

UK Open Banking Configuration Guide
Oracle Banking APIs
Patchset Release 22.2.1.0.0

Part No. F72988-01

May 2023

ORACLE®

UK Open Banking Configuration Guide

May 2023

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway

Goregaon (East)

Mumbai, Maharashtra 400 063

India

Worldwide Inquiries:

Phone: +91 22 6718 3000

Fax:+91 22 6718 3001

www.oracle.com/financialservices/

Copyright © 2006, 2022, 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 decompilation 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.

Table of Contents

1. Preface	1-1
1.1 Intended Audience.....	1-1
1.2 Documentation Accessibility	1-1
1.3 Access to Oracle Support.....	1-1
1.4 Structure	1-1
1.5 Related Information Sources	1-1
2. Objective and Scope	2-1
3. Technology Stack	3-1
4. Pre-requisites	4-1
5. Headers Configuration	5-1
6. Properties	6-1
7. OAuth Configuration	7-1
7.1 UI configuration.....	7-1
8. Extensibility and Code Conventions	8-1
8.1 Key Providers support	8-4
9. Keystore and Certificate for UK Open Banking Directory	9-7

1. Preface

1.1 Intended Audience

This document is intended for the following audience:

- Customers
- Partners

1.2 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

1.3 Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

1.4 Structure

This manual is organized into the following categories:

Preface gives information on the intended audience. It also describes the overall structure of the User Manual.

The subsequent chapters describes following details:

- Introduction
- Preferences & Database
- Configuration / Installation.

1.5 Related Information Sources

For more information on Oracle Banking APIs Patchset Release 22.2.1.0.0, refer to the following documents:

- Oracle Banking APIs Installation Manuals

2. Objective and Scope

Background

Open Banking Configuration Document provides the various configurations required to enable UK Open Banking in OBAPI

Scope

- Headers Configuration
- Properties
- OAuth Configuration
- Code Convention and Extensibility

[Home](#)

3. Technology Stack

Software	Version
Java	Java JDK or JRE version 8
OBDX/OBAPI	21.1.0.0.0
OAuth	OBAPI Internal OAuth

Abbreviations

OOTB	Out of the Box
TPP	Third Party Providers
ASPSP	Account Servicing Payment Service Provider

[Home](#)

4. Pre-requisites

- Java JDK or JRE version 7 or higher must be installed. For installation of Java please refer [installation guide](#).
- OAuth Setup

[Home](#)

5. Headers Configuration

There are two types of headers configuration available for UK Open Banking.

- System Headers (i.e. Mandatory Headers and its respective value validation)
- Configuration Headers (i.e. Mandatory Headers).

Below are the configuration steps and Out of the box header already configured in the system.

System Headers:- As of now in OOTB one header has been added as mandatory “x-fapi-financial-id” with value as “491308330388688” (This is a random value and can be changed. This value is issued by OBIE and corresponds to the Organization Id of the ASPSP in the Open Banking Directory). This value needs to be configured by Bank or ASPSP. This header needs to be sent by the TPP to the ASPSP mandatorily with the same value. Both Header name and Header value are validated for System Headers.

For configuring more system headers, below script is to be executed in the OBAPI Admin schema.

```
Insert into DIGX_FW_CONFIG_ALL_B (PROP_ID, CATEGORY_ID, PROP_VALUE,
FACTORY_SHIPPED_FLAG, PROP_COMMENTS, SUMMARY_TEXT, CREATED_BY,
CREATION_DATE, LAST_UPDATED_BY, LAST_UPDATED_DATE, OBJECT_STATUS,
OBJECT_VERSION_NUMBER) values ('uk%%HEADER
NAME%%','OpenbankingSystemHeaders','%%HEADERVALUE%%','N',null,'Open
Banking','ofssuser',sysdate,'ofssuser',sysdate,'Y',1);
```

Below Query is used to check the System Headers in the system

```
select * from digx_fw_config_all_b where category_id = 'OpenbankingSystemHeaders';
```

Configuration Headers :- As of now in OOTB one header has been added as mandatory - “x-fapi-interaction-id”. This header is required to be sent by the TPP to the ASPSP mandatorily with any value.

Only header name is validated in case of Configuration Headers.

For configuring more config headers, below script is to be executed in the OBDX/OBAPI Admin schema.

```
Insert into DIGX_FW_CONFIG_ALL_B (PROP_ID, CATEGORY_ID, PROP_VALUE,
FACTORY_SHIPPED_FLAG, PROP_COMMENTS, SUMMARY_TEXT, CREATED_BY,
CREATION_DATE, LAST_UPDATED_BY, LAST_UPDATED_DATE, OBJECT_STATUS,
OBJECT_VERSION_NUMBER) values ('uk%%HEADER NAME%%','
OpenbankingConfigHeaders',null,'N',null,'Open
Banking','ofssuser',sysdate,'ofssuser',sysdate,'Y',1);
```


Below Query is used to check the System Headers in the system

```
select * from digx_fw_config_all_b where category_id = 'OpenbankingConfigHeaders';
```

[Home](#)

6. Properties

Below are the properties required to be updated in the UK Open Banking. Please find the below properties, its purpose and OOTB values.

Table:- DIGX_FW_CONFIG_ALL_B

Category-Id :- OpenBankingConfig

Property Id	Property Value (Out of the Box)	Purpose
CONSENT_EXPIRY_DAYS	90	This value is used to check if expiry date send by TPP for the Account Access Consent is not more than 90 days and if it is more than 90 days then ASPSP will reject this value

Token Settings

Table:- DIGX_FW_CONFIG_ALL_B

Category-Id :- OAuthTokenConfig

Note: Prior to changing the value of OAUTH_TOKEN_SIGNER to X509RS256 or X509PS256, make sure to generate Public and Private Key Pair in Security Keys Section by logging in as admin.

Property Id	Property Value	Purpose
OAUTH_TOKEN_SIGNER	X509RS256 – x509 signed token with RS256 algorithm X509PS256 - x509 signed token with PS256 algorithm	The algorithm used to generate JWT token.
refreshTokenExpiry	86400	Default expiry for Refresh Token.
tokenExpiry	3600	Default expiry for Access Token.
ISSUER	OBAPI-OAUTH	Issuer of the access/refresh token.
AUDIENCE	OBAPITestResServer	Audience of the access/refresh token.
OPAQUE_ACCESS_TOKEN_FLAG	Values can be Y or N.	Flag to enable/disable opaque access token.
CODE_CHALLENGE_FLAG	Values can be Y or N.	Flag to enable/disable code challenge verification as per the FAPI requirement.

Common Settings

Table:- DIGX_FW_CONFIG_ALL_B

Category-Id :- OAuthCommonConfig

Property Id	Property Value	Purpose
OAUTH_REDIRECT_HOST_PORT	http://{{HOST}}:{{PORT}}	'HOST' refers to the hostname/IP of the application 'PORT' refers to the application's port

DCR(Dynamic Client Registration) Configs

Table:- DIGX_FW_CONFIG_ALL_B

Category-Id :- OAuthDCRConfig

Property Id	Property Value	Purpose
PUBLIC_KEY_FETCH_URL	e.g. https://keystore.openbankingtest.org.uk/keystore/openbanking.jwks	Open Banking Directory URL to fetch the public key for payload jwt verification.
SCIM_PRIVATE_KEY	This should be value of obseal_dec.key without any space or enter character	This is ASPSP's private key for signing the jwt payload while communicating to the Open Banking Directory.
SSA_REQUEST_PAYLOAD_PUBLIC_KEY	e.g. software_jwks_endpoint	This is TPP's SSA claimset value, which is used to fetch the public key to verify SSA.
DCR_REQUEST_PAYLOAD_PUBLIC_KEY	e.g. software_jwks_endpoint	This is TPP's claimset value, which is used to fetch the public key to verify DCR payload.
ID_TOKEN_PRIVATE_KEY	This should be value of obseal_dec.key without any space or enter character	This is ASPSP's private key for signing the jwt payload.
ID_TOKEN_PRIVATE_KEYID	This should be value of key id generated when the ASPSP's certificate is uploaded in the Open Banking Directory.	This is ASPSP's key id to fetch the ASPSP's public key from the Open Banking Directory by the TPP.
OBIE_CLAIM	iss	To identify the issuer claimset in the DCR payload.
OBIE_CLAIM_VALUE	OpenBanking Ltd	To identify the value of the issuer claimset in the DCR payload.
OBIE_MEMBSTATE_VALUE	GB	Member state of the SSA.
OBIE_SOFTENV_VALUE	Values can be sandbox/production.	To identify software environment. Value should be 'production' for the production environment.

