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

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

Abbreviation	Detailed Description
OFSLL	Oracle Financial Services Lending and Leasing
IDM	Identity Management
OAuth	Open Authorization
SaaS	Software as a service
PaaS	Product as a service
OAM	Oracle Access Management
API	Application Program Interface
URL	Uniform Resource Locator
XML	Extensible Markup Language
JWT	JSON Web Token
CSF	Critical success factor

2. Web services authentication using OAuth2

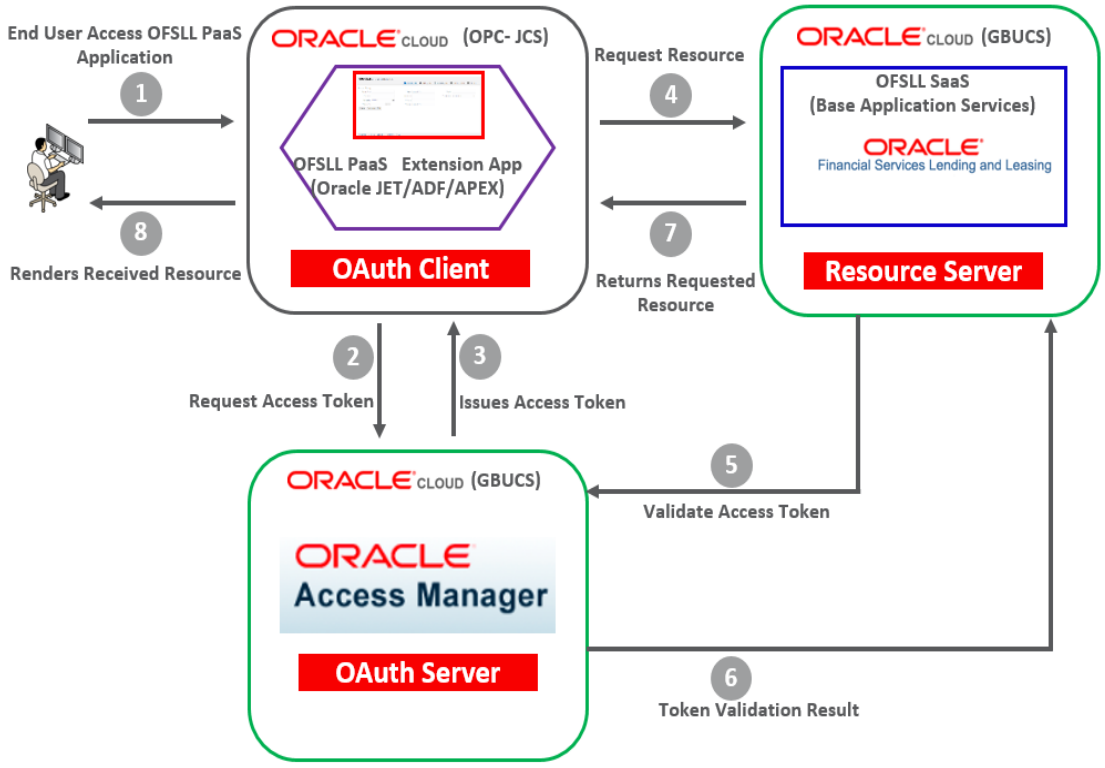
Web services authentication using OAuth2 is one of the best approach for securing user authentication to extend OFSLL SaaS. This uses Oracle / Non-Oracle PaaS to authenticate service access request from an external partner application without sharing OFSLL environment access credentials (UID / Password) 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.

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 OfsllRestWS.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
```

},
{
"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}],
ConsentPageURL - oam/pages/consent.jsp,
ErrorPageURL - /oam/pages/error.jsp,
CustomAttrs - Attribute of user in IDStore to store the encrypted secretkey for TOTP

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


