OAuth2 based Web Services Access Authentication
Oracle Financial Services Lending and Leasing
Version 1.0
May 2019

ORACLE
Financial Services
# Table of Contents

1. INTRODUCTION ................................................................................................................................. 1-1  
   1.1 BACKGROUND ................................................................................................................................. 1-1  
   1.2 PURPOSE ......................................................................................................................................... 1-1  
   1.3 ABBREVIATIONS ............................................................................................................................. 1-1  

2. WEB SERVICES AUTHENTICATION USING OAuth2 ........................................................................... 2-1  
   2.1 UNDERSTANDING OAuth SERVICES ............................................................................................... 2-1  
      2.1.1 Identity Domains ......................................................................................................................... 2-2  
      2.1.2 Clients ......................................................................................................................................... 2-2  
      2.1.3 Resource Server ......................................................................................................................... 2-2  
      2.1.4 Resource Owner .......................................................................................................................... 2-2  
      2.1.5 Types of OAuth REST API ....................................................................................................... 2-2  

3. ENABLING OAuth SETUP CONFIGURATIONS .................................................................................... 3-3  
   3.1 Enabling OAuth Support for OFSLL REST APIs .............................................................................. 3-3  
   3.2 Identity Domain Creation ................................................................................................................ 3-3  
   3.3 Resource Server Creation ................................................................................................................ 3-5  
   3.4 Client Creation ................................................................................................................................... 3-7  
   3.5 Getting Access Token ....................................................................................................................... 3-8  
      3.5.1 How OFSLL API works with access token? ............................................................................... 3-8  
      3.5.2 Access Token for CLIENT_CREDENTIALS grant type .............................................................. 3-9  
      3.5.3 Access Token for PASSWORD grant type ................................................................................... 3-10  
      3.5.4 Access Token for JWT_BEARER grant type .............................................................................. 3-11  
      3.5.5 Access Token for REFRESH_TOKEN grant type ..................................................................... 3-13  
      3.5.6 How to get access token through Basic Authentication .......................................................... 3-15  
      3.5.7 How to access the REST API using the access token .............................................................. 3-16  
   3.6 Embedding External Application within OFSLL .............................................................................. 3-16
1. Introduction

1.1 Background

Oracle Financial Services Lending and Leasing (OFSLL) suite is a comprehensive, end-to-end solution that supports full lifecycle of direct and indirect consumer lending business with Origination, Servicing and Collections modules. This enables financial institutions to make faster lending decisions, provide better customer service and minimize delinquency rates through a single integrated platform. It addresses each of the lending processes from design through execution. Its robust architecture and use of leading-edge industry standard products ensure almost limitless scalability.

To extend OFSLL SaaS, OAuth2 can be used for securing OFSLL web services user access Authentication. This document details the process of web services authentication using OAuth services and enabling OAuth setup configurations.

1.2 Purpose

The purpose of this document is to provide detailed information for consulting and partner teams to implement an OAuth2 based REST API access authentication mechanism for OFSLL customers.

1.3 Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Detailed Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFSLL</td>
<td>Oracle Financial Services Lending and Leasing</td>
</tr>
<tr>
<td>IDM</td>
<td>Identity Management</td>
</tr>
<tr>
<td>OAuth</td>
<td>Open Authorization</td>
</tr>
<tr>
<td>SaaS</td>
<td>Software as a service</td>
</tr>
<tr>
<td>PaaS</td>
<td>Product as a service</td>
</tr>
<tr>
<td>OAM</td>
<td>Oracle Access Management</td>
</tr>
<tr>
<td>API</td>
<td>Application Program Interface</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>JWT</td>
<td>JSON Web Token</td>
</tr>
<tr>
<td>CSF</td>
<td>Critical success factor</td>
</tr>
</tbody>
</table>
2. Web services authentication using OAuth2

Web services authentication using OAuth2 is one of the best approaches for securing user authentication to extend OFSLL SaaS. This uses Oracle/Non-Oracle PaaS to authenticate service access requests from an external partner application without sharing OFSLL environment access credentials (UID/Password) and leverages the built-in support for OAuth 2.0.