DCR(Dynamic Client Registration) SCIM Configs to Connect to Open Banking Directory

Table:- DIGX_FW_CONFIG_ALL_B

Category-Id :- OAuthDCRSCIMConfig

Property Id	Property Value	Purpose
MTLS_CERTIFICATE_ALIAS	Alias which was used to create the MLTS certificate e.g. openbanking_obtrans	Required for communication over MTSL with the Open Banking Directory.
MTLS_CERTIFICATE_PWD	Password which was used to create the MLTS certificate	Required for communication over MTSL with the Open Banking Directory.
IDENTITY_STORE_PATH	Path of the identity store jks file. e.g. /scratch/obapi/wls/OpenBanking/SCIM/openbanking_custom_identity.jks	Required for communication over MTSL with the Open Banking Directory.
TRUST_STORE_PATH	Path of the trust store jks file. e.g. /scratch/obapi/wls/OpenBanking/SCIM/openbanking_custom_trust.jks	Required for communication over MTSL with the Open Banking Directory.
PROXY_ENABLED	Values can be Y/N.	To identify whether the proxy is enabled or not for the communication.
PROXY_URL	Value of the proxy url.	Required for communication over MTSL with the Open Banking Directory with proxy enabled.
PROXY_PORT	Value of the proxy port.	Required for communication over MTSL with the Open Banking Directory with proxy enabled.
HTTPS_ENABLED	Values can be Y/N.	To identify whether the https is enabled or not for the communication.
softwareStatementId	This should be the Software Statement Id of the ASPSP.	Required for communication over MTSL with the Open Banking Directory.
clientScopes	'TPPReadAll AuthoritiesReadAccess QTSPReadAccess'	These are the scopes defined by the Open Banking Directory.
keyId	This should be the ASPSP's key id to be used for the MSTL communication.	Required for communication over MTSL with the Open Banking Directory.
tokenUrl	https://matls-sso.openbankingtest.org.uk/as/token.oauth2	This is defined by the Open Banking Directory to get the access token for accessing the Open Banking APIs.

certUrl	https://matls-dirapi.openbankingtest.org.uk/certificate/validate	This is defined by the Open Banking Directory to get the ASPSP's certificate validated for the MTLS communication.
orgDetUrl	https://matls-api.openbankingtest.org.uk/scim/v2/OBThirdPartyProviders/	This is defined by the Open Banking Directory to get the organisation details.
aud	https://matls-sso.openbankingtest.org.uk/as/token.oauth2	This is defined by the Open Banking Directory for the 'audience' claimset for communication over MTSL.
iss	This should be the Software Statement Id of the ASPSP.	Required for communication over MTSL with the Open Banking Directory.
sub	This should be the Software Statement Id of the ASPSP.	Required for communication over MTSL with the Open Banking Directory.
grantType	client_credentials	This is defined by the Open Banking Directory for communication over MTSL.
clientAssertionType	urn:ietf:params:oauth:client-assertion-type:jwt-bearer	This is defined by the Open Banking Directory for communication over MTSL.

[Home](#)

7. OAuth Configuration

7.1 UI configuration

OAuth Identity Domain Maintenance will require below maintenance to configure UI Component for Authorizing consent.

The value of Consent Page URL (Menu -> OAuth -> Identity Domain Maintenance) is configured as <http://host:port?homeComponent=authorize-consent&homeModule=open-banking&applicationType=auth>.

[Home](#)

8. Extensibility and Code Conventions

Code Convention of Account API's

Accounts related API should use below arguments and return type for working with UK Open Banking

Arguments

SessionContext sessionContext

com.ofss.digx.app.openbanking.dto.accounts.uk.AccountRequestDTO
accountRequestDTO

Return Type

BaseResponseDTO<T>

Where T extends DataTransferObject

Any service implemented with the above type of argument will be compatible with UK Open Banking.

Code Convention of Payment API's

Payment related API should use below arguments and return type for working with UK Open Banking

Arguments

Create and Read Method

SessionContext sessionContext

Any DTO Object which extends com.ofss.digx.app.openbanking.dto.consent.uk.UKPaymentDTO

Any service implemented with the above type of argument will be compatible with UK Open Banking.

Error Message Framework

The Error Message Framework helps convert the OBAPI error response according to the UK Open Banking Specifications.

The error response structure for Open Banking Read/Write APIs is as follows:

```

{
  "Code": "...",
  "Id": "...",
  "Message": "...",
  "Errors": [
    {
      "ErrorCode": "...",
      "Message": "...",
      "Path": "...",
      "Url": "..."
    }
  ]
}
1.1

```

The UK Open Banking specified error response is handled using DIGX_OB_UK_OBAPI_ERROR_MAP table.

The contents of the table are as follows:

Column Name	Description
DIGX_ERROR_CODE	Represents the OBAPI error codes. This is a Primary and Unique Key
UK_ERROR_CODE	Represents the Open Banking specified error code
PATH	Represents the reference to the JSON Path of the field with error. Can be null.
URL	Represents the URL to help remediate the problem, or provide more information etc. Can be null.

For mapping OBAPI error codes with UK Open Banking specified codes below script can be used:

```
Insert into DIGX_OB_UK_OBAPI_ERROR_MAP
(DIGX_ERROR_CODE,UK_ERROR_CODE,PATH,URL) values ('%%OBAPI Error
Code%%',%%Open Banking specified error code%%', '%%Path%%', '%%URL%%');
```

For example –

```
Insert into DIGX_OB_UK_OBAPI_ERROR_MAP
(DIGX_ERROR_CODE,UK_ERROR_CODE,PATH,URL) values
('DIGX_OB_0010','UK.OBIE.Field.Missing', 'Data.Initiation ',null);
```

Below Query is used to check the OBAPI errors mapped with UK Open Banking specified error codes in the system

```
select * from DIGX_OB_UK_OBAPI_ERROR_MAP;
```

For configuring HTTP status codes with custom message, below script can be used:

```
Insert into DIGX_FW_CONFIG_ALL_B (PROP_ID, CATEGORY_ID, PROP_VALUE,
FACTORY_SHIPPED_FLAG, PROP_COMMENTS, SUMMARY_TEXT, CREATED_BY,
CREATION_DATE, LAST_UPDATED_BY, LAST_UPDATED_DATE, OBJECT_STATUS,
OBJECT_VERSION_NUMBER)
values ('%%HTTP Status code%%','OpenBankingErrorConfig','%%Error
Message%%','N',null,'OpenBanking Error Message','ofssuser',sysdate,'ofssuser',sysdate,'Y',1);
```

Below Query is used to check the Open Banking HTTP status codes in the system

```
select * from digx_fw_config_all_b where category_id = ' OpenBankingErrorConfig';
```

Permission Response Handler

Permissions is used in only Account API's. Based on Permissions, Response is generated based on permissions.

OBAPI consists of Permission Handler against each type of permissions. This configuration is available in the table **DIGX_OB_UK_PERMISSIONS_PRIMARY**

The contents of the table are as follows:

Column Name	Description
SERVICEID	Represents the OBAPI Service Id for which the permission and its handler is available
PERMISSION	Represents Permission
RESPONSEHANDLER	Represent Permission Handler

Permission Handler can be overridden or can be newly introduced. This will be required for additional fields mapping which is not available OOTB. Steps for the same are as follows

Introducing Permission Handler

New Permission Handler should implement interface IResponseHandler

New Permission Handler should have below methods

public static <T implements IResponseHandler> getInstance()

public <T extends DataTransferObject> assembleResponse(DataTransferObject object, List<String> permissions) – This method assembles response from object to the required response object which needs to be shown in the API response. Object is the response got from the base service and T will be the response object required by API specifications. Assembling of the values will be done in this method

public int getPriority() – This defines the high priority of the handler to be applied for assembling response in case of permissions and its handler has been consented by the user i.e. Basic and Detail permission will have different handlers but if the consent is for both permissions the priority of the handler will decide which needs to be executed on high priority.

8.1 Key Providers support

Key Providers Overview

Whenever TPP initiates a DCR request, the payload is signed with the TPP's private key and same needs to be verified with the TPP's public key at the Bank's side. There could be different ways to get the TPP's public key which can vary as per open banking directory services and the geographical regions.