```
Sucessfully created entity - OAuthResourceServer, detail -
IdentityDomain="OFSLL_OAUTH_DOMAIN,
Name="OFSLL_OAUTH_SERVER", Description="OFSLL_OAUTH_SERVER",
resourceServerId="99a3e782-ce6d-467c-baec-df687fe326a6",
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",
```

"attrValue":"$session.id",
"attrType":DYNAMIC},
{"attrName":"resSrvAttr","attrValue":"RESOURCECONST","attrType":STATIC}]

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-bcee-c0282554a1a0,

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":"customeAttr1",

"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.

OFSLL System Parameter Name	Default Value	Actual Value Required to configure OAuth Feature
OAM_OAUTH_ENABLED_IND	SETME	Y

OFSLL System Parameter Name	Default Value	Actual Value Required to configure OAuth Feature
OAM_OAUTH_TOKEN_URL	SETME	http://<hostname>:<port>/oauth2/rest/token
OAM_OAUTH_TOKEN_VALID_URL	SETME	http://<hostname>:<port>/oauth2/rest/token/info?access_token=<AccessToken>

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

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

Mandatory Request Headers

Headers	Expected Value
X-OAUTH-IDENTITY-DOMAIN-NAME	OFSLL_OAUTH_DOMAIN
Authorization	Bearer <Base64encoded value of client credentials>

Response JSON payload

{
"AuthResponse": {
"Token":
"eyJraWQiOiJPRlNMTF9TU09fVEVTVF9ET01BSU4iLCJ4NXQiOiJjQldCa0pqV2JVdHR
HczFmZFdlYzdteE0tMWsiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJodHRwOi8vbXVtMDBja
WUuaW4ub3JhY2xlLmNvbToxNDEwMC9vYXV0aDIiLCJhdWQiOltdLCJleHAiOjE1NDU2
NjYyNDAsImp0aSI6IjZrTDA4RGxhUXRGYXdLb0xRaGxnLVEiLCJpYXQiOjE1NDU2NjI2
NDAsInN1YiI6Ik9GU0xMX0IyQl9KRVRfQ0xJRU5UIiwiY2xpZW50IjoiT0ZTTExfQjJCX0pF
VF9DTElFTlQiLCJzY29wZSI6WyJPRlNMTF9TU09fVEVTVF9TRVJWRVJfQjJCLkFjY291
bnREZXRhaWxzIl0sImRvbWFpbiI6Ik9GU0xMX1NTT19URVNUX0RPTUFJTiJ9.djXIN9o
gjgGuxSFlSb_tMoi9KSSnUgWar7OfjOsqML0Xi0KSlvc7TKO_R8RvcQ_q8T8HpGTumx-
mXbhH_XuWhSojnpVH_O3Jj3EMSp5_w9IVIrKVanXej1teWP9udO1kOmieLLtyY328bon
SIT3ljtsviMhDfn06k1Qk7WlFb9CZmkAdEd-
B4lZV2hsYr2i2QHvbpyj62tbgtXDkJbeKSaGR62JRGyx4vZ2u9tAmDGk17sxFtAFyPquzrL
VLh_fmEBvI3jVfacJmDcQBP2Wry8QYppWxulsaf996kdB4OsAtlO5iK_DMZKtYOwxftgzat
lMZYYpIng871bEMqGFSGw",
"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

Headers	Expected Value
X-OAUTH-IDENTITY-DOMAIN-NAME	OFSLL_OAUTH_DOMAIN
Authorization	Bearer <Base64encoded value of client credentials>

Response JSON payload

{
"AuthResponse": {
"Token":
"eyJraWQiOiJPRlNMTF9TU09fVEVTVF9ET01BSU4iLCJ4NXQiOiJjQldCa0pqV2JVdHR
HczFmZFdlYzdteE0tMWsiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJodHRwOi8vbXVtMDBja
WUuaW4ub3JhY2xlLmNvbToxNDEwMC9vYXV0aDIiLCJhdWQiOltdLCJleHAiOjE1NDU2
NjYzMDIsImp0aSI6IjQ3WnF5Q1RNbHN1OE1yenp3Mnlzc3ciLCJpYXQiOjE1NDU2NjI3M