OAuth 2.0 is an open standard token-exchange technology for verifying a user’s identity across multiple systems and domains without risking the exposure of a password.

Third-party applications (those not hosted on Oracle Cloud PaaS) can use OAuth for making calls into OFSLL Cloud REST APIs. PaaS/On-Premise application can pass a user’s authentication information and request an OAuth token from OFSLL Cloud, and then use the token to interact with an OFSLL Cloud API. PaaS or On-Premise and SaaS components can be with different ID Domains and security is managed with Shared IDM.

2.1 Understanding OAuth Services

Oracle Access Management (OAM) implemented the OAuth core 2.0 specifications to offer OAuth services. OAuth is an open standard authorization protocol that provides authentication and access control between a Client (such as Web services) and a Resource Owner (Service Provider) on the web.
2.1.1 Identity Domains

The Identity domains are entities that contain all artifacts required to provide standard OAuth services. Identity domains are independent entities and the primary use of this is to provide multi-tenants deployments. Each Identity domain will correspond to a tenant. This will also be useful for cloud deployments where each Identity domain can correspond to a separate tenant or entity.

Following are some of the components configured within an OAuth services Identity domain.

- One or More Clients
- One or More Resource Servers

2.1.2 Clients

The client is an application which makes protected resource requests on behalf of the resource owner using its authorization. For example, OFSLL. The Client initiates the OAuth Protocol by invoking the OAuth services. The client may be public or confidential.

There are two types of clients:

- **Confidential Clients**: Web Applications are of confidential client types assigned with a client ID and secret key. These clients can interact with the OAuth services server by sending the Client ID and secret as part of an authorization header.
- **Public Clients**: Public Clients or untrusted clients are assigned with a client ID but no secret key. These are the type of external applications that are not capable of keeping a client password confidential.

2.1.3 Resource Server

The Resource server is the machine on which protected resource is hosted. The Resource server is deployed in a different location from OAM and Client. The Resource server needs to be capable of accepting and responding to protected resource requests using access tokens.

2.1.4 Resource Owner

This is an entity capable of granting access to a protected resource. When the resource owner is a person, it is referred as an end-user.

2.1.5 Types of OAuth REST API

OAuth services are enabled as part of OAM version 12c Installation process. OAM provides an API based approach for configuring OAuth Services. There are 2 types of API OAuth services providers namely Admin API and Runtime API.

The Admin API provides capability to create mandatory admin components like Identity domain, Resource Server and client etc. They must be configured before the client makes the token request.

**Note**: To Execute Admin API, you can refer to Oracle OAM OAuth REST API documentation available at [https://docs.oracle.com/en/middleware/idm/access-manager/12.2.1.3/oroau/api-admin-identity-domain.html](https://docs.oracle.com/en/middleware/idm/access-manager/12.2.1.3/oroau/api-admin-identity-domain.html).
3. **Enabling OAuth Setup Configurations**

3.1 **Enabling OAuth support for OFSLL REST APIs**

The OAuth support for OFSLL REST API can be enabled with the following steps:

1. Add context Parameters in web.xml

2. Remove URL Security constraint tags in web.xml

   Add the below configuration in web.xml of OfsslRestWS.ear:
   
   `<context-param>
   <description>This parameter will decide the jersey filter to be loaded</description>
   <param-name>OAUTH_AND_BASIC_ENABLED</param-name>
   <param-value>Y</param-value>
   </context-param>