To accommodate those varying approaches of getting the public key, OBAPI has provided a factory pattern to get a 'Key Provider'. The main job of the key provider is to get the public key of the TPP, to verify the DCR payload, based on the Software Statement Issuer Name.

To implement the above, one IKeyProvider interface is added. This contains the methods which may differ based on the parameters mentioned above.

```

2
3* import java.security.interfaces.RSAPublicKey;
5
6 public interface IKeyProvider {
7
8     public String getPublicKey(String clientId, String kid);
9
10    public Map<String, String> fetchPublicKey(String dcr_request_token);
11
12    public Map<String, String> getPublicKeyClaims(String x509Certificate, String keyId);
13
14    /**
15     * Derives the RSA public key from the Base64 public key/certificate
16     *
17     * @param encodedKeyOrCert
18     * @return
19     */
20    public RSAPublicKey getRSAPublicKey(String encodedKeyOrCert);
21 }

```

There are 4 methods to be implemented.

1. `public Map<String, String> fetchPublicKey(String dcr_request_token);` - to fetch the TPP's public key when the TPP is being onboarded with the bank with the help of DRC Request Token data.
2. `public String getPublicKey(String clientId, String kid);` - to fetch the TPP's public key based on the client id and the key id for further requests processing as and when required when the TPP is already onboarded with the bank.
3. `public Map<String, String> getPublicKeyClaims(String x509Certificate, String keyId);` - to get the various types of claims like certificate type, validity, expiry, revocation etc.
4. `public RSAPublicKey getRSAPublicKey(String encodedKeyOrCert);` - to get the decrypted RSA public key from the encoded key or extracted from the certificate.

In addition to above methods, to make the key provider class singleton, provider class must implement to return the singleton instance of the class

```
public static IKeyProvider getInstance();
```

Key Provider Implementation & Configuration

To create a key provider, one needs to create a KeyProvider class by extending the `com.ofss.digx.oauth2.spi.IKeyProvider` interface and making the provider class entry in the `DIGX_FW_CONFIG_ALL_B` table.

For example, we have a SSA Issuer called 'XYZ Ltd'.

We will need to follow below two steps to configure the XYZ key provider

1. Need to create a new key provider implementation class - `com.ofss.digx.openid.service.XYZKeyProvider` which must implement the `IKeyProvider` interface. Name and the package of the key provider class could be anything, those are not compelled to be same as the mentioned above, but it must implement the `IKeyProvider` interface.
2. Need to make the provider class entry in the `DIGX_FW_CONFIG_ALL_B` with `prop_id = 'XYZ Ltd_KEY_PROVIDER'`. In this entry, the naming convention should strictly be followed as `<SSA_Issuer>_KEY_PROVIDE` and the `CATEGORY_ID` must be 'openBankingConfig'.

To configure new key provider in DB, refer below insert query and its values are described as below:

```

Insert into DIGX_FW_CONFIG_ALL_B
(PROP_ID,CATEGORY_ID,PROP_VALUE,FACTORY_SHIPPED_FLAG,PROP_COMMENTS,S
UMMARY_TEXT,CREATED_BY,CREATION_DATE,LAST_UPDATED_BY,LAST_UPDATED_D
ATE,OBJECT_STATUS,OBJECT_VERSION_NUMBER,EDITABLE,CATEGORY_DESCRIPTOR
N) values ('XYZ
Ltd_KEY_PROVIDER','openBankingConfig','com.ofss.digx.openid.service.XYZKeyProvider','N',n
ull,'XYZ Ltd Key Provider Class','ofssuser',sysdate,'ofssuser',sysdate,'A',1,'N',null);

```

As per the current standards, there are mainly two open banking authorities in European Continent:

1. Open Banking Directory (OBD)
2. European Banking Authority (EBA)

A Third-Party Provider (TPP) gets registered with any of the above two authorities and obtains the Software Statement (SSA) before getting onboarded with the bank.

In this release, OBAPI has provided the out of the box implementation of key providers for both directory services.

1. com.ofss.digx.openid.service.OBDKeyProvider – for Open Banking Directory
2. com.ofss.digx.openid.service.EBAKeyProvider – for European Banking Authority

To get the public key, OBD has provided 'software_jwks_endpoint'. This endpoint provides a JSON Web Key Set (JWKS), which is a set of keys containing the public keys used to verify any JSON Web Token (JWT). Based on the key id, TPP's public key is extracted from the JWKS to verify the payload.

Both the key providers currently communicate with the Open Banking Directory to fetch the TPP's public key currently as per the implementation.

We have below two configurations:

1. OpenBanking Ltd_KEY_PROVIDER – to fetch the public keys of TPP's whose SSA Issuer is the 'OpenBanking Ltd'.
2. DEFAULT_KEY_PROVIDER - to fetch the public keys of TPP's whose SSA Issuer is NOT the 'OpenBanking Ltd'.

Besides above two configured providers, we have a mock key provider (for which, no configuration is needed in the DB):

3. MOCK_KEY_PROVIDER - "com.ofss.digx.oauth2.service.DBBasedKeyProvider" – this is only a dummy DB based key provider. If none of the above two providers are configured in the DB, KeyProviderFactory would return the mock key provider. It stores only single public-private key pair in the DB itself and uses the same pair for all the TPP payload verifications.

Below is a sample code snippet to get the key provider for reference:

```

IKeyProvider keyProvider = KeyProviderFactory.getInstance().getProvider(issuer);
Map<String, String> publicKeyMap = keyProvider.fetchPublicKey(dcr_request_token);

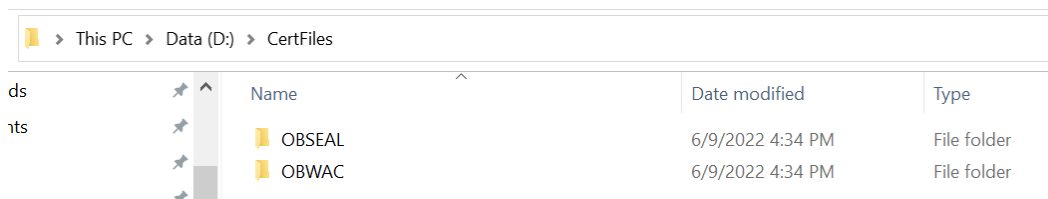
```

9. Keystore and Certificate for UK Open Banking Directory

This section describes the steps to generate the 'jks' files and configure the same in OBAPI for Open Banking Directory integration.

Steps to create 'identity' & 'trust' JKS files

1. Create two different folders OBWAC and OBSEAL and perform the below steps in the respective folders.



2. One should have the bank's OBWAC and OBSEAL configuration files(.cnf) to proceed further. Copy the files in the respective folders created above.
3. To generate CSR and key files for OBWAC and OBSEAL certificate with the help of .cnf file, execute below openssl commands

OBWAC> **openssl req -new -config obwac.cnf -out obwac.csr -keyout obwac.key**

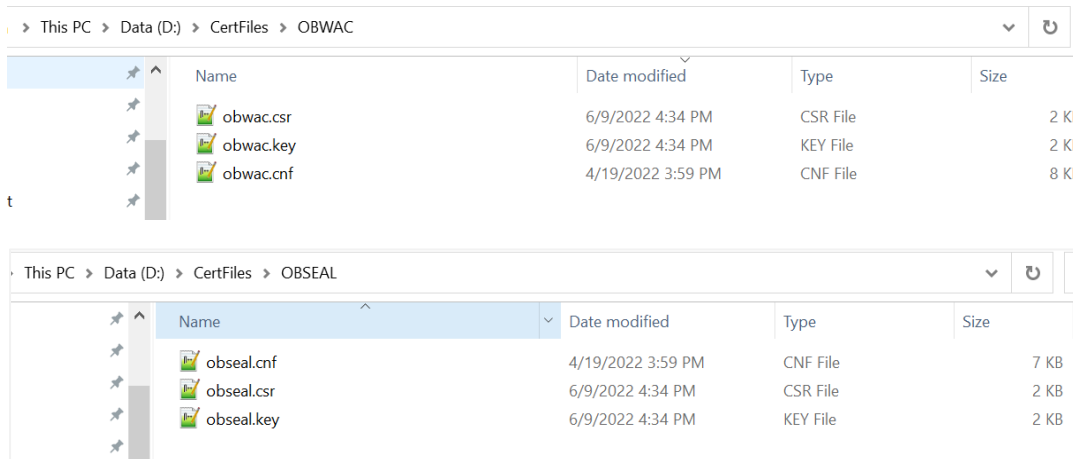
OBSEAL> **openssl req -new -config obseal.cnf -out obseal.csr -keyout obseal.key**