DIsInN1YiI6Ik9GU0xMU1VQUiIsIk9BVVRIX1RPS0VOIjoiZXlKcmFXUWlPaUprWldaaGR
XeDBJaXdpZURWMElqb2lZa3c1VkRJNGJIaE1RakowY1c1eGQyZDRZMEZPUW5vdFF
YWnpJaXdpWVd4bklqb2lVbE15TlRZaWZRLmV5SmxlSEFpT2pFMU5EVTJOall6TURJc
0ltcDBhU0k2SWpOSFZITjNOM1ZuTVhGQ2MwMXJOa1JIVFVGSWJsRWlMQ0pwWVhR
aU9qRTFORFUyTmpJM01ESXNJbk4xWWlJNklrOUdVMHhNVTFWUVVpSXNJbk5sYzN
OcGIyNWZhV1FpT2lKM2QwSTVkbU12Y21WSVFYQlpPRGxHZGxOeWVYVm5QVDEtY
jAxRlNIUkVibGcyVGtsRmRYVXZWM2hZZWtKTlFUMDlJaXdpWkc5dFlXbHVJam9pWkd
WbVlYVnNkQ0o5LkdKNlJOM19HUHNSQW0yTmGllbnQiOiJPRlNMTF9CMkJfSkVUX0N
MSUVOVCIsInNjb3BlIjpbIk9GU0xMX1NTT19URVNUX1NFUlZFUl9CMkIuQWNjb3VudE
RldGFpbHMiXSwiZG9tYWluIjoiT0ZTTExfU1NPX1RFU1RfRE9NQUlOIn0.YJJqj32KhCQf
YfvUzAi5XAhbBL3s8E29AiJxQXGqMrkDU57YIFt5l36bHjUGFRTNXnHZ2UxP5bhZiZJcm
ivOTqs_j1Iaz0-TkHKCbHX2_-8NhelwEXKtTyqx8-
9JKak1T8jsknXXKvO1FSv46siu2mBSxKul6rW7yeyC-TRiBBMj48h_u-
dlSfLQc98X_5jxXQU8FpCV18Cb9l2HbGh9zuUmEP8G87leYQ7KTMBWbcQklbAVQVxb
F0FVkua2efjW2LIz5XOJ_o_U-
6GvudCCiQvbeVY3VbUl4hgxJGXCs5e3ubQ9wPF8flCd05MAStFd30KzpeKxRtGZDXjuD
g3NSw",
"Expires_in": 3600,
"TokenType": "Bearer",
"RefreshToken":
"A79Gdo4lhOSCGmvmsRqWMg==~nHVr44Sa3QZgpl3eeplO8t323SjYEd3r6+IF24xBoct
9SxybWy6PcpHDjSoLTOmW+OcqtfqTenEmoIWCyfh0cTGzcmycH1KMOMfCGns+M2Kk
wusUCCGWnyrhoUevwhbKI4U20B3E6orBVkZxhtmQLqkATXbvHS0tGqlKIQwrgUCjNlws
SDFgBCj4umfQMIlt63pmgcKNtwpQcOedxB6y2B9fl3BFY8j2D53xogK3coE40pI4f+SufnZ
0Wl+0DkcCGHTfdaDzdcA2TwwoA5VVjZaQ16A+nCx144uHaBj1eOOpiUaypL730tK2N8a
ES1CSDUlZPjbl3NlEY360VvLJRoxdRq2nL4SgS0wJ7XIdu39wuxoTGtjLBWHQsDEtc0eB
bgFUma2q8ug29+67cl/9H6TwOEgF+T981H+7JQTCkSrma7gtyMr7MKy0QtmxR4Ns6w=
", "Result": { "Status": "SUCCESS", "StatusDetails": "Token Generated Successfully"
}
}
}

3.5.4 Access Token for JWT_BEARER grant type

This is the grant type is 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 OFSLL 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":"eyJraWQiOiJkZWZhdWx0IiwieDV0IjoiYkw5VDI4bHhMQjJ0cW5xd2d4Y0FO
QnotQXZzIiwiYWxnIjoiUlMyNTYifQ.eyJleHAiOjE1NDU2NDQ0NTEsImp0aSI6IlRUS0lsS
DdWR1VyWVhVbHdyZ2lUOWciLCJpYXQiOjE1NDU2NDA4NTEsInN1YiI6Ik9GU0xMU1
VQUiIsInNlc3Npb25faWQiOiJCeW90c2h6LzR3K2hhekVHcnNqWnJBPT1-
bVN2eU5DaEtLa29xTk5tcUIyQkUvM3lOUTBiNENYVWlTQktqWXdlY1JlazdQYXBzajN6a
1pkbnJqYWViOURPbWViRTFBSURocG1QN0tTd1hKUDVFdzRpbmZHTEs1VGlsYldDY
UJWL0VmVkIxQlM5K2FaY1oxQ25oUTV0VVFSU3ciLCJkb21haW4iOiJkZWZhdWx0In0.
NfLQHdh219p2NjzR44q9xgrQ9m6ky1paJ2GpHf2Re8tXjKyiZNFXjYu9Tb78RoX3-
xlsXOdmrRJBMw0_z1vy-
0NrnHkU2fpBrBVdauqsXadCCKFFnkYy8AAJZg2WXyUNmaAcZWPT9z3svcQBHq9OQ
MdrkUvq3WbD91LbS5MA5pOkU8LofMn2j8nisoLRaQ904CXillKPl8jWILXtai-
8hHgz5t62Z-B-
Yis3m1xiWPJ7zEctMRoule5pyFRYHxwudBht3Y9M04uDEQaIAk3d0uiVDup4eFJBt-
Vt1Jt42f5hX28GyQQNu13s-rVAraXYxHGx4hzNZZTlw9EUdDPuEg",
"GrantType" : "JWT_BEARER"
}
}
```


