Oracle® Utilities Opower Integration Hub

Standalone Application Single Sign-On Configuration Guide

Latest Release
G23537-04
April 2025

ORACLE

Oracle Utilities Opower Integration Hub Standalone Application Single Sign-On Configuration Guide, Latest Release

G23537-04

Contents

1

Configuring SSO for Third-Party Standalone Applications

Welcome to the Oracle Utilities Opower Configuring SSO for Third-Party Partner Standalone Applications Guide, which describes how to for implement single sign-on (SSO) with the Oracle Utilities Opower and third-party applications.

Oracle Utilities Opower follows OAuth2 and the OpenID Connect specification to integrate with third-party partners. A partner must obtain an access token that allows retrieving user data from Oracle Utilities Opower GraphQL APIs.

Prerequisites

The following prerequisites are required prior to configuring SSO for third-party standalone applications:

- **Configuration Data**: Oracle Utilities provides the third-party partner with various configuration data. For more information on these details, refer to Configuration Data.
- **Third-Party Partner Enrollment**: Third-party partners must be members of the Oracle Partner Network. For information on this program and steps to join the Oracle Partner Network, refer to https://www.oracle.com/partnernetwork/program/.

Configuration Data

Configuration data is required to be shared between Oracle Utilities Opower and the third-party partner:

- **Information that the third-party partner provides to Oracle Utilities Opower**: Prior to Oracle Utilities Opower setting up integration, the third-party partner must provide the **`redirect_uri`** for all applicable utilities and tiers relevant to the third-party partner. For a given utility and tier, this value is the URL that Oracle Utilities Identity and Access Management uses to return the authorization code. Identity and Access Management supports only a single URL, which must exactly match the one provided to the `/authorize` endpoint when the authorization code flow is initiated. For example, a `redirect_uri` would be of the format `https://examplepartner.net/authenticate/redirect`.
- **Information that Oracle Utilities Opower provides to third-party partners for every client and tier**:
  - **List of endpoints**:
    * The utility-specific OpenId Connect Discovery endpoint for the stage tier. Endpoints are of the format `https://idcs-[INSTANCE_ID_Stage].oraclecloud.com/.well-known/openid-configuration`.
    * The utility-specific OpenId Connect Discovery endpoint for the production tier. Endpoints are of the format `https://idcs-`

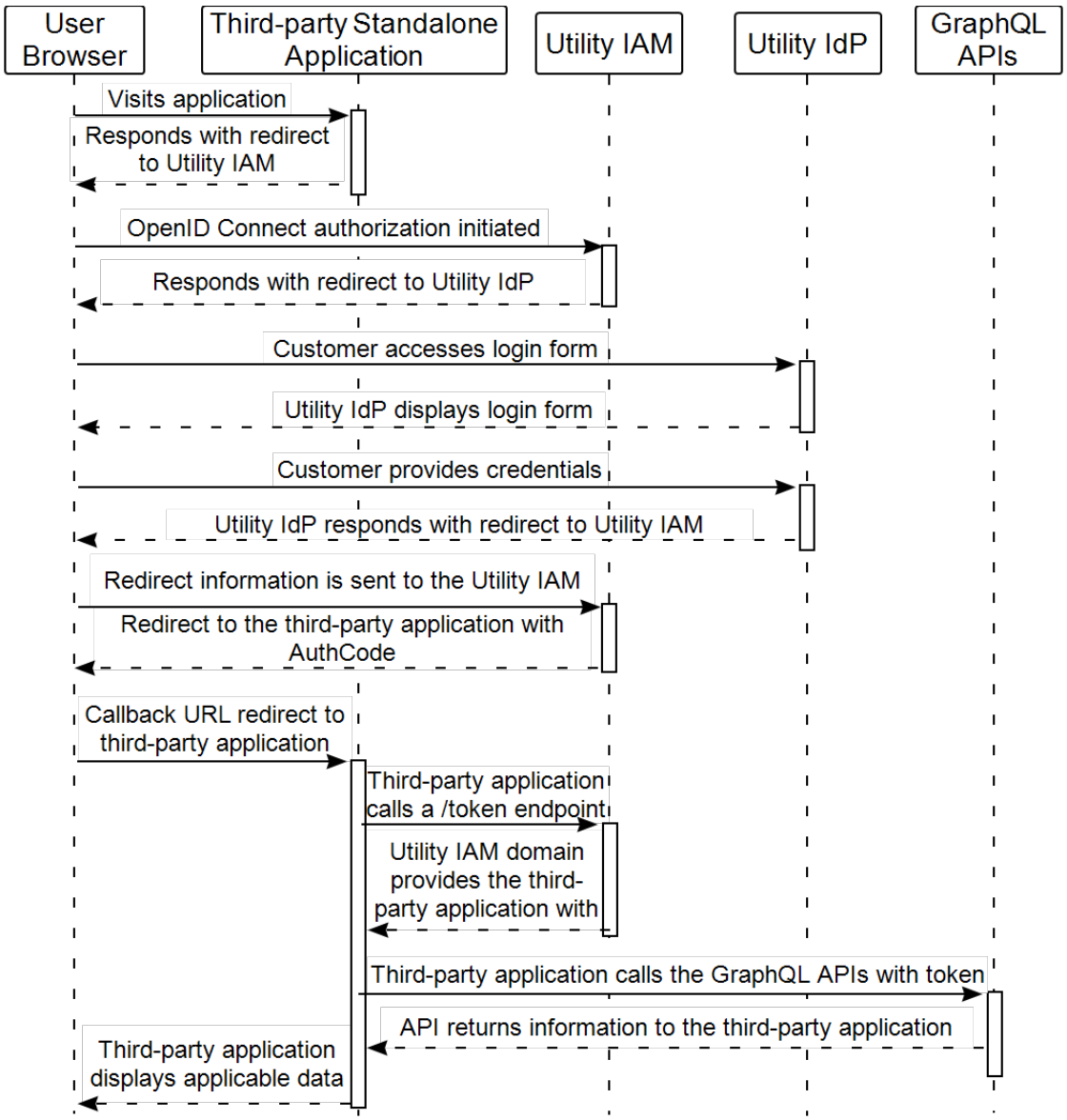
`[INSTANCE_ID_Production].oraclecloud.com/.well-known/openid-configuration`.

- **Client credentials for a given utility and tier**:
  - client_id
  - client secret
- **Scope management**: The list of scopes that are allowed for the given utility and tier.

Authorization Code Flow

For integration, OpenID Connect Authorization Code Flow is used. More details on implementing Authorization Code Flow with Identity and Access Management and request examples can be found in the Identity and Access Management documentation.

A sequence diagram of the entire flow for a successful user authorization is provided below.



1. A customer accesses the website of the third-party standalone application. Since the user is not logged in, the third-party website responds with a redirect to the Utility Identity and Access Management `[IDCS_IAM_URL]/oauth2/v1/authorize` endpoint. This endpoint requires a `redirect_uri` parameter containing a callback URL, which defines where to return the user after successful authentication.
2. The Utility Identity and Access Management instance responds with a redirect to the Utility Identity Provider (IdP) to begin the authentication process according to SAML or OpenId protocol.
3. The Utility IdP displays the login form to the user.
4. The user provides credentials in the login form and submits it to the Utility IdP. Upon successful authentication on the Utility IdP side, the user is redirected to the Utility IAM providing authentication assertion according to SAML or OpenId protocols.
5. User get logged in to the Utility Identity and Access Management instance and then it redirects back to the third-party application callback URL.
6. The user visits the third-party application callback URL with `AuthCode` provided as a query parameter.
7. The third-party application makes a call to `[IDCS_IAM_URL]/oauth2/v1/token` endpoint of the Utility Identity and Access Management instance providing the `AuthCode` and client credentials. The Utility Identity and Access Management instance responds back with an ID token, an access token, and optionally a refresh token.
8. The third-party application makes a customer data request call to the Oracle Utilities Opower GraphQL API, authorized by the access token with pre-defined scopes. For more information on scopes, refer to the GraphQL API Documentation.
9. The third-party application issue a session cookie and displays the website to the logged in customer, including applicable data and insights from the Oracle Utilities Opower GraphQL API call.

Offline Data Access

In order to access user data after the user access token has expired, which is 1 hour by default, and the user cannot be prompted to log in again, Identity and Access Management can be configured to return a refresh token. The refresh token is a one-time token that can be used to obtain a new access token and another refresh token. Utilities are required to configure Identity and Access Management to return refresh tokens to support their use by third-party partners for periodic user data access.

To obtain a refresh token, Identity and Access Management supports special scope `offline_access`. Examples of requesting a refresh token in Authorization Code Flow are available at https://docs.oracle.com/en/cloud/paas/identity-cloud/rest-api/ACWebServerAppAuth.html .