3. Remove Security configuration from weblogic.xml as well.

   **Note:** If this context parameter is not set, only the existing basic authentication flow is supported.

3.2 **Identity Domain Creation**

To create identity domain, any valid reliable REST client application/tool can be used to invoke the REST API. For example, Postman tool

http:<AdminServerHost:Port>/oam/services/rest/ssa/api/v1/oauthpolicyadmin/oauthidentitydomain
**Request JSON payload**

```
{
    "name": "OFSLL_OAUTH.DOMAIN",
    "identityProvider": "OUD_LDAP",
    "description": "OFSLL_OAUTH.DOMAIN",
    "tokenSettings": [
        {
            "tokenType": "ACCESS_TOKEN",
            "tokenExpiry": 3600,
            "lifeCycleEnabled": true,
            "refreshTokenEnabled": true,
            "refreshTokenExpiry": 86400,
            "refreshTokenLifeCycleEnabled": true
        },
        {
            "tokenType": "AUTHZ_CODE",
            "tokenExpiry": 3600,
            "lifeCycleEnabled": true,
            "refreshTokenEnabled": true,
            "refreshTokenExpiry": 86400,
            "refreshTokenLifeCycleEnabled": true
        },
        {
            "tokenType": "SSO_LINK_TOKEN",
            "tokenExpiry": 3600,
            "lifeCycleEnabled": true,
            "refreshTokenEnabled": true,
            "refreshTokenExpiry": 86400,
            "refreshTokenLifeCycleEnabled": false
        }
    ],
    "errorPageURL": "/oam/pages/error.jsp",
    "consentPageURL": "/oam/pages/consent.jsp",
    "customAttrs": "Attribute of user in IDStore to store the encrypted secretkey for TOTP"
}
```

**Response JSON payload**

Sucessfully created entity - OAuthIdentityDomain, detail - OAuth Identity Domain :: Name - OFSLL_OAUTH.DOMAIN, Id - 37b278eb5e894085ab1656b9641cca1a, Description - OFSLL_OAUTH.DOMAIN, TrustStore Identifiers - [OFSLL_OAUTH.DOMAIN], Identity Provider - OUD_LDAP, TokenSettings - [{
    "tokenType": "ACCESS_TOKEN",
    "tokenExpiry": 3600,
    "lifeCycleEnabled": true,
    "refreshTokenEnabled": true,
    "refreshTokenExpiry": 86400,
    "refreshTokenLifeCycleEnabled": true
}]

---

ORACLE
Financial Services

3-4
3.3 Resource Server Creation

**Resource Server Name**: OFSLL_OAUTH_SERVER

**Identity Domain**: OFSLL_OAUTH_DOMAIN
**Request JSON payload**

```
{
    "name":"OFSLL_OAUTH_SERVER",
    "description":"OFSLL_OAUTH_SERVER",
    "scopes": [
        {
            "scopeName":"OFSLL_REST_ALL",
            "description":"ALLOW_ALL"
        },
        {
            "scopeName":"OFSLL_REST_NONE",
            "description":"ALLOW_NONE"
        }
    ],
    "tokenAttributes": [
        {
            "attrName":"sessionId",
            "attrValue":"$session.id",
            "attrType":"DYNAMIC"
        },
        {
            "attrName":"resSrvAttr",
            "attrValue":"RESOURCECONST",
            "attrType":"STATIC"
        }
    ],
    "idDomain":"OFSLL_OAUTH_DOMAIN",
    "audienceClaim": {"subjects": ["OFSLL_B2B_OAUTH_CLIENT"]}
}
```