Mandatory Request Headers

Headers	Expected Value
X-OAUTH-IDENTITY-DOMAIN-NAME	OFSLL_OAUTH_DOMAIN
Authorization	Bearer <Base64encoded value of client credentials>

Sample JSON Response

{
"AuthResponse": {
"Token":
"eyJraWQiOiJPRlNMTF9TU09fVEVTVF9ET01BSU4iLCJ4NXQiOiJjQldCa0pqV2JVdHR
HczFmZFdlYzdteE0tMWsiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJodHRwOi8vbXVtMDBja
WUuaW4ub3JhY2xlLmNvbToxNDEwMC9vYXV0aDIiLCJhdWQiOltdLCJleHAiOjE1NDU2
NzQ1NzQsImp0aSI6InZTcE1LVzlheVF2VngxZU5KRUZ1ZVEiLCJpYXQiOjE1NDU2NzA
5NzQsInN1YiI6Ik9GU0xMU1VQUiIsImN1c3RvbWVBdHRyMSI6IkN1c3RvbVZhbHVlIiwic
2Vzc2lvbklkIjoiMzExY2MxOWUtODhkZi00ZDdhLTg5YzQtODFjNmRiMDE5ZDUzfEZqN2
NTcTZQcWlQUVZSNXU4TTNlRVpqbUJEZHVkQm9Hek5sMDBNQkFnS009IiwicmVzU3
J2QXR0ciI6IlJFU09VUkNFQ09OU1QiLCJjbGllbnQiOiJPRlNMTF9CMkJfSkVUX0NMSUV
OVCIsInNjb3BlIjpbIk9GU0xMX1NTT19URVNUX1NFUlZFUl9CMkIuQWNjb3VudERldGF
pbHMiXSwiZG9tYWluIjoiT0ZTTExfU1NPX1RFU1RfRE9NQUlOIn0.guHeG7eZilGWpJhM
WlIpH4K3lbGtM8buuwJPIlk6EengFTeicbfpd0E3qZwp8SYRFuzvw4FX7wCSbbBt2WM9G
4L6uM0NTvZpSTcwUeOljuysMiCmPzQ-
8cSijpM4G55Fb35laulC7eiCNdMtKoH34A2lScX7lamjlpC0u4SV4V-
8cB4VviGtrd_sXIqOfgSadpjrXQatuaRID1at4aNoAGv1Da7E4xrMzy9m41cxHtujSNU2aDx
G73-b2qOJiNZbvf-
zlaaa2pu1TOOr1ynZDvbe3STsZkAKO1VKFcHzmnYW8Tpqovc6MNd0TPyhNFUJHDBs
PH-nKV_nkFQHyu0_jw",
"Expires_in": 3600,
"TokenType": "Bearer",
"Result": {
"Status": "SUCCESS",
"StatusDetails":
"Token Generated Successfully"
}
}
}

