Oracle Utilities Opower Single Sign-On

Configuration Guide

E84772-19

Last Updated: June 16, 2023

Oracle Utilities Opower Authentication Configuration Guide

E84772-19

Contents

Getting Started

Single sign-on (SSO) makes it easier for customers to access their energy usage details by allowing them to use their utility web application user name and password to gain access. Customers can log into their utility website and then navigate to the Energy Efficiency Web Portal without creating an additional account. If customers attempt to access the Energy Efficiency Web Portal and are not currently logged into the utility website, they are automatically directed to the utility website to sign in, and then are returned to the content they were trying to view.

Note: While this documentation refers to the Energy Efficiency Web Portal, the SSO configuration information also applies to Digital Self Service - Energy Management, except where explicitly noted. For more information about the features and requirements of Digital Self Service - Energy Management, see the *Oracle Utilities Opower Digital Self Service - Energy Management Cloud Service Product Overview*.

Refer to "General SAML Requirements" on page 1 and "General OpenID Connect Requirements " on page 2below for information on SSO configurations with SAML or OpenID Connect. Utilities also have the option of implementing SAML-based single logout (SLO) with Oracle Utilities. When SLO is implemented, and a customer logs out of the Energy Efficiency Web Portal or the utility website, the customer is automatically logged out of both sites. See "SAML Single Logout (SLO) Configuration" on page 11 for more information.

General SAML Requirements

Oracle Utilities supports Security Assertion Markup Language (SAML) 2.0 to implement SSO with utilities. The use of SAML for SSO is for standalone web implementations only. If new versions are announced, Oracle Utilities will work to incorporate support for the latest SAML versions.

A utility's Oracle Cloud Infrastructure Identity and Access Management environment acts as the Service Provider (SP) and the utility acts as the Identity Provider (IdP). This means that customers log in on the utility website using their user name and password for the utility website. Customers can then access the Energy Efficiency Web Portal without having to log in again.

General OpenID Connect Requirements

Oracle Utilities supports the OpenID Connect protocol to authenticate users that interact with Oracle Utilities embedded widgets that are integrated using custom elements. OpenID Connect is built on top of the OAuth 2.0 authorization framework.

Note: For more information on integrating using custom elements, see the Oracle Utilities Opower Digital Self Service - Energy Management Embeddable Widgets Integration Guide.

The utility website acts as the Relying Party (RP) and must integrate with an OpenID Connect Provider. With this SSO implementation, customers can log in on the utility website and access embedded widgets without having to log in again. For more information on implementing SSO with OpenID Connect, refer to "OpenID Connect Single Sign-On (SSO) Configuration" on page 18.

Important: SLO implementation for OpenID Connect is configured between the RP and OpenID Connect Provider, and thus information on configuring OpenID Connect SLO or testing SLO are out of scope of this documentation. In general, when the RP website requests a customer logout, a redirect to an OpenID Connect Provider endpoint can complete the customer logout.

SAML Single Sign-On (SSO) Configuration

There are two configurations for implementing SAML-based SSO with the Oracle Utilities Opower Energy Efficiency Web Portal. The first is single account SSO, in which customers only have access to one account with their utility. See "SAML Attribute for Single Account SSO" on page 5. The second is multiple account SSO, in which customers have access to one or more accounts with their utility and need the ability to switch between them. The main difference in supporting these scenarios is that the multiple account implementation requires additional data elements in the SAML assertion that is used in the authentication process. The multiple account SSO implementation option can support users who have a single account or multiple accounts, and is the recommended implementation option for all utilities. See "SAML Attribute for Multiple Account SSO" on page 6.

Utility Configuration Checklist

SSO relies on standards-based communication between a federation server managed by the utility and the server managed by Oracle Utilities. The following steps are required to set up and configure SSO for single or multiple account SSO.

1. Set up two SAML 2.0 Identity Provider federated servers: Stage and Production. Also set up authentication services for Stage and Production.
2. Oracle Cloud Infrastructure Identity and Access Management (IAM) domains must be configured prior to implementation. Be aware that Oracle Utilities representatives will access these domains to assist with configuration. This access requires Oracle Utilities to create accounts for the domain, and email notifications are sent to the domain administrator when these accounts are created.
3. Provide Oracle Utilities with SAML metadata to connect to these servers.
4. Provide Oracle Utilities with test login accounts for end-to-end testing on these servers. If necessary, provide Oracle Utilities with VPN access to the Stage login page. If your test site is behind a firewall, ensure that you add the Oracle Utilities IP address to your allowlist. Contact your Delivery Team to retrieve the Oracle Utilities IP address value.