**Response JSON payload**

Successfully created entity - OAuthResourceServer, detail - IdentityDomain="OFSLL_OAUTH_DOMAIN",
Name="OFSLL_OAUTH_SERVER", Description="OFSLL_OAUTH_SERVER",
resourceServerId="99a3e782-ce6d-467c-baec-df87fe326a6",
resourceServerNameSpacePrefix="OFSLL_OAUTH_SERVER.",
audienceClaim="{
    "subjects": ["OFSLL_B2B_OAUTH_CLIENT"]",
    resServerType="CUSTOM_RESOURCE_SERVER",
    Scopes="[
        {
            "scopeName":"OFSLL_REST_ALL",
            "description":"ALLOW_ALL",
        },
        {
            "scopeName":"OFSLL_REST_NONE",
            "description":"ALLOW_NONE",
        },
        {
            "scopeName":"DefaultScope",
            "description":"DefaultScope"
        }]
    tokenAttributes="[
        {
            "attrName":"sessionId",
```
3.4 Client Creation

Name: OFSLL_B2B_OAUTH_CLIENT

idDomain: OFSLL_OAUTH_DOMAIN

{
  "attributes": [{
    "attrName": "customeAttr1",
    "attrValue": "CustomValue",
    "attrType": "static"
  }],
  "secret": "welcome1",
  "id": "OFSLL_B2B_OAUTH_CLIENT",
  "scopes": [
    "OFSLL_OAUTH_SERVER.OFSLL_REST_ALL",
    "OFSLL_OAUTH_SERVER.OFSLL_REST_NONE"
  ],
  "clientType": "CONFIDENTIAL_CLIENT",
  "idDomain": "OFSLL_OAUTH_DOMAIN",
  "description": "Client Description",
  "name": "OFSLL_B2B_OAUTH_CLIENT",
  "grantTypes": [
    "PASSWORD", "CLIENT_CREDENTIALS",
    "JWT_BEARER", "REFRESH_TOKEN",
    "AUTHORIZATION_CODE"
  ],
  "defaultScope": "OFSLL_OAUTH_SERVER.OFSLL_REST_ALL"
}
Response JSON payload

Sucessfully created entity - OAuthClient, detail - OAuth Client - uid = 236936a6-ed77-4d6a-bc282554a1a0,
name = OFSLL_B2B_OAUTH_CLIENT, id = OFSLL_B2B_OAUTH_CLIENT,
identityDomain = OFSLL_OAUTH_DOMAIN,
description = Client Description, secret = welcome1, clientType = CONFIDENTIAL_CLIENT,
grantTypes = [PASSWORD, CLIENT_CREDENTIALS, JWT_BEARER, REFRESH_TOKEN, AUTHORIZATION_CODE],
attributes = [{
  "attrName": "custmeAttr1",
  "attrValue": "CustomValue",
  "attrType": STATIC
},
{
  "attrName": "sessionId",
  "attrValue": "$session.id",
  "attrType": DYNAMIC
},
{
  "attrName": "resSrvAttr",
  "attrValue": "RESOURCECONST",
  "attrType": STATIC
}],
scopes = [OFSLL_OAUTH_SERVER.OFSLL_REST_ALL, OFSLL_OAUTH_SERVER.OFSLL_REST_NONE],
defaultScope = OFSLL_OAUTH_SERVER.OFSLL_REST_ALL, redirectURIs = []

3.5 Getting Access Token

A client application which wants to obtain an access token from OAuth server can access OFSLL Authentication API which in turn accesses the OAM OAuth API and generates token. The authentication REST service OFSLL provides a wrapper around OAM OAuth API.

3.5.1 How OFSLL API works with access token?

1. Client calls OFSLL authentication API (OFSLL REST API) with required headers along with body and obtains the token as response.
2. OFSLL REST API validates the token and retrieves the user ID from access token.
3. If the token is valid, then provides access to the protected resource.

Note: To use OAM OAuth API, update the following OFSLL system parameters with valid values.

<table>
<thead>
<tr>
<th>OFSLL System Parameter Name</th>
<th>Default Value</th>
<th>Actual Value Required to configure OAuth Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAM_OAUTH_ENABLED_IND</td>
<td>SETME</td>
<td>Y</td>
</tr>
</tbody>
</table>
### OFSLL System Parameter Name

<table>
<thead>
<tr>
<th>OFSLL System Parameter Name</th>
<th>Default Value</th>
<th>Actual Value Required to configure OAuth Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAM_OAUTH_TOKEN_URL</td>
<td>SETME</td>
<td>http://&lt;hostname&gt;:&lt;port&gt;/oauth2/rest/token</td>
</tr>
</tbody>
</table>

While client applications are allowed to access OAM OAuth REST API directly, it is recommended for clients to access OFSLL Authentication REST API for all token generation and token validation features.

**Authentication Resource URL:**

http://<<hostname>>:<<port>>/<<context_path>>/service/api/resources/auth/token

### 3.5.2 Access Token for CLIENT_CREDENTIALS grant type

**Request JSON payload**