3.5.5 Access Token for REFRESH_TOKEN grant type

Sample JSON Request

{
"AuthRequest": {
"GrantType" : "REFRESH_TOKEN",

"RefreshToken":"Mtn+NHd1zyCwelBiAcfy3w==~ALGXo2ieTTwvnk8tk5HOi5L4FN900wW
PCpZgmHMaDX8PgbwQRU/HH+TGtamC5YU4Pqp3ZRM7va/TzIc8EHEQxsfY0e3pBGl/
KD2CkfeL0MOPfqstzE1Lq7bkfQxKj9jZNIFvkEHnFVLPGh2XR0xSjIbTK3EeP5Eyoxf7aaT
NU79CBM3Ws5mrkGKhj2l2o2LfTTAbw/thEQLK9A7vXlGj8cSAuuFsAREfAy/skriZZfMQK
xKY3Ewh1Cm1coBmG5aT3DFe9LxgPuSmght1kLKqyDhjK0nOokUO1XsHwPh0cWXge
WBlKXVzd3o06h0eJ/SaTvFIUEEictWAgyMGDlSVVDfriZnHVQAO5TQsXFskoTPtrjA14C
PWoibLgYRdiPlObcj32c+CMmywlcLK/+v4+xh44lBu92fvW/HzPxAWDJ3lPyM2AvgP1SEd
8LiHcrwyh8rR8JTkJpWkNMmxW4S0p/6g//dUiWD9QEGCwUnBINweUpRoA76k2Gbgq+
NJ18ohF8epeUy+ftvjjeefNuoU41KI3mprlrReHuG00I5GJScwC3w1zuEUgJsBZc4QN9Fju
Js/I/aMTPKRT683x33WLQt3NkHOCW+g8XyeiSVaic/k2DtDyPWpnCuzNbZzqvjgx93Zga
o0vBMlDB9+hZzfNMybOVrIlcdvwegAubETgIpLEdmcu/2BpclXjLJgJ8hMd9xE5xrv/7NDg+
sUjpMkriN8OnEq0rzUlwLMgC2al05PMgkGA2RudhM/4VUXzKEK4ItGoR2CEMPHTGrKp
uFvdvZoyjh4JMfwTgH6F1KUV4AMgE+vPzQNENgSxxseewavmJbjB8OjGvuC/G00hdY3
YYWvz2CtEWAO8261unvuroUceNfGbTSK/Z3x4I8iuCfg8n7ZLyc9a3m8WTVsoodkLnZS
8mYU4dFExXvsS35gnuyzZvXR0BkJ44+2VpfdmB55qOPUziZK2UKZJ1Hg5jqTCwSMDg
+9qWVhbnpdzFtjqETF0edUI7F0QkotXkEHUTMSOTgR4d47paOJQDlSW3Lr1n8+7YNA/r
KlUTTIoP6I7lcl/rZ2BDazmVgf3axZ/Oo+xbtJCAbrfGqNJAJlOcBf/hJMktSXY+osj4CairKGh
cteFPziEOeo5+sbAsTthAadaLYcPs6/4mNoK7yvyLoxulOEY7CSXZSaPOotsV49LX3fEjH
gkvkDU3dhcPPc9DmlyDDySKO18K7wgaPnJtCSu1fq2AwVwDmwRd6BZsAIsn3dHqGDu
+XTgr7dt7ag3JyxmtuZQrGJiPbJp5gExgnS6JyjIF2co75kXvWoHm3O/p8="
}
}

Mandatory Request Headers

Headers	Expected Value
X-OAUTH-IDENTITY-DOMAIN-NAME	OFSLL_OAUTH_DOMAIN
Authorization	Bearer <Base64encoded value of client credentials>