Supported SAML Single Sign-On Profiles

Oracle Utilities requires Service Provider (SP)-initiated SSO. SP-initiated SSO allows users to bookmark pages. Also, if an Oracle Utilities session expires while a user still has a window open, SP-initiated SSO allows them to log in again and automatically return to the resource they are using. Performing SP-initiated SSO requires that the utility have a functional SSO URL that Oracle Utilities can access to begin the SSO process.

Oracle Utilities also supports Identity Provider (IdP)-initiated SSO. Utilities may create links that take users to specific pages on the Energy Efficiency Web Portal (such as the Data Browser or **My Energy Use** section) by passing these URLs in the SAML `RelayState` parameter. Utilities must send Oracle Utilities a valid URL as a `RelayState` parameter. Oracle Utilities will provide utilities with the appropriate URL for the Energy Efficiency Web Portal, which should be used as the default `RelayState` parameter.

> **Note**: The Digital Self Service - Energy Management widgets require SP-initiated SSO.

Whether user access attempts employ IdP-initiated or SP-initiated SSO, utilities must ensure that their federation server only authenticates users that have permission to access the Energy Efficiency Web Portal.

For further information on SAML SSO profiles, see the Security Assertion Markup Language (SAML) V2.0 Technical Overview.

SAML Bindings

Identity Provider to Service Provider Binding

Oracle Utilities accepts SAML assertions from IdPs using the HTTP POST binding method. This means that all SAML assertions are sent as HTTP POST requests to the Oracle Utilities federation server. Oracle Utilities requires using HTTP POST and having the browser transmit the SAML assertion to the Oracle Utilities federation server. Oracle Utilities does not support artifact binding for SAML 2.0.

Note: Ensure that `X-Frame-Options` HTTP response header is excluded from the HTTP response. This requirement allows content to be embedded using iframes, which includes Digital Self Service - Energy Management embeddable widgets.

Service Provider to Identity Provider Binding

Oracle Utilities supports either HTTP redirect binding, or HTTP POST binding when sending authentication requests to the IdP. By default, Oracle Utilities uses HTTP redirect binding. This means that when Oracle Utilities begins the SP-initiated SSO process, Oracle Utilities issues an HTTP redirect to the user’s browser directing them to the Identity Provider. The Identity Provider federation service will then receive an HTTP GET request from the consumer and initiate the authorization process. Oracle Utilities does not support artifact binding on communication from Oracle Utilities to the Identity Provider.

SAML Single Sign-On Assertion Requirements

The SAML assertion for an SSO implementation requires a `RelayState` parameter, as well as specific data elements and security information. Many of these requirements are the same for both single and multiple account SSO. The main differences are in the required data elements, which are identified in the following sections.

RelayState Parameter

Identity Providers must send Oracle Utilities a `RelayState` parameter in the SAML assertion sent to Oracle Utilities. In IdP-initiated SSO, utilities can set up links that will take users directly to a specific page by sending the appropriate URL in the `RelayState`. Oracle Utilities will provide utilities with our Dashboard URL to be used as a default `RelayState` parameter. The Dashboard is an appropriate general page to send customers to once they log in. Utilities must use this Dashboard URL or another valid URL as the `RelayState` URL when using IdP-initiated SSO.

In SP-initiated SSO, if Oracle Utilities sends a `RelayState` parameter to the Identity Provider, the Identity Provider must send the `RelayState` parameter back to Oracle Utilities without any modifications, as stated in the SAML 2.0 specification. When Oracle Utilities sends a `RelayState` parameter in SP-initiated SSO, it will be an alpha-numeric token that refers to saved state information on the Oracle Utilities federation server, and the `RelayState` will not be a URL.

In some instances of SP-initiated SSO, Oracle Utilities will not send a `RelayState` parameter. For example, this occurs when the user clicks on the **Sign In** link on the Energy Efficiency Web Portal home page. In these situations, the Identity Provider should use the Dashboard URL for the `RelayState`.

SAML Data Elements

The required SAML data elements include the SAML subject and the SAML attribute. The elements of the SAML attributes can vary slightly depending on your implementation.

> **Warning**: Ensure that carriage return characters are not included in SAML responses. Inclusion of carriage returns can cause errors such as signature mismatches.

SAML Subject

The SAML subject must contain an anonymous user identifier, such as a GUID, that is used for the login process. This identifier must not contain any personally identifiable information (PII). Oracle Utilities will describe the exact value required for this field based on the implementation plan. It will be one or more of the fields passed in the historical and iterative data files about the particular customer. Oracle Utilities will use this identifier to determine which account to display to the customer.