```json
{
  "AuthRequest": { "GrantType": "CLIENT_CREDENTIALS" }
}
```

**Mandatory Request Headers**

<table>
<thead>
<tr>
<th>Headers</th>
<th>Expected Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-OAUTH-IDENTITY-DOMAIN-NAME</td>
<td>OFSLL_OAUTH_DOMAIN</td>
</tr>
<tr>
<td>Authorization</td>
<td>Bearer &lt;Base64encoded value of client credentials&gt;</td>
</tr>
</tbody>
</table>
Response JSON payload

```
{
  "AuthResponse": {
    "Token": "eyJraWQiOiJPRlNMTF9TU09fVEVTVF9ET01BSU4iLCJ4NXQiOiJjQldCa0pqV2JVdHRHczFmZFdIYzdeE0tMWsiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJodHRwOi8vbnVtMDBjaWUuaW4ub3JhY2xLmNvbToxNDEwMC9vYXV0aDItLCJhWQiOiJtdLCJieHAIQjE1NDU2NyYyNDAsIiI6Ik9GUI0xMX1NTT19URVNUX0RPTUFJTiJ9.djXIN9ogjgGuxSFlSb_tMo9KSSnUgWar7OfjOsqML0X0i0Kc7sT9KO_R8rC_q8T8HpGTumx-mXbhH_XuWhSojnPvH_O3Jj3EMSp5_w9iVrKVXe1teW9uduO1kOmieLlyY328bonSIT3ljtsviMhDfn06k1Qk7WIFb9CZmkAdEd-B4lZW2hsYr2tQHvbpyj62tbgDXkJbeKSA8GR62JRGy4Z2u9tAmDGk17sxFlAFyPquzrLVLh_fmEBvI3jVfacJmDcQP2Pwry8QYppXwulsaf996kdB4OsAti5iKaDMZkToYxwfgzatIMZYYPMng871bEMqGSGw",
    "Expires_in": 3600,
    "TokenType": "Bearer",
    "Result": {
      "Status": "SUCCESS",
      "StatusDetails": "Token Generated Successfully"
    }
  }
}
```

3.5.3 Access Token for PASSWORD grant type

Request JSON payload

```
{
  "AuthRequest": {
    "UserName": "OFSLLSUPR",
    "Password": "Demo1234",
    "GrantType": "PASSWORD"
  }
}
```

Mandatory Request Headers

<table>
<thead>
<tr>
<th>Headers</th>
<th>Expected Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-OAUTH-IDENTITY-DOMAIN-NAME</td>
<td>OFSSL_OAUTH_DOMAIN</td>
</tr>
<tr>
<td>Authorization</td>
<td>Bearer &lt;Base64encoded value of client credentials&gt;</td>
</tr>
</tbody>
</table>
3.5.4 Access Token for JWT_BEARER grant type

This is the grant type to achieve the seamless SSO between the different mixes of application. This grant type provides facility to link the mainstream application SSO session with OAuth token. When the SSO session is generated, JWT User token also generated. The generated JWT User token has the SSO “session_id” as part of its claims. The consumer client application must call OFSSL Authentication API with JWT_BEARER token grant type to get access token to access the protected resource.

Note: The rules of SSO session are applied to the OAuth Access token.
Sample Request JSON