Sample JSON Response

{
"AuthResponse": {
"Token":
"eyJraWQiOiJPRlNMTF9TU09fVEVTVF9ET01BSU4iLCJ4NXQiOiJjQldCa0pqV2JVdHR
HczFmZFdlYzdteE0tMWsiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJodHRwOi8vbXVtMDBja
WUuaW4ub3JhY2xlLmNvbToxNDEwMC9vYXV0aDIiLCJhdWQiOltdLCJleHAiOjE1NDU2
NjI4MTgsImp0aSI6InZGZ082eUlFb0k2X0xoZ3czcmczTUEiLCJpYXQiOjE1NDU2NjI3NT
gsInN1YiI6Ik9GU0xMU1VQUiIsIk9BVVRIX1RPS0VOIjoiZXlKcmFXUWlPaUprWldaaGRX
eDBJaXdpZURWMElqb2lZa3c1VkRJNGJIaE1RakowY1c1eGQyZDRZMEZPUW5vdFFY
WnpJaXdpWVd4bklqb2lVbE15TlRZaWZRLmV5SmxlSEFpT2pFMU5EVTJORFF4T0RBc
0ltcDBhU0k2SWpGSlZWbGlibXBFWkVwWVdWQk5TblJvUkVOTmIzY2lMQ0pwWVhRa
U9qRTFORFUyTkRBMU9EQXNJbk4xWWlJNklrOUdVMHhNVTFWUVVpSXNJbk5sYzN
OcGIyNWZhV1FpT2lJemNWRXplbVZ2TldWWlEzVnJhbk5qUW1aRWFFSlJQVDEtU2sx
amJHMW1TMW8wUmxKUk9GTnJSVXBPVXpCdGR6MDlJaXdpWkc5dFlXbHVJam9pW
kdWbVlYVnNkQ0o5LlpHQWdTZGR5S0szRGplMlZsTUlmbzMxTTV0cFBpaXluUVpGY2R
ibEFiV2xhektPWVFzb2hqbzdrODQxQm9SMWUtWXZWLXpWbk5hamhCYy1CbzAwZlNs
VDdsVmNmRXA2ekxUdENHRnc1MkNZRXpfOHpYdjc1YkF2Q2FWeGRvZnlUaGFhdFox
cVF5Qm5TLUJ1MGdKNjJSOThkaDFwY1FOSmFhWDZ1RkxkSGVoVUlDY1pNLUJwbU
drbi01TXhHcm5fQnloT2oxc0JnRjZ1SWY0N3d6NUlNOU4zODdHQ29WZDBPR3c0QXlm
VVk2T2FGSlNOS1hYbnpsYUVDTkktMEJmQXhFQklSX1oxVE1wZEdSOHkwaHltMWNT
SzRGYkJjYnQxZnlhYmdLMG1SQ0tFVHdMWFJrcnFMcmkxTnVtbjNKeFhmZVBsWTVJT
m1ndDA2U1BVaFpYUEU2ZyIsImN1c3RvbWVBdHRyMSI6IkN1c3RvbVZhbHVlIiwic2Vzc
2lvbklkIjoiQ09PS0lFX0JBU0VEIiwicmVzU3J2QXR0ciI6IlJFU09VUkNFQ09OU1QiLCJjbG
llbnQiOiJPRlNMTF9CMkJfSkVUX0NMSUVOVCIsInNjb3BlIjpbIk9GU0xMX1NTT19URVN
UX1NFUlZFUl9CMkIuQWNjb3VudERldGFpbHMiXSwiZG9tYWluIjoiT0ZTTExfU1NPX1R
FU1RfRE9NQUlOIn0.BKsWO1yBEmc_f0jCdG16DxzkTkkN805VmYlBbyMmmMqnziNsyc
orlzHAZ0RHTDqNLjKdq--
wxzTNQK4PRM9ChBeHKBCU5dzHD64ddbscyt0YxpdPnF0grMZHipIoNC_-
nZxyZRbLI5aQeGPXOZ4qtPEZ1ggBkgoXXa16eJ2JLZbYOtcvPbLcbkfHpMCz-
wOzi_o0t30KG9T1931NyMaCvYp4O-ZODTneHc9-
c7cJaj2zVhkOFej796TTrEHV4jv7p2OTsawkm8vSYmRBv5K1J8M_a1PgEIuqc4kS6d0op
UAJOKT6C356OMdEpeO_zkXGyfodUFKojdG3PWHXG0O7ww",
"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

CSF Map name	Key
ofsll.int.common	ofsll.jwt.JwtSecretKey

The Ofsll.jwt.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

Headers	Expected Value
Content-Type	application/json
Authorization	Bearer <Base64encoded value of resource owner credentials>