SAML Attribute for Single Account SSO

A SAML attribute can provide additional attributes that are associated with a particular customer. The name of the attribute must be `userDataXML` and must have `<sso_user_properties>` as its value. Be aware that the attribute value must be enclosed in a `CDATA` block, which is demonstrated below, or alternatively use escaped XML characters. In the following example, the attribute value contains a `<property>` tag that specifies customer preferences or characteristics, such as a preferred language.

```
<saml:AttributeStatement xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic" Name="userDataXML">
    <saml:AttributeValue xsi:type="xs:string"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><![CDATA[<?xml
version="1.0" encoding="UTF-8" ?>
    <sso_user_properties>
          <property>
                <name>language_preference</name>
                <value>zh_HK</value>
          </property>
    </sso_user_properties>
    ]]></saml:AttributeValue>
  </saml:Attribute>
</saml:AttributeStatement>
```

A more complete code sample is defined in the appendix. See "XML Schema for Single Account SSO" on page 15 for more information.

Note: The language preference specification shown in the code sample above is optional and only applicable to utilities that allow their customers to view the Energy Efficiency Web Portal in different languages. See the *Oracle Utilities Opower Multilingual Configuration Guide* for more information.

SAML Attribute for Multiple Account SSO

A SAML attribute provides additional attributes that are associated with the particular customer. The information contained within the SAML attribute is what Oracle Utilities uses to determine which accounts a customer should be able to view once they are logged in. The name of the attribute must be `userDataXML`. The required values of the attribute, such as the `<authorized_accounts>` and `<user>` elements, are defined in the schema in SAML Assertion XML Schemas.

The SAML Attribute and Account ID values come from one or more of the fields passed in the historical and iterative data files about the particular customer. The transfer of these data files is set up with Oracle Utilities at the beginning of your program. The fields passed in the SAML must uniquely identify a customer record. While the required fields vary for each client, the fields `customer_id` and `premise_id` are commonly used to identify each customer record. If multiple fields are required, the values are concatenated and separated with a hyphen (-).

The SAML attribute also defines the initial account that should be shown to the customer after they are logged in. This is done using the `<initial_account id>` tag. An initial account must be defined regardless of whether the customer has access to one account or many accounts.

For example, the following is an example of the XML Oracle Utilities is expecting within the SAML attribute in a successful SAML assertion. Be aware that the attribute value must be enclosed in a `CDATA` block, which is demonstrated below, or alternatively use escaped XML characters. The example represents a scenario where two fields are required to uniquely identify a customer record:

```
<saml:AttributeStatement xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic" Name="userDataXML">
  <saml:AttributeValue xsi:type="xs:string"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><![CDATA[<?xml
version="1.0" encoding="UTF-8" ?>
      <authorized_accounts>
        <user>
          <display_name>John Smith</display_name>
            <language_preference>en_us</language_preference>
        </user>
        <initial_account id="123456-987654"/>
        <accounts>
          <account id="123456-987654">
            <name>Primary Residence</name>
          </account>
          <account id="123456-987655">
            <name>Secondary Residence</name>
          </account>
        </accounts>
      </authorized_accounts>
    ]]></saml:AttributeValue>
  </saml:Attribute>
</saml:AttributeStatement>
```

Note: The language preference specification shown in the code sample above is optional and only applicable to utilities that allow their customers to view the Energy Efficiency Web Portal in different languages. See the *Oracle Utilities Opower Multilingual Configuration Guide* for more information.

The following is an example of the XML Oracle Utilities is expecting within the SAML attribute for an unsuccessful SAML assertion:

```
<authorized_accounts>
  <error>Error - No such user</error>
</authorized_accounts>
```

Note that this XML is optional. An example of this code in the context of the larger schema is available in the appendix. See "XML Schema for Multiple Account SSO" on page 16 for more information.

Account Data Selector Formatting

When an SSO customer with multiple accounts goes to the Energy Efficiency Web Portal, their request to sign in is re-routed through a federation server, which goes back to the utility to ensure that the customer's credentials are correct. When the customer is returned to the Energy Efficiency Web Portal, Oracle Utilities is passed certain information about the customer in a SAML assertion. Information that Oracle Utilities gathers includes:

- **User Display Name**: The name that should be displayed at the top of the Energy Efficiency Web Portal, immediately after the **Logged in as...** text.
- **Primary Account**: The account that should be the default account to display in the account selector.
- **Account ID**: Represents the account identification for each customer. This number is created by the utility.
- **Account Name**: Indicates the name to use for the account. For example, a customer could name one account “Home” and another account “Office”.

An example of this formatting is:

```
<authorized_accounts>
  <user>
    <display_name>John Smith</display_name>
  </user>
  <initial_account id="123456-987654"/>
  <accounts>
    <account id="123456-987654">
      <name>Primary Residence</name>
    </account>
    <account id="123456-987655">
      <name>Secondary Residence</name>
    </account>
  </accounts>
```

```
</authorized_accounts>
```

Security Requirements

Security for SAML is achieved through several mechanisms. First, SAML assertions sent using POST binding from the Identity Provider must be digitally signed with the Identity Provider’s private key using XML signature. This is a requirement per the SAML specifications. Oracle Utