

Oracle Banking APIs

OpenID Guide

Release 19.2.0.0.0

Part No. F26907-01

December 2019

ORACLE®

OpenID Guide
December 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

www.oracle.com/financialservices/

Copyright © 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 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 | 4 |
| 1.1 Intended Audience | 4 |
| 1.2 Documentation Accessibility | 4 |
| 1.3 Access to Oracle Support | 4 |
| 1.4 Structure | 4 |
| 1.5 Related Information Sources..... | 4 |
| 2. OPENID | 5 |
| 2.1 openid-discovery-endpoint.properties | 5 |
| 2.1.1 dynamic-client-registration.properties | 7 |
| 2.2 userinfo.properties | 7 |
| 3. MESSAGE SIGNING AND VALIDATION | 9 |
| 3.1 Authorization Server..... | 9 |
| 3.1.1 common.properties..... | 9 |
| 3.2 Resource Server | 10 |
| 4. HANDLERS | 13 |
| 4.1 Authorization Server..... | 13 |
| 4.2 Resource Server | 13 |

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=accandid=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=accandid=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=accandid=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:

- Purpose
- Configuration / Installation.

1.5 Related Information Sources

For more information on Oracle Banking APIs Release 19.2.0.0.0, refer to the following documents:

- Oracle Banking APIs Installation Manuals

2. OPENID

OpenID Connect is a simple identity layer on top of the OAuth 2.0 protocol. It enables Clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User in an interoperable and REST-like manner.

OBAPI has configurations which when altered will affect the behavior of OpenID in various ways. These configurations are composed into the following properties files :

2.1 openid-discovery-endpoint.properties

This properties file contains the information about the URLs and certain parameters supported by ASPSP that needs to be displayed to the TPP when requested. The information is displayed through discovery endpoint.

| Parameter | Description | Values |
|------------------------|---|---|
| issuer | This parameter represents Issuer's endpoint. | * {{ISSUER'S_URL}} Example: https://server.example.com |
| authorization_endpoint | This parameter represents ASPSP's authorization endpoint. | * {{AUTHORIZATION_ENDPOINT_URL}} Example: https://server.example.com/connect/authorize |
| token_endpoint | This parameter represents ASPSP's token endpoint. | * {{TOKEN_ENDPOINT_URL}} Example: https://server.example.com/connect/token |
| userinfo_endpoint | This parameter represents ASPSP's userinfo endpoint. | * {{USERINFO_ENDPOINT_URL}} Example: https://server.example.com/connect/userinfo |

| Parameter | Description | Values |
|---|--|---|
| jwt_uri | This parameter represents ASPSP's jwt uri. | * {{JWT_URI}} Example: https://server.example.com/jwt.json |
| registration_endpoint | This parameter represents ASPSP's Dynamic Client Registration endpoint. | * {{REGISTRATION_ENDPOINT_URL}} Example: https://server.example.com/connect/register |
| response_types_supported | This parameter represents ASPSP's supported response Types. | code,code token,code id_token,code token id_token |
| grant_types_supported | This parameter represents ASPSP's supported grant types. | AUTHORIZATION_CODE,PASSWORD,CLIENT_CREDENTIALS,REFRESH_TOKEN |
| subject_types_supported | This parameter represents ASPSP's supported subject type. | public |
| id_token_signing_alg_values_supported | This parameter represents ASPSP's supported id_token signing algorithm. | RS256,PS256 |
| request_object_signing_alg_values_supported | This parameter represents ASPSP's supported request object signing algorithm. | RS256,PS256 |
| token_endpoint_auth_methods_supported | This parameter represents ASPSP's supported token endpoint authentication methods. | client_secret_basic |
| identityDomain | This parameter represents the default configured Identity Domain. | * {{IDENTITY_DOMAIN_NAME}} Example: UKOPENBANKING |

2.1.1 dynamic-client-registration.properties

This properties file contains the parameters related to Dynamic Client Registration.

| Parameter | Description | Values |
|-----------------|---|---|
| client_Type | This parameter represents the default configured Client Type. | CONFIDENTIAL_CLIENT |
| resource_server | This parameter represents the default configured Resource Server. | * RESOURCE_SERVER_NAME } Example: AIPISP2 |

2.2 userinfo.properties

This properties file represents the mapping of OpenID claims to the corresponding claims available from user details in OBAPI. The parameter is the OpenID claim while it's value is the corresponding claim available from user details in OBAPI.