Request JSON payload

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

Response JSON payload


```
{
"AuthResponse": {
"Token":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJPRlNMTFNVUFIiLCJpc3MiOiJ
PRlNMTF9SRVNUX0FQSSIsImV4cCI6MTU0NTY3MjEyMSwiaWF0IjoxNTQ1NjcxMjIxfQ.
tM5JB4h6VjJ59dXteth9ZY4b0ayz9XpT5J2jYu8zIHr4uvkKan-
yvRgU1OSXhovdyw8zMl_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

Headers	Expected Value
ofsll_access_token	The valid access token received from any of the above mention flow
X-OAUTH-IDENTITY-DOMAIN-NAME	Valid OAuth Identity domain associated with access token

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.

Menu Access Keys
FLL.ORG.EXT.ONE.MENU
FLL.SER.EXT.ONE.MENU
FLL.COL.EXT.ONE.MENU

The URLs for External Link will be defined through System parameters. The following URL keys need to be defined with proper external link.

URL Keys	Example
FLL.ORG.EXT.ONE.URL	http://<<hostname>>:<<port>>/<<contextpath>>/index.html
FLL.SER.EXT.ONE.URL	http://<<hostname>>:<<port>>/<<contextpath>>/index.html
FLL.COL.EXT.ONE.URL	http://<<hostname>>:<<port>>/<<contextpath>>/index.html

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