Note: enter the same pass phrase(pass1234 for example) for both obwac and obseal and make a note of it.

```
C:\Windows\System32\cmd.exe
D:\CertFiles\OBWAC>openssl req -new -config obwac.cnf -out obwac.csr -keyout obwac.key
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'obwac.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
D:\CertFiles\OBWAC>
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.
D:\CertFiles\OBSEAL>openssl req -new -config obseal.cnf -out obseal.csr -keyout obseal.key
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'obseal.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
D:\CertFiles\OBSEAL>
```

.csr and .key files have been generated with the above commands



4. Upload the above generated .csr files in Open Banking Directory Account to get OBWAC and OBSEAL pem files.

Let's assume, below output on uploading .csr files in the OB directory account
 Your OB WAC certificate xT-9_jWfAME1feTKZGaf8Dd_x1s was successfully created
 Your OB Seal certificate l6cfLYUSt91fOw13kdO0HYdIVTc was successfully created

Below are the steps to generate the OB WAC and OB Seal certificates in the Open Banking Directory Account (**Note: below screenshots are from the Sandbox account, kindly use Production Open Banking Directory Account details for the production setup**):

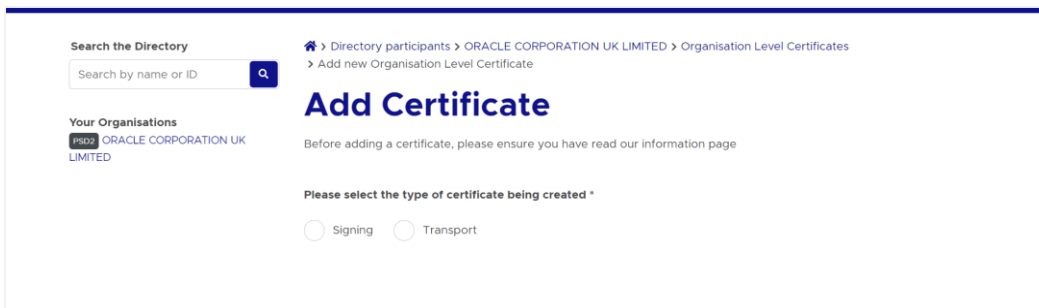
- a. Login with Open Banking Directory account credentials and select the desired Directory Participant(Your Organization).



- b. Go to 'Certificates' tab

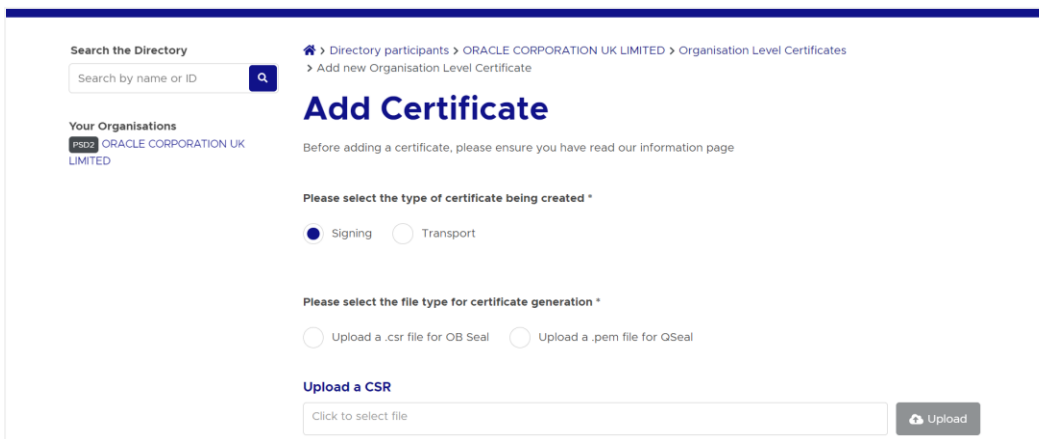


c. Click on 'Add new Organisation Certificate' button



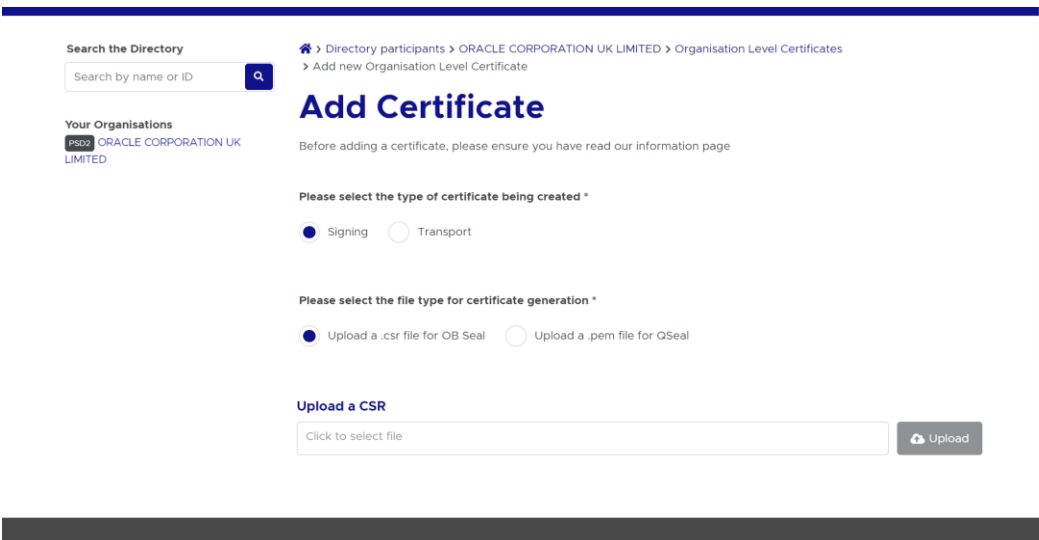
The screenshot shows the 'Add Certificate' page. At the top left, there is a search bar labeled 'Search the Directory' with the text 'Search by name or ID' and a magnifying glass icon. Below it, under 'Your Organisations', there is a list item for 'PSD2 ORACLE CORPORATION UK LIMITED'. The main heading is 'Add Certificate' in large blue font. Below the heading, there is a breadcrumb trail: 'Directory participants > ORACLE CORPORATION UK LIMITED > Organisation Level Certificates > Add new Organisation Level Certificate'. A sub-heading reads 'Add new Organisation Level Certificate'. Below this, a message says 'Before adding a certificate, please ensure you have read our information page'. The next section is 'Please select the type of certificate being created *' with two radio buttons: 'Signing' and 'Transport'. The 'Add new Organisation Certificate' button is highlighted in blue.

d. Select 'Signing' radio button to upload OB Seal .crs file

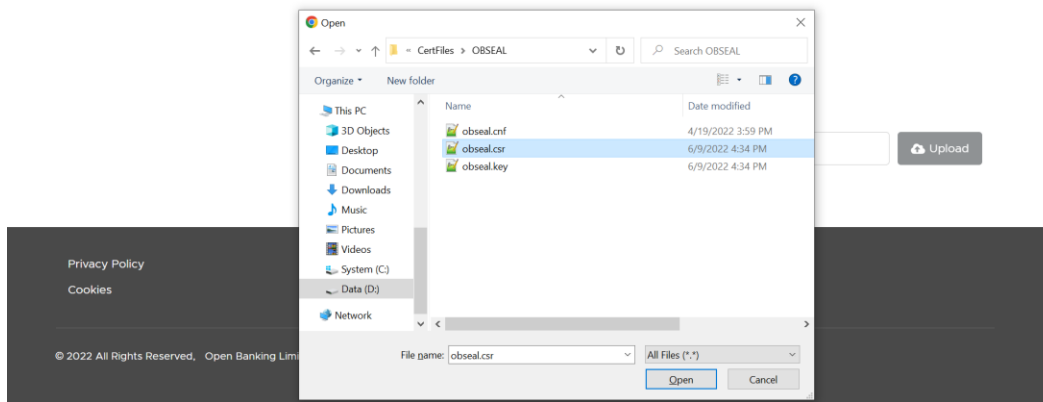


This screenshot is identical to the previous one, but the 'Signing' radio button is now selected. Below the radio buttons, there is a section 'Please select the file type for certificate generation *' with two radio buttons: 'Upload a .csr file for OB Seal' and 'Upload a .pem file for QSeal'. Below this, there is a section 'Upload a CSR' with a text input field containing 'Click to select file' and an 'Upload' button.

e. Select and upload the OB Seal .csr file



This screenshot is identical to the previous one, but the 'Upload a CSR' section is highlighted in blue. The text input field contains 'Click to select file' and the 'Upload' button is visible.