Any new parameter and its OBAPI counterpart can be configured by adding in this property file.

| Parameter | Description | Values |
|--------------|---|-------------|
| sub | This parameter represents Subject. | userName |
| name | This parameter represents User's name. | userName |
| given_name | This parameter represents User's given name. | firstName |
| family_name | This parameter represents User's family name. | lastName |
| middle_name | This parameter represents User's middle name. | middleName |
| email | This parameter represents User's email. | emailId |
| birthdate | This parameter represents User's date of birth. | dateOfBirth |
| phone_number | This parameter represents User's phone number. | phoneNumber |

| Parameter | Description | Values |
|-----------|---|---------|
| address | This parameter represents User's address. | address |

* – These values are a part of Day one configurations and are not factory shipped. These values are mandatory and if not provided will result in error.

3. MESSAGE SIGNING AND VALIDATION

OBAPI has message signing and validation configurations, which when altered will affect the response of Open Banking API's.

3.1 Authorization Server

The configurations are composed into the following properties files:

3.1.1 common.properties

| Parameter | Description | Values |
|--------------------|---|--|
| oauthHandlerConfig | <p>This parameter is responsible for choosing the required Handler. The Parameter's value is the fully qualified name of the Handler Class.</p> <p>The handler is responsible for implementing methods/validations that are over and above OpenID methods/validations. By default DefaultOauthHandler is used. It contains the methods to validate request Object Claims, fetch public key and private key, etc.</p> <p>UKOpenBankingHandler extends DefaultOauthHandler and overrides the methods to implement the UK OpenBanking specific validations.</p> <p>Any new Handler to be written for UK OpenBanking should extend UKOpenBankingHandler and override the methods and the fully qualified name of the Handler should be given against this oauthHandlerConfig parameter.</p> | <p>*</p> <p>{{FULLY_QUALIFIED_HANDLER_CLASS_NAME}}</p> <p>Example:</p> <p>com.ofss.digx.oauth2.handler.openbanking.uk.UKOpenBankingHandler</p> |

* – These values are a part of Day one configurations and are not factory shipped. These values are mandatory and if not provided will result in error.

3.2 Resource Server

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 |
|-------------|--------------------------------|---------|
|-------------|--------------------------------|---------|

| | | |
|----------------------------------|--------------|---|
| <p>MESSAGE_SIGNATURE_HANDLER</p> | <p>-----</p> | <p>This property is responsible for choosing the required Handler. The Parameter's value is the fully qualified name of the Handler Class.</p> <p>The handler is responsible for implementing methods/validations of OpenBanking. By default DefaultMessageSignatureHandler is used. It contains the methods to validate jwt token headers, fetch public key and private key, etc.</p> <p>Any new Handler to be written for UK OpenBanking should extend DefaultMessageSignatureHandler and override the methods and the fully qualified name of the Handler should be given against this property Id and committed in database.</p> <p>Example Query :</p> <pre>"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 ('MESSAGE_SIGNATURE_HANDLER','openBankingConfig','com.ofss.digx.appx.openbanking.uk.message.signature.handler.UKMessageSignatureHandler','N',null,'Message signature handler','ofssuser',sysdate,'ofssuser',sysdate,'A',1);"</pre> |
|----------------------------------|--------------|---|

| | | |
|--------------------------------|----------|---|
| <p>MESSAGE_ENCRYPTION_FLAG</p> | <p>Y</p> | <p>Flag to enable or disable the Message Signing and Validation.</p> <p>Set 'Y' to enable and 'N' to disable message signing and validations.</p> <p>Example Query :</p> <pre>"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 ('MESSAGE_ENCRYPTION_FLAG','openBankingConfig','Y','N',null,'Open Banking payload signing and validation flag','ofssuser',sysdate,'ofssuser',sysdate,'A',1);"</pre> |
|--------------------------------|----------|---|

4. HANDLERS

Handlers for OpenBanking provide extensibility. The following are the two sets of Handlers which can be utilised directly or can be extended to implement custom functionality.

4.1 Authorization Server

The handler on Authorization Server is responsible for implementing methods/validations that are over and above OpenID methods/validations.

- If no configuration is provided, DefaultOAuthHandler is used by default. It contains the methods to validate request Object Claims, fetch public key and private key, etc.
- UKOpenBankingHandler extends DefaultOAuthHandler and overrides the methods to implement the UK OpenBanking specific validations.

NOTE : Any new Handler to be written for UK OpenBanking should extend UKOpenBankingHandler and override the required methods. Also the fully qualified name of the Handler should be given against this oauthHandlerConfig parameter.

4.2 Resource Server

The handler on Resource Server is responsible for implementing methods/validations of OpenBanking.

- If no configuration is provided, DefaultMessageSignatureHandler is used by default. It contains the methods to validate jwt token headers, fetch public key and private key, etc.

NOTE :Any new Handler to be written for UK OpenBanking should extend DefaultMessageSignatureHandler and override the required methods. Also the fully qualified name of the Handler should be given against MESSAGE_SIGNATURE_HANDLER property Id and committed in database.