```
{
  "AuthRequest": {
    "Assertion": "eyJraWQiOiJkZWZhdWx0liwieDV0ljoiYkw5VIDI4bHhMQjJ0cW5xd2d4Y0FOQnotQXZzlwiYXnljoiUIMyNTYifQ.ejyJleHAiOjE1NDU2NDQ0NTEsImlmpoaSl6IIRUS0lssDdWR1VyWVhVbHdyZ2lJUOWciLCJpYXQiOjE1NDU2NDk4NTEnN1YiI6k9GUxMU1VQULisInNlc3Npb25faWQiOiJCeW90c2h6LzR3bhekVHcnNqWnJbPT1bV52eU5DaEtLa29xTk5tcUlyQkUvM3IOUTBInENYYWITQktqWXdY1JiazdQYXBzajN6a1pkbnJqYWViOURPbViRTFBSURocG1QbTd1hKUDVFdzRpbmZHTEs1VGlsYldDYUJLW0vYkxQnM5K2FaY1oxQ25oUT0VVFSU3ciLCJkb21haW4iW4OiJkJZWzdWx0In0.NfLQHdh219p2NjzR44q9xgrQm6ky1paJ2GpHf2Re8tXjKyZNFXjYu9Tb78RoX3-xlsXOdmmRJBMw0_z1vy-0NrnHkU2fpBrBVdaqswXadCCKFFnkYy8AAJZg2WXyUNmaAcZWPT9z3svcQBHq9QQMdrkUvq3WbD91LbS5MA5pOkU8LofMn2j8nisoLraQ904CXiiKPl8jWILXtai-8hHgz5t62Z-B-Yis3m1ixWJPJ7zEcMRoule5pyFRYHxwudBht3Y9M04uDEQaIAk3d0uiVDup4eFJBt-Vt1Jt42f5hX28GyQQNu13s-rVArAXYxHGx4hzNZTw9E UdDPuEg", "GrantType": "JWT_BEARER"
  }
}
```

### Mandatory Request Headers

<table>
<thead>
<tr>
<th>Headers</th>
<th>Expected Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-OAUTH-IDENTITY-DOMAIN-NAME</td>
<td>OFSLL_OAUTH_DOMAIN</td>
</tr>
<tr>
<td>Authorization</td>
<td>Bearer &lt;Base64encoded value of client credentials&gt;</td>
</tr>
</tbody>
</table>
3.5.5 Access Token for REFRESH_TOKEN grant type

Sample JSON Request

```
{
  "AuthRequest": {
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
    "GrantType": "REFRESH_TOKEN",
"RefreshToken": "Mtn+NHd1zyCwelBiAcfy3w==~ALGXo2ieTTwnvk8tk5Hoi5L4FN900wW PCpZgmHMAx8PgbwORU/HH/HTGtamC5Yu4Pqp3ZR7mva/Tzl68EHOxsfY0e3pBG/1 KX2CkfeLOMOPqszE1Lq7bkfQxKj9jZNI1FwEHnFVLPmgh2XR0xSjbjTK3eE5Eyoxy7aaT NU79CBM3Ws5ntrKHkJ2i2oLftTAT/w/thEQLK9A7vXI7Gj8cSAuuFsAREF/Ay/skriZZfMQK xKY3Ewth1Cm3cmoBmG5aT3DFe9LxPvSmght1cLKvDhjk0nOoKU01XsHwPh0cWXge WBlkKXvzdso6h0eJkSaTvFIUEEictWAgymGdISVVDfri2nHVQAO5TQsXFskoTPtrjA14C PWcriOLyYRdiPlOBc3j2c+CMmywIcLK+vv4+xh44lBu92fVW/HHzPmAWD3jIPyM2AvgP15Ed 8LBHcrlwyr8rR8JTkJpWkNMmxW4S0p/6g/dU/WD9QECGwUnBIlcwUplRoA76k2Gbqgq +NJ18ohF8epeUy/+ftvjejefNuU4I1Kl3mprlReHuG00l5GJScwC31w1uEUgJsBZc4QN9Fju Js/l/aMTPKRT683x33WLOLc3NkHOCW+g8XyeiSWa/c/k2DtyPWpnCuzNbzqvyjx93Zga o0vBmIDB9+hZZlNMdyOVdLcvwegAubETFpLdEmcu/2BplJXJLjgj8hMd9xE5xvr/7NDg+sUspmkrn806rEq0rzWILMgC2a105PmckGAM2RdU6Du/2CERPHTGrKp uFvdvZoyjh4JMFtWtH6f1KUV4AMgE++vPzQNEngSxxseewavmJbJbB0jGvuC/G00hdY3 YYWvz2CtEWAO8261unvuro1CeNFbTSMK/Z3x4l8uCf8n7Lyc9a3m8WTVsoodkLnZS 8mYU4dFEsxVS355wuyZvXR0BkJ44+2VpdmB55qOPZiZk2UKZJHg5jqcTcWSDmg +9qWvhwppdzFlqETF0e6U7FQkotXkEHUTMOSOTgR4d9paOJQDISW3Lr1n8+7YNA/r KIU7IoP6l7lcl/rZ2BDazmVgf3axz/Z/Oo+xbtJCAbrfQgQNAJLioCbf/hJMKtSX+osj4CairKg cteFPziEOeo5+sbaSThAadalLycPs6/4mNoK7yywLoxlOEY7CSXZSAPootsV49LX3fEjH gkvkDU3dhcPpc9DmlYDdySKO18KwWgPnjtcSU1fqa2wVWdmwR60ZsAIsn3dHqGdU +XTGr7dt7aq3JyxmtiuZQrGjIlbPj5gExgnS1JyjF2co755xXvWom3O/p8="