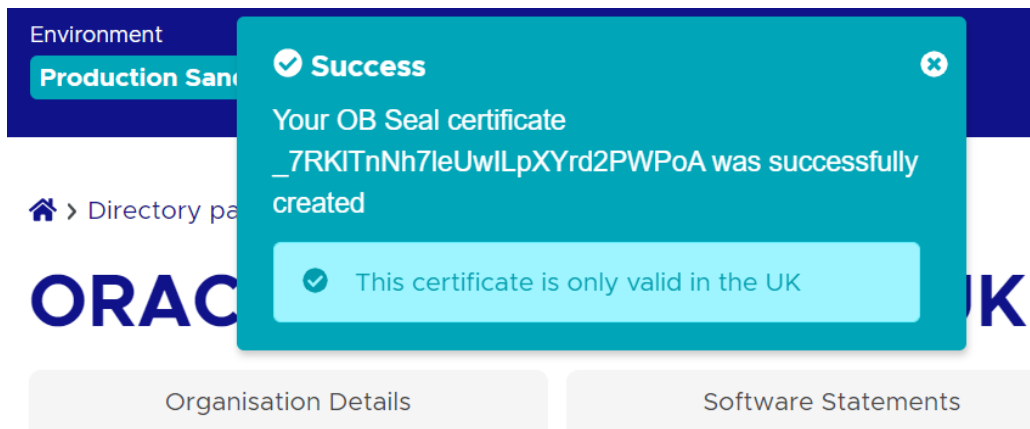


Upload a CSR

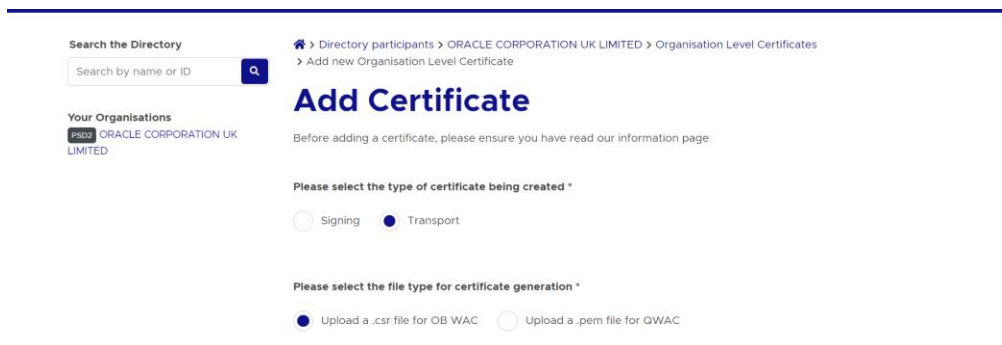
File: obseal.csr

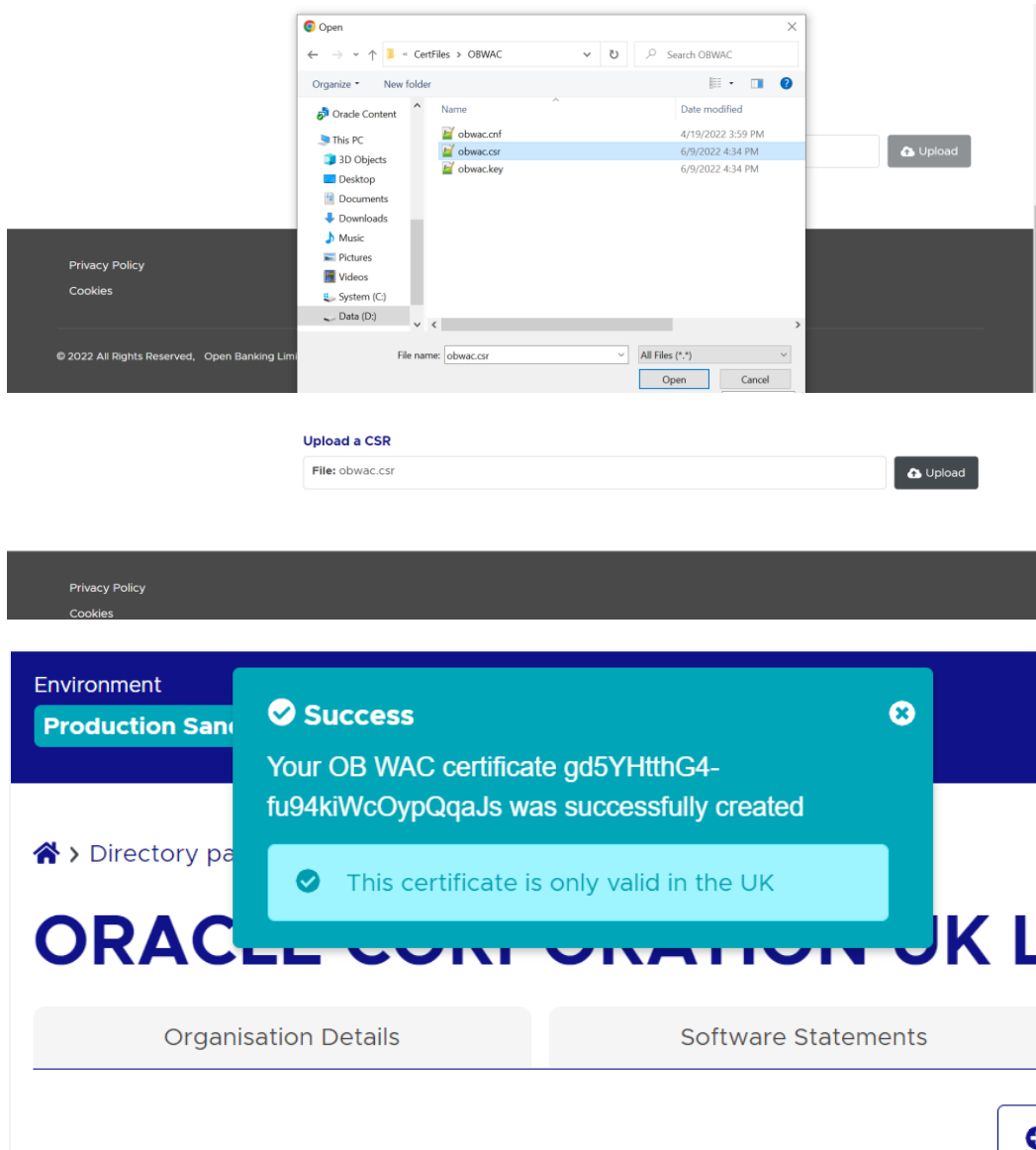


f. Clicking on the 'Upload' button will upload and display success popup



g. Repeat the above steps for OB WAC certificate generation. Select the 'Transport' radio button for OB WAC.





- h. Generated certificates would be visible on the certificates listing page. Certificate .pem files can be downloaded with the help of 'Get PEM' button displayed next to the respective certificates

Search the Directory

Directory participants > ORACLE CORPORATION UK LIMITED

ORACLE CORPORATION UK LIMITED

Your Organisations

PSB2 ORACLE CORPORATION UK LIMITED

Organisation Details Software Statements **Certificates**

[Add new Organisation Certificate](#)

Organisation level certificates ⓘ

⚠ Since 01 Jan 2021, EU and UK TPPs have different restrictions applied when minting/uploading certain certificates in the ecosystem.