Documentation for using the refresh token grant type is available at https://docs.oracle.com/en/cloud/paas/identity-cloud/rest-api/RefreshGT.html.

Single Logout (SLO)

Single Logout (SLO) is a supported feature for third-party partner SSO. The SLO profile of the OAuth specification provides for coordinated and near-simultaneous logout across applications with a federated authentication context. Oracle's implementation of OAuth 2.0 and OpenID

Connect can include SLO capabilities, which allows a user to log out of multiple connected applications simultaneously when initiating a logout from one of them. Refer to the following information to configure SLO:

- Review the available Oracle Identity and Access Management documentation for the logout endpoint.
- The endpoint can be constructed using one of the following methods:
  - `https://{IDCS_HOST}/oauth2/v1/userlogout` where `IDCS_HOST` is the host and port information for you Identity and Access Management server.
  - Use the Identity and Access Management discovery URL `https://{IDCS_HOST}/.well-known/openid-configuration`, where `IDCS_HOST` is the host and port information for you Identity and Access Management server, to retrieve the value from `end_session_endpoint`.
- To trigger an SLO request, you must pass the following parameters to the `userlogout` endpoint: `https://{IDCS_HOST}/oauth2/v1/userlogout?id_token_hint={ID_Token}&post_logout_redirect_uri={Post_Logout_URL}`
  - `IDCS_HOST` is the host and port information for you Identity and Access Management server
  - `ID_Token`: The ID token from the Identity and Access Management response to a user login, post logout URL. This location is where a user is returned to when the Authorization Code Flow is completed.
  - `Post_Logout_URL`: The URL must be allow-listed in Identity and Access Management before it can be used. Multiple URLs can be allow-listed, and examples per tier are provided below:
    - **Stage**: `https://www.qadomain.net/utility_name/Secure/Logout.aspx`
    - **Production**: `https://www.proddomain.net/utility_name/Secure/Logout.aspx`

Testing Procedures

For third-party applications using OAuth and Identity and Access Management (IAM), testing procedures must focus on verifying the secure exchange of authorization credentials and validating that the application can only access resources explicitly permitted by the user. Testing must cover functional, security, and integration aspects to ensure that the application's implementation correctly follows the OAuth and IAM protocols.

To verify a successful connection and assist with troubleshooting, Oracle Utilities needs the ability to log in on the third-party partner’s stage environment. This may require VPN access if the stage environment is located behind a firewall. This access also requires at least one valid login on the stage environment. After testing is complete, the configurations are migrated to the production applications for both Oracle Utilities and the third-party partner. To verify these connections, Oracle Utilities also needs a test account on production.

The stage and production test accounts should be available for the life of the program for continuous verification of end-to-end SSO functionality. For coordination of providing these test account, refer to Contact Your Delivery Team

Functional Testing

Functional testing confirms that the authorization flows work as expected and the application can successfully access requested resources, and is recommended to include the following:

- **Test All OAuth 2.0 Flows**:
  - **Authorization Code Flow (with PKCE)**: Simulate the complete authorization code grant flow, including obtaining the authorization code, exchanging it for an access token, and using the token to access a protected resource. This is the recommended flow for most applications, especially public clients, including both mobile and single-page applications.
  - **Client Credentials Flow**: Verify that machine-to-machine communications for trusted applications can securely request an access token using only the client ID and secret.
- **Validate Token Handling**:
  - **Access Token Requests**: Confirm that the application can successfully request, receive, and use access tokens from the authorization server.
  - **Token Expiration and Refresh**: Test that the application can handle token expiration gracefully, which involves using the refresh token to obtain a new access token without requiring the user to re-authenticate.
  - **Token revocation**: Validate that revoking a token on the authorization server correctly invalidates the access for the third-party application.
- **Perform User Experience Testing**:
  - **Consent Screen Review**: Verify that the user consent screen, which displays the permissions the third-party application is requesting, is accurate, clear, and informative.
  - **User Login and Redirection**: Confirm that the user is properly redirected to the third-party application after a successful login and authorization.

2

Contact Your Delivery Team

Your Oracle Delivery Team is the group responsible for setting up, configuring, launching, or expanding your Oracle Utilities Opower program. Contact your Delivery Team if you have any questions about your program products and implementation.

To contact your Delivery Team:

1. Sign in to Inside Opower (https://inside.opower.com). This is your portal for questions and information related to your program.
2. Go to the Community tab to see who is on your Delivery Team.
3. Contact any of the team members using the information provided.

If you need to report an issue or get technical support, contact My Oracle Support.

Glossary

Index