<table>
<thead>
<tr>
<th>Headers</th>
<th>Expected Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-OAUTH-IDENTITY-DOMAIN-NAME</td>
<td>OFSSL_OAUTH_DOMAIN</td>
</tr>
<tr>
<td>Authorization</td>
<td>Bearer &lt;Base64encoded value of client credentials&gt;</td>
</tr>
</tbody>
</table>
Sample JSON Response

```json
{
  "AuthResponse": {
    "Token": "eyJraWQiOiJPRlNMTF9TU09fVEVTVF9ET01BSU4iLCJ4NXQiOiJjQlJlczNpQ25hNnB2SmZkZXdqNnlSe0ZjZmFkZTVmdGFyMVZwUml4aFp4bmUyNTV5VWdtZhN1QzZnQ0b0JjWWl6Ul91U1BnQVUyX2Y1aG52MmZ0eG1iNzZ5a2dzcWxkZmJpNlZ5YzB5cGxuc2F2YiIsImF1dF90b21lIjoiQ09PS0lFX09UIn0."
  },
  "Expires_in": 3600,
  "TokenType": "Bearer",
  "Result": {
    "Status": "SUCCESS",
    "StatusDetails": "Token Generated Successfully"
  }
}
```

### 3.5.6 How to get access token through Basic Authentication

**Mandatory CSF Key**

<table>
<thead>
<tr>
<th>CSF Map name</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>ofssl.int.common</td>
<td>ofssl.jwt.JwtSecretKey</td>
</tr>
</tbody>
</table>
The Ofsljwt.JwtSecretKey refers to the secret that must be associated at the time of token generation. This is the key would be used to validate the token.

### Mandatory Request headers

<table>
<thead>
<tr>
<th>Headers</th>
<th>Expected Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Type</td>
<td>application/json</td>
</tr>
<tr>
<td>Authorization</td>
<td>Bearer &lt;Base64encoded value of resource owner credentials&gt;</td>
</tr>
</tbody>
</table>

#### Request JSON payload

```json
{ "AuthRequest": {
    "GrantType" : "PASSWORD"
}}
```

#### Response JSON payload

```json
{
    "AuthResponse": {
        "Token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJPRlNMTFNVUFIiLCJCJpc3MiOiJPRlNMTF9SRVNUX0FQSSIsImV4ciI6MTU0NTY3MjEyMSwiaWF0IjoxNTQ1NjcxMjIxfQ.eyJTIj4h6VjJ59dXteth9Zy4b0ayz9XpT5j2Yu8zlHr4uvkKan-yvRgU10OSXhovdyw8zMl_ajqDLdESc_lZv3w",
        "Result": {
            "Status": "SUCCESS",
            "StatusDetails": "Token Generated Successfully"
        }
    }
}
```

### 3.5.7 How to access the REST API using the access token

In Every OFSLL REST API request, please send the following headers with correct values

#### Mandatory Request headers

<table>
<thead>
<tr>
<th>Headers</th>
<th>Expected Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ofssl_access_token</td>
<td>The valid access token received from any of the above mention flow</td>
</tr>
<tr>
<td>X-OAUTH-IDENTITY-DOMAIN-NAME</td>
<td>Valid OAuth Identity domain associated with access token</td>
</tr>
</tbody>
</table>

### 3.6 Embedding External Application within OFSLL

As part of subsequent releases of OFSLL, to embed external application within OFSLL base application, we would provide one external link each under origination, servicing and collection modules. The associated menu links can be enabled through access screens.
The URLs for External Link will be defined through System parameters. The following URL keys need to be defined with proper external link.

<table>
<thead>
<tr>
<th>URL Keys</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLL.ORG.EXT.ONE.URL</td>
<td>http://&lt;&lt;hostname&gt;&gt;:&lt;&lt;port&gt;&gt;/&lt;&lt;contextpath&gt;&gt;/index.html</td>
</tr>
<tr>
<td>FLL.SER.EXT.ONE.URL</td>
<td>http://&lt;&lt;hostname&gt;&gt;:&lt;&lt;port&gt;&gt;/&lt;&lt;contextpath&gt;&gt;/index.html</td>
</tr>
<tr>
<td>FLL.COL.EXT.ONE.URL</td>
<td>http://&lt;&lt;hostname&gt;&gt;:&lt;&lt;port&gt;&gt;/&lt;&lt;contextpath&gt;&gt;/index.html</td>
</tr>
</tbody>
</table>

Note: The base OFSLL SSO application would send the user identity token called ‘authorizedCode’ through the URL query parameter which should be read by external application to generate the actual access token (by calling authentication service) in order to access protected OFSLL Rest API.

**Request JSON payload for JWT_BEARER grant type**

```json
{
  "AuthRequest": {
    "Assertion": ".eyJraWQiOiJkZWZhdWx0Iiw5IjV5IjI4bHhMfQjJ0cW5xd2d4Y0FOQnotQXZliwiYWxniUjUIMyNTYifQ.eyJleHAiOjE1NDU2NzQ1MDEsImp0aSI6ImRdUFAobERSVFrX2Iw21aTDBV/HciLCJpYXQiOjE1NDU2NzA5MDEsIm1vd0FZIiwiY3V0aWQIbGAliiwiYWxnIjoiUlMyNTYiLCJleHAiOjE1NDU2NzQ1MDEsInZpZGVvIjoiVHciIn0."
    "GrantType": "JWT_BEARER"
  }
}
```

**Mandatory Request Headers**

<table>
<thead>
<tr>
<th>Headers</th>
<th>Expected Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-OAUTH-IDENTITY-DOMAIN-NAME</td>
<td>OFSLL_OAUTH_DOMAIN</td>
</tr>
<tr>
<td>Authorization</td>
<td>Bearer &lt;Base64encoded value of client credentials&gt;</td>
</tr>
</tbody>
</table>
Response JSON payload

```json
{
  "AuthResponse":{
    "Token": "eyJraWQiOiJPRlNMTF9TU09fVEVTVF9ET01BSU4iLCJ4NXQiOiJiQlIa0pqV2JvYRcHczFmZFdlYzYtdeE0tMWsiLChJhbGciOiJPSUzIjJ9.eyJpc3MiOiJPRlNMTF9CMkJfSkVUX0NMSUVOVCIsInNjb3BlIjpbIk9GU0xMX1NTT19URVNUX1NFUlZFUl9CMkIuQWNjb3VudERldGFpbHMiXSwiZG9tYWluIjoiV0ZTTExfU1NFVTF3In0.BKsWO1yBEmc_f0jCdG16DxzkTkkN805VHFs5QICrCWbSV
    },
    "TokenType": "Bearer",
    "Result": {"Status": "SUCCESS",
               "StatusDetails": "Token Generated Successfully"
            }
  }
}
```
OAuth2 based Web Services Access Authentication
Oracle Financial Services Lending and Leasing
May 2019

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

Copyright © 1998, 2019, 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.