Transport certificates

Type	Key ID	Valid from	
OB WAC	ORXBzNnuteEe-AVILlogNSM-PRQ	24/01/2022	View JWKS

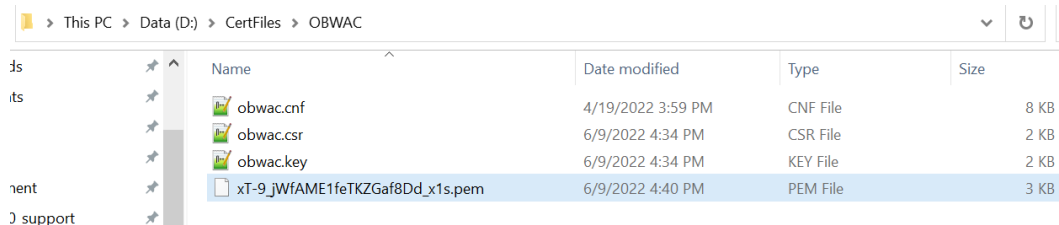
Transport certificates

Type	Key ID	Valid from	
OB WAC	ORXBzNnuteEe-AVILlogNSM-PRQ	24/01/2022	View JWKS Get PEM Revoke
	Associated software statements	Valid to	
	VT7llkzXFbaGbat9NSOIRH	24/02/2023	
OB WAC	xT-9_JWfAMEIteTKZGa8Dd_xts	25/04/2022	View JWKS Get PEM Revoke
	Associated software statements	Valid to	
	NoddlrPsQP0Myxb463YZG6SiroCGEGnwlPYZZTRFJ4zvJl2B3bZu16uCMFEkjhVgFEyVAoEYFM6L2J8tWfsfEZNlnivBZtdBB8nCbUvXcYks24RC0cVFVscOBsbwrsI E3R6bn5kxCupBISStwANeQ MokrxE33Vt4ssoBo9foyxGXGRSh55CMZSQblnxDzNrb	25/05/2023	
OB WAC	gd5YrHthG4-fu94kiWcOypQaaJs	29/06/2022	View JWKS Get PEM Revoke
	Associated software statements	Valid to	
	<i>This certificate is not associated with any software statements</i>	29/07/2023	

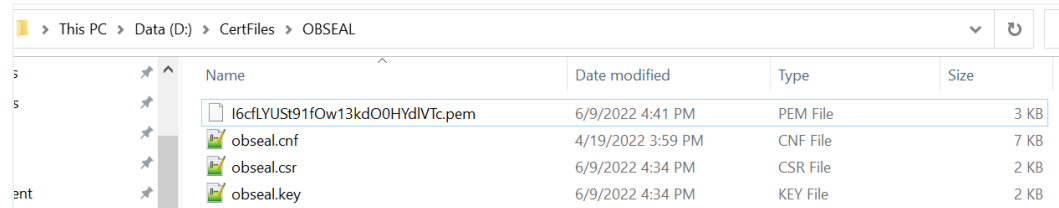
Signing certificates

Type	Key ID	Valid from	
OB Seal	BDHknaKeeNnP_XHjSwJLJu4IXs	24/01/2022	View JWKS Get PEM Revoke
	Associated software statements	Valid to	
	VT7llkzXFbaGbat9NSOIRH	24/02/2023	
OB Seal	l6cL_YUst9fOw13kdOOHYdIVTc	25/04/2022	View JWKS Get PEM Revoke
	Associated software statements	Valid to	
	NoddlrPsQP0Myxb463YZG6SiroCGEGnwlPYZZTRFJ4zvJl2B3bZu16uCMFEkjhVgFEyVAoEYFM6L2J8tWfsfEZNlnivBZtdBB8nCbUvXcYks24RC0cVFVscOBsbwrsI E3R6bn5kxCupBISStwANeQ MokrxE33Vt4ssoBo9foyxGXGRSh55CMZSQblnxDzNrb	25/05/2023	
OB Seal	_7RRkTnNh7eUwLpXYrd2PWPoA	29/06/2022	View JWKS Get PEM Revoke
	Associated software statements	Valid to	
	<i>This certificate is not associated with any software statements</i>	29/07/2023	

- Download the generated OBWAC and OBSEAL files and copy in the respective folders which have created locally. Change the extension from '.cer' to '.pem' of the downloaded files if required.



Name	Date modified	Type	Size
obwac.cnf	4/19/2022 3:59 PM	CNF File	8 KB
obwac.csr	6/9/2022 4:34 PM	CSR File	2 KB
obwac.key	6/9/2022 4:34 PM	KEY File	2 KB
xT-9_jWfAME1feTKZGaf8Dd_x1s.pem	6/9/2022 4:40 PM	PEM File	3 KB



Name	Date modified	Type	Size
l6cfLYUSt91fOw13kdO0HYdIVTc.pem	6/9/2022 4:41 PM	PEM File	3 KB
obseal.cnf	4/19/2022 3:59 PM	CNF File	7 KB
obseal.csr	6/9/2022 4:34 PM	CSR File	2 KB
obseal.key	6/9/2022 4:34 PM	KEY File	2 KB

6. Generate decrypted keys by executing below commands
OBWAC> openssl rsa -in obwac.key -out obwac_dec.key
OBSEAL> openssl rsa -in obseal.key -out obseal_dec.key

Enter the pass phrase 'pass1234' when provided, which had been entered at the time of the .key files.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

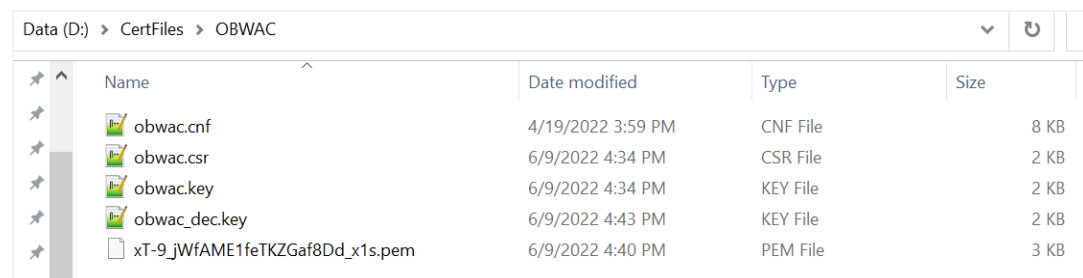
D:\CertFiles\OBWAC>openssl rsa -in obwac.key -out obwac_dec.key
Enter pass phrase for obwac.key:
writing RSA key

D:\CertFiles\OBWAC>
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

D:\CertFiles\OBSEAL>openssl rsa -in obseal.key -out obseal_dec.key
Enter pass phrase for obseal.key:
writing RSA key

D:\CertFiles\OBSEAL>
```



Name	Date modified	Type	Size
obwac.cnf	4/19/2022 3:59 PM	CNF File	8 KB
obwac.csr	6/9/2022 4:34 PM	CSR File	2 KB
obwac.key	6/9/2022 4:34 PM	KEY File	2 KB
obwac_dec.key	6/9/2022 4:43 PM	KEY File	2 KB
xT-9_jWfAME1feTKZGaf8Dd_x1s.pem	6/9/2022 4:40 PM	PEM File	3 KB

Name	Date modified	Type	Size
I6cflYUSt91fOw13kdO0HYdIVTc.pem	6/9/2022 4:41 PM	PEM File	3 KB
obseal.cnf	4/19/2022 3:59 PM	CNF File	7 KB
obseal.csr	6/9/2022 4:34 PM	CSR File	2 KB
obseal.key	6/9/2022 4:34 PM	KEY File	2 KB
obseal_dec.key	6/9/2022 4:44 PM	KEY File	2 KB

7. Download OB Root and Issuing Certificates from the Open Banking directory
 - a. URL for sandbox certificates:
<https://openbanking.atlassian.net/wiki/spaces/DZ/pages/252018873/OB+Root+and+Issuing+Certificates+for+Sandbox>
 - b. URL for production certificates:
<https://openbanking.atlassian.net/wiki/spaces/DZ/pages/80544075/OB+Root+and+Issuing+Certificates+for+Production>
8. Create a copy of both the downloaded certificate files and change the extension from .cer to .pem and copy in the OBWAC folder. Keep the file names same
 - a. OB_SandBox_PP_IssuingCA.cer to OB_SandBox_PP_IssuingCA.pem
 - b. OB_SandBox_PP_RootCA.cer to OB_SandBox_PP_RootCA.pem

Note: remove the spaces from the pem file names if there are any.

Name	Date modified	Type	Size
OB_SandBox_PP_IssuingCA.cer	6/9/2022 4:48 PM	Security Certificate	2 KB
OB_SandBox_PP_IssuingCA.pem	6/9/2022 4:48 PM	PEM File	2 KB
OB_SandBox_PP_RootCA.cer	6/9/2022 4:48 PM	Security Certificate	2 KB
OB_SandBox_PP_RootCA.pem	6/9/2022 4:48 PM	PEM File	2 KB
obwac_dec.key	6/9/2022 4:43 PM	KEY File	2 KB
xT-9_jWfAME1feTKZGaf8Dd_x1s.pem	6/9/2022 4:40 PM	PEM File	3 KB
obwac.csr	6/9/2022 4:34 PM	CSR File	2 KB
obwac.key	6/9/2022 4:34 PM	KEY File	2 KB
obwac.cnf	4/19/2022 3:59 PM	CNF File	8 KB

9. Use 'cat' command on linux or 'type' command in Windows machine to build the certificate chain from the above three .pem files
 - a. `cat xT-9_jWfAME1feTKZGaf8Dd_x1s.pem OB_SandBox_PP_IssuingCA.pem OB_SandBox_PP_RootCA.pem > chain.pem`
or
 - b. `type xT-9_jWfAME1feTKZGaf8Dd_x1s.pem OB_SandBox_PP_IssuingCA.pem OB_SandBox_PP_RootCA.pem > chain.pem`