```
{
"AuthRequest": {
"Assertion":"eyJraWQiOiJkZWZhdWx0IiwieDV0IjoiYkw5VDI4bHhMQjJ0cW5xd2d4Y0FO
QnotQXZzIiwiYWxnIjoiUlMyNTYifQ.eyJleHAiOjE1NDU2NzQ1MDEsImp0aSI6ImRDcUFa
bERSVFdrX2lwd21aTDBfVHciLCJpYXQiOjE1NDU2NzA5MDEsInN1YiI6Ik9GU0xMU1V
QUiIsInNlc3Npb25faWQiOiJEelVmNlZyN05FQ2NHYnVBbVVINU9RPT1-
NmdhemlUU2Y1OVdPb0FLZHRhc0h0R3R4L3p6TktSK0lEZDd5OG1OemRPZ2FMQUk2
NjlIRytEbzY0NFdGV3NWMTMwS1pUbDJrTWd1OHp5TTFEaWtteFdtYVUrK0dyUXlidGo
3WVB6dlRwdzlmbFVMT3d1cEFBbUpqMi9VLzlVYVUiLCJkb21haW4iOiJkZWZhdWx0In0
.Qi4gJ4kiEcaxgs51fRU3633RcPMDNjqOpRrnzBOq8M9pKIeRmNe2Zyu7ikBXqIjFMd0Iz-
N9hUgvD9i8-5lPeEER15_FqLsPtoCUX3u8NuPPfzqA_xT2LTcc0-
6AdGz7QrsqAU_qr3n2FGF5qhwiHU7437X_AzoMBwTYovWsl8Rjra_tdWoCMsMRisN7x
qIeW7Jk3aWYQeoOHbwfuVqDE18m67du9rUszNUrX483KXwCfZL1ffbqYIFYIGekGpm4
AbCq5aazK8-HtrmzKyt-
Q1Monx2dOrUorLKm6AtKXTOqdrA2YwASYM96A4ENLTdxjcIyTUDcwIu4WgeBUcwJzw
",
"GrantType" : "JWT_BEARER"
}
}
```


Mandatory Request Headers

Headers	Expected Value
X-OAUTH-IDENTITY-DOMAIN-NAME	OFSLL_OAUTH_DOMAIN
Authorization	Bearer <Base64encoded value of client credentials>

Response JSON payload

{
"AuthResponse":{
"Token":
"eyJraWQiOiJPRlNMTF9TU09fVEVTVF9ET01BSU4iLCJ4NXQiOiJjQldCa0pqV2JVdHR
HczFmZFdlYzdteE0tMWsiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJodHRwOi8vbXVtMDBja
WUuaW4ub3JhY2xlLmNvbToxNDEwMC9vYXV0aDIiLCJhdWQiOltdLCJleHAiOjE1NDU2
NjI4MTgsImp0aSI6InZGZ082eUlFb0k2X0xoZ3czcmczTUEiLCJpYXQiOjE1NDU2NjI3NT
gsInN1YiI6Ik9GU0xMU1VQUiIsIk9BVVRIX1RPS0VOIjoiZXlKcmFXUWlPaUprWldaaGRX
eDBJaXdpZURWMElqb2lZa3c1VkRJNGJIaE1RakowY1c1eGQyZDRZMEZPUW5vdFFY
WnpJaXdpWVd4bklqb2lVbE15TlRZaWZRLmV5SmxlSEFpT2pFMU5EVTJORFF4T0RBc
0ltcDBhU0k2SWpGSlZWbGlibXBFWkVwWVdWQk5TblJvUkVOTmIzY2lMQ0pwWVhRa
U9qRTFORFUyTkRBMU9EQXNJbk4xWWlJNklrOUdVMHhNVTFWUVVpSXNJbk5sYzN
OcGIyNWZhV1FpT2lJemNWRXplbVZ2TldWWlEzVnJhbk5qUW1aRWFFSlJQVDEtU2sx
amJHMW1TMW8wUmxKUk9GTnJSVXBPVXpCdGR6MDlJaXdpWkc5dFlXbHVJam9pW
kdWbVlYVnNkQ0o5LlpHQWdTZGR5S0szRGplMlZsTUlmbzMxTTV0cFBpaXluUVpGY2R
ibEFiV2xhektPWVFzb2hqbzdrODQxQm9SMWUtWXZWLXpWbk5hamhCYy1CbzAwZlNs
VDdsVmNmRXA2ekxUdENHRnc1MkNZRXpfOHpYdjc1YkF2Q2FWeGRvZnlUaGFhdFox
cVF5Qm5TLUJ1MGdKNjJSOThkaDFwY1FOSmFhWDZ1RkxkSGVoVUlDY1pNLUJwbU
drbi01TXhHcm5fQnloT2oxc0JnRjZ1SWY0N3d6NUlNOU4zODdHQ29WZDBPR3c0QXlm
VVk2T2FGSlNOS1hYbnpsYUVDTkktMEJmQXhFQklSX1oxVE1wZEdSOHkwaHltMWNT
SzRGYkJjYnQxZnlhYmdLMG1SQ0tFVHdMWFJrcnFMcmkxTnVtbjNKeFhmZVBsWTVJT
m1ndDA2U1BVaFpYUEU2ZyIsImN1c3RvbWVBdHRyMSI6IkN1c3RvbVZhbHVlIiwic2Vzc
2lvbklkIjoiQ09PS0lFX0JBU0VEIiwicmVzU3J2QXR0ciI6IlJFU09VUkNFQ09OU1QiLCJjbG
llbnQiOiJPRlNMTF9CMkJfSkVUX0NMSUVOVCIsInNjb3BlIjpbIk9GU0xMX1NTT19URVN
UX1NFUlZFUl9CMkIuQWNjb3VudERldGFpbHMiXSwiZG9tYWluIjoiT0ZTTExfU1NPX1R
FU1RfRE9NQUlOIn0.BKsWO1yBEmc_f0jCdG16DxzkTkkN805VmYlBbyMmmMqnziNsyc
orlzHAZ0RHTDqNLjKdq--
wxzTNQK4PRM9ChBeHKBCU5dzHD64ddbscyt0YxpdPnF0grMZHipIoNC_-
nZxyZRbLI5aQeGPXOZ4qtPEZ1ggBkgoXXa16eJ2JLZbYOtcvPbLcbkfHpMCz-
wOzi_o0t30KG9T1931NyMaCvYp4O-ZODTneHc9-
c7cJaj2zVhkOFej796TTrEHV4jv7p2OTsawkm8vSYmRBv5K1J8M_a1PgEIuqc4kS6d0op
UAJOKT6C356OMdEpeO_zkXGyfodUFKojdG3PWHXG0O7ww", "Expires_in": 3600,
"TokenType": "Bearer",
"Result": {
"Status": "SUCCESS",
"StatusDetails": "Token Generated Successfully"
}
}
}

ORACLE
Financial Services

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