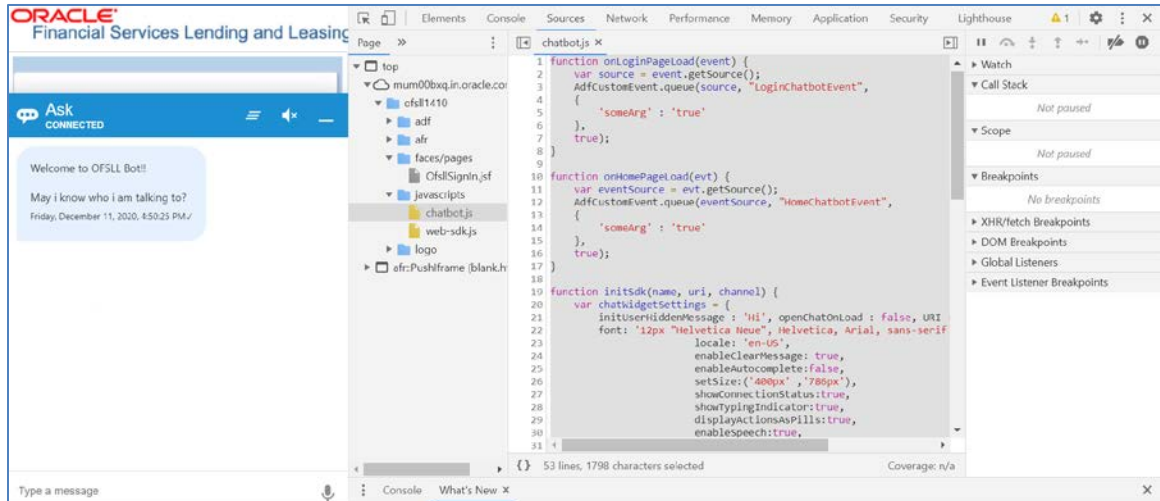
```

```

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

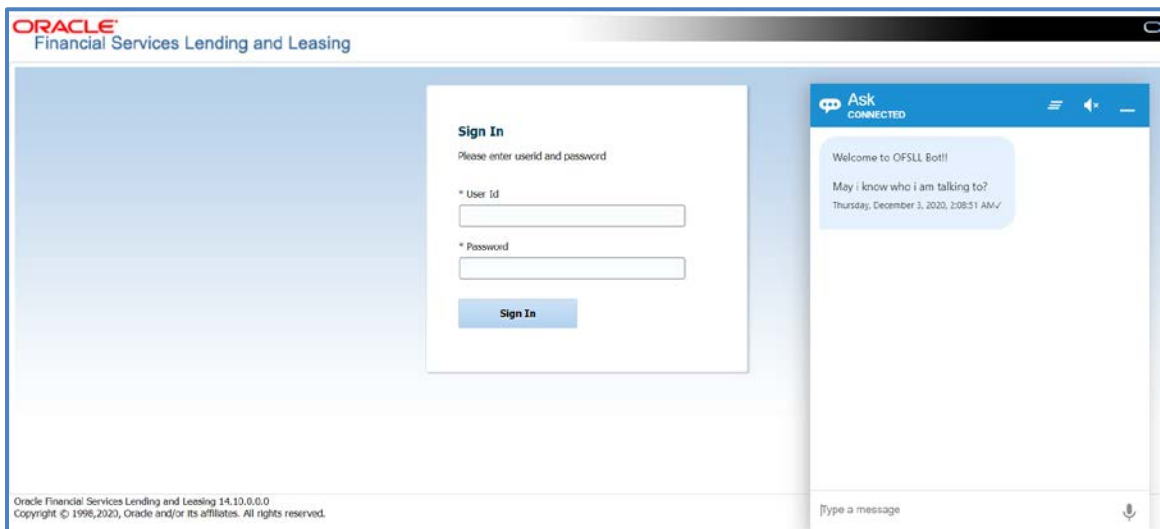
```

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/ofslbot/WebApp/scripts.

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



ORACLE®

Financial Services

CHATBOT Integration with OFSLL
Oracle Financial Services Lending and Leasing Release 14.10.0.0
December 2020

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>

Copyright © 1998, 2020, 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 recompilation 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.