```

C:\Windows\System32\cmd.exe
D:\CertFiles\OBWAC>type xT-9_jWfAME1feTKZGaf8Dd_x1s.pem OB_SandBox_PP_IssuingCA.pem OB_SandBox_PP_RootCA.pem > chain.pem
xT-9_jWfAME1feTKZGaf8Dd_x1s.pem

OB_SandBox_PP_IssuingCA.pem

OB_SandBox_PP_RootCA.pem

D:\CertFiles\OBWAC>

```

Name	Date modified	Type	Size
chain.pem	6/9/2022 5:42 PM	PEM File	5 KB
OB_SandBox_PP_IssuingCA.cer	6/9/2022 4:48 PM	Security Certificate	2 KB
OB_SandBox_PP_IssuingCA.pem	6/9/2022 4:48 PM	PEM File	2 KB
OB_SandBox_PP_RootCA.cer	6/9/2022 4:48 PM	Security Certificate	2 KB
OB_SandBox_PP_RootCA.pem	6/9/2022 4:48 PM	PEM File	2 KB
obwac_dec.key	6/9/2022 4:43 PM	KEY File	2 KB
xT-9_jWfAME1feTKZGaf8Dd_x1s.pem	6/9/2022 4:40 PM	PEM File	3 KB
obwac.csr	6/9/2022 4:34 PM	CSR File	2 KB
obwac.key	6/9/2022 4:34 PM	KEY File	2 KB
obwac.cnf	4/19/2022 3:59 PM	CNF File	8 KB

10. Creating Custom Keystore and importing chain

WebLogic Server Java Utilities is used to create the custom keystore and importing private key & the certificates chains.

Resource URL for reference:

https://docs.oracle.com/cd/E13222_01/wls/docs81/admin_ref/utlils20.html

Execute the below command with files in the OBWAC directory

```
java -cp /home/devops/Oracle/Middleware/Oracle_Home/wlserver/server/lib/weblogic.jar
utils.ImportPrivateKey -certfile chain.pem -keyfile obwac_dec.key -keystore
openbanking_custom_identity.jks -storepass pass1234 -alias openbanking_obtrans
```

Note: “/home/devops/Oracle/Middleware/Oracle_Home/wlserver/server/lib/” this path is to locate the weblogic.jar file, this may differ as per the setup.

```
[devops@obdxwls OBWAC]$ ls -l
total 40
-rwxrwxrwx 1 54323 54323 1559 Jun  9 16:48 OB_SandBox_PP_IssuingCA.pem
-rwxrwxrwx 1 54323 54323 1380 Jun  9 16:48 OB_SandBox_PP_RootCA.pem
-rwxrwxrwx 1 54323 54323 5048 Jun  9 17:42 chain.pem
-rwxrwxrwx 1 54323 54323 7712 Apr 19 15:59 obwac.cnf
-rwxrwxrwx 1 54323 54323 1392 Jun  9 16:34 obwac.csr
-rwxrwxrwx 1 54323 54323 1884 Jun  9 16:34 obwac.key
-rwxrwxrwx 1 54323 54323 1706 Jun  9 16:43 obwac_dec.key
-rwxrwxrwx 1 54323 54323 2109 Jun  9 16:40 xT-9_jWfAME1feTKZGaf8Dd_x1s.pem
[devops@obdxwls OBWAC]$ java -cp /home/devops/Oracle/Middleware/Oracle_Home/wlserver/server/lib/weblogic.jar
utils.ImportPrivateKey -certfile chain.pem -keyfile obwac_dec.key -keystore openbanking_custom_identity.jks
-storepass pass1234 -alias openbanking_obtrans
No password was specified for the key entry
Keystore password will be used.
<Jun 9, 2022 6:23:49 PM IST> <Info> <Security> <BEA-090905> <Disabling the CryptoJ JCE Provider self-integrity
check for better startup performance. To enable this check, specify -Dweblogic.security.allowCryptoJDefaultJCEVerification=true.>
<Jun 9, 2022 6:23:49 PM IST> <Info> <Security> <BEA-090906> <Changing the default Random Number Generator in
RSA CryptoJ from ECDRBG128 to HMACDRBG. To disable this change, specify -Dweblogic.security.allowCryptoJDefaultFRNG=true.>
Imported private key obwac_dec.key and certificate chain.pem
into a new keystore openbanking_custom_identity.jks of type jks under alias openbanking_obtrans
[devops@obdxwls OBWAC]$
```

A new .jks file with the filename ‘openbanking_custom_identity.jks’ is created.

Name	Size	Changed	Rights	Owner
openbanking_custom_identity.jks	3 KB	6/9/2022 6:02:42 PM	rwxrwxrwx	obdxdevops
chain.pem	5 KB	6/9/2022 5:42:07 PM	rw-rw-r--	1000
OB_SandBox_PP_IssuingCA.pem	2 KB	6/9/2022 4:48:14 PM	rwxrwxrwx	obdxdevops
OB_SandBox_PP_RootCA.pem	2 KB	6/9/2022 4:48:08 PM	rwxrwxrwx	obdxdevops
obwac_dec.key	2 KB	6/9/2022 4:43:12 PM	rwxrwxrwx	obdxdevops
xT-9_jWfAME1feTKZGaf8Dd_x1s.pem	3 KB	6/9/2022 4:40:39 PM	rwxrwxrwx	obdxdevops
obwac.key	2 KB	6/9/2022 4:34:17 PM	rwxrwxrwx	obdxdevops
obwac.csr	2 KB	6/9/2022 4:34:17 PM	rwxrwxrwx	obdxdevops
obwac.cnf	8 KB	4/19/2022 3:59:26 PM	rwxrwxrwx	obdxdevops

11. Creating Custom Identity Trust

Execute below two commands.

Enter 'yes' and press enter when prompted "Trust this certificate? [no]:".

Note: "/home/devops/jdk18/bin/" this path is to locate the java keytool, this may differ as per the setup.

1) /home/devops/jdk18/bin/keytool -keystore openbanking_custom_identity.jks -importcert -file OB_SandBox_PP_RootCA.cer -alias openbanking_rootca -storepass pass1234

```
@obdswls/scratch/obdx/OpenBanking/CertFiles/OBWAC
[devops@obdswls OBWAC]$ ls -l
total 52
-rw-rw-r-- 1 54323 54323 1559 Jun  9 16:48 OB_SandBox_PP_IssuingCA.cer
-rwxrwxrwx 1 54323 54323 1559 Jun  9 16:48 OB_SandBox_PP_IssuingCA.pem
-rw-rw-r-- 1 54323 54323 1380 Jun  9 16:48 OB_SandBox_PP_RootCA.cer
-rwxrwxrwx 1 54323 54323 1380 Jun  9 16:48 OB_SandBox_PP_RootCA.pem
-rwxrwxrwx 1 54323 54323 5048 Jun  9 17:42 chain.pem
-rwxrwxrwx 1 54323 54323 7712 Apr 19 18:59 obwac.cnf
-rwxrwxrwx 1 54323 54323 1392 Jun  9 16:34 obwac.csr
-rwxrwxrwx 1 54323 54323 1884 Jun  9 16:34 obwac.key
-rwxrwxrwx 1 54323 54323 1706 Jun  9 16:43 obwac_dec.key
-rw-rw-r-- 1 devops devops 2884 Jun  9 18:23 openbanking_custom_identity.jks
-rwxrwxrwx 1 54323 54323 2109 Jun  9 16:40 xt-9_jwFAME1feTK2oaE9Dd_xls.pem
[devops@obdswls OBWAC]$ /home/devops/jdk18/bin/keytool -keystore openbanking_custom_identity.jks -importcert -file OB_SandBox_PP_RootCA.cer -alias openbankin
g_rootca -storepass pass1234
Owner: CN=OpenBanking Pre-Production Root CA, O=OpenBanking, C=GB
Issuer: CN=OpenBanking Pre-Production Root CA, O=OpenBanking, C=GB
Serial number: 58cf6f6
Valid from: Fri Sep 22 17:09:42 IST 2017 until: Tue Sep 22 17:39:42 IST 2037
Certificate fingerprints:
    MD5: 38:BC:2F:F0:7F:34:A0:E0:42:DB:65:81:51:F9:6C:D7
    SHA1: 3c:9f:Ad:3f:c3:9b:21:EF:00:F3:39:93:90:61:6C:8A:7D:0D:5F:03
    SHA256: 73:24:4E:0D:1F:5B:01:C5:F6:E5:A1:A0:2A:18:AC:16:71:01:4F:2C:AF:A3:0A:53:52:87:FE:37:A3:70:74:2F
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 4096-bit RSA key
Version: 3
Extensions:
#1: ObjectID: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen:2147483647
]
#2: ObjectID: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  Crl_Sign
]
#3: ObjectID: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
  KeyIdentifier [
    0000: EC 38 8E 0B DA F3 F9 37 3E 90 DE 7D 5F 6A E6 60 .8.....7>...j.
    0010: CD 79 42 83 .y8.
  ]
]
Trust this certificate? [no]: yes
Certificate was added to keystore

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool -importkeystore -srcke
ystore openbanking_custom_identity.jks -destkeystore openbanking_custom_identity.jks -deststoretype pkcs12".
[devops@obdswls OBWAC]$
```

2) /home/devops/jdk18/bin/keytool -keystore openbanking_custom_trust.jks -importcert -file OB_SandBox_PP_IssuingCA.cer -alias openbanking_issueca -storepass pass1234


```
@obdxwls/scratch/obdx/OpenBanking/CertFiles/OBWAC
[DistributionPoint:
  [URIName: http://ob.trustis.com/ob_pp_rootca.crl]
]]
#5: ObjectID: 2.5.29.32 Criticality=false
CertificatePolicies [
  [CertificatePolicyId: [1.3.6.1.4.1.5237.134.1.100]
  [PolicyQualifierInfo: [
    [PolicyQualifierID: 1.3.6.1.5.7.2.1
    qualifier: 0000: 16 1E 68 74 74 70 3A 2F 2F 6F 62 2E 74 72 75 73 ..http://ob.trus
0010: 74 69 73 2E 63 6F 6D 2F 70 6F 6C 69 63 69 65 73 tis.com/policies
  ]
  ], PolicyQualifierInfo: [
    [PolicyQualifierID: 1.3.6.1.5.7.2.2
    qualifier: 0000: 30 81 86 0C 81 83 55 73 65 20 6F 66 20 74 68 69 0.....Use of thi
0010: 73 20 43 65 72 74 69 66 69 63 61 74 65 20 63 6F s Certificate co
0020: 6E 73 74 69 74 75 74 65 73 20 61 63 63 65 70 74 nstitutes accept
0030: 61 6E 63 65 20 6F 66 20 74 68 65 20 4F 70 65 6E ance of the Open
0040: 42 61 6E 68 69 6E 67 20 52 6F 6F 74 20 43 41 20 Banking Root CA
0050: 43 65 72 74 69 66 69 63 61 74 69 6F 6E 20 50 6F Certification Po
0060: 6C 69 63 69 65 73 20 61 6E 64 20 43 65 72 74 69 lishes and Certi
0070: 66 69 63 61 74 65 20 50 72 61 63 74 69 63 65 20 ficate Practice
0080: 53 74 61 74 65 6D 65 6E 74 Statement
  ]
]] ]
#6: ObjectID: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  Crl_Sign
]
#7: ObjectID: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
  KeyIdentifier [
    0000: 50 73 91 C6 21 72 D3 77 F4 FE 00 12 06 81 5C 79 Pa..lr.w.....\y
0010: 79 6E 3F 50 yn??
  ]
]
Trust this certificate? [no]: yes
Certificate was added to keystore
[devops@obdxwls OBWAC]$
```

Another .jks file with filename 'openbanking_custom_trust.jks' is created.

```
@obdxwls/scratch/obdx/OpenBanking/CertFiles/OBWAC
[devops@obdxwls OBWAC]$ ls -l
total 60
-rw-rw-r-- 1 54323 54323 1559 Jun 9 16:48 OB_SandBox_PP_IssuingCA.cer
-rwxrwxrwx 1 54323 54323 1559 Jun 9 16:48 OB_SandBox_PP_IssuingCA.pem
-rw-rw-r-- 1 54323 54323 1380 Jun 9 16:48 OB_SandBox_PP_RootCA.cer
-rwxrwxrwx 1 54323 54323 1380 Jun 9 16:48 OB_SandBox_PP_RootCA.pem
-rwxrwxrwx 1 54323 54323 5048 Jun 9 17:42 chain.pem
-rwxrwxrwx 1 54323 54323 7012 Apr 19 15:59 obwac.cnf
-rwxrwxrwx 1 54323 54323 1392 Jun 9 16:34 obwac.csr
-rwxrwxrwx 1 54323 54323 1884 Jun 9 16:34 obwac.key
-rwxrwxrwx 1 54323 54323 1706 Jun 9 16:43 obwac_dec.key
-rw-rw-r-- 1 devops devops 4307 Jun 9 19:43 openbanking_custom_identity.jks
-rw-rw-r-- 1 devops devops 1635 Jun 9 19:47 openbanking_custom_trust.jks
-rwxrwxrwx 1 54323 54323 2109 Jun 9 16:40 xT-9_jWfAME1feTKZGaf8Dd_x1s.pem
[devops@obdxwls OBWAC]$
```

Name	Size	Changed	Rights	Owner
..		6/9/2022 6:02:42 PM	rwxrwxrwx	obdxdevops
openbanking_custom_trust.jks	2 KB	6/9/2022 6:47:47 PM	rw-rw-r--	1000
openbanking_custom_identity.jks	5 KB	6/9/2022 6:43:54 PM	rw-rw-r--	1000
chain.pem	5 KB	6/9/2022 5:42:07 PM	rwxrwxrwx	obdxdevops
OB_SandBox_PP_IssuingCA.pem	2 KB	6/9/2022 4:48:14 PM	rwxrwxrwx	obdxdevops
OB_SandBox_PP_IssuingCA.cer	2 KB	6/9/2022 4:48:14 PM	rw-rw-r--	obdxdevops
OB_SandBox_PP_RootCA.pem	2 KB	6/9/2022 4:48:08 PM	rwxrwxrwx	obdxdevops
OB_SandBox_PP_RootCA.cer	2 KB	6/9/2022 4:48:08 PM	rw-rw-r--	obdxdevops
obwac_dec.key	2 KB	6/9/2022 4:43:12 PM	rwxrwxrwx	obdxdevops
xT-9_jWfAME1feTKZGaf8Dd_x1s.pem	3 KB	6/9/2022 4:40:39 PM	rwxrwxrwx	obdxdevops
obwac.key	2 KB	6/9/2022 4:34:17 PM	rwxrwxrwx	obdxdevops
obwac.csr	2 KB	6/9/2022 4:34:17 PM	rwxrwxrwx	obdxdevops
obwac.cnf	8 KB	4/19/2022 3:59:26 PM	rwxrwxrwx	obdxdevops

Note: OpenSSL 1.1.1n 15 Mar 2022 is used to perform above steps.

```
C:\Windows\System32\cmd.exe  
D:\CertFiles\OBWAC>openssl version  
OpenSSL 1.1.1n 15 Mar 2022  
D:\CertFiles\OBWAC>
```

[Home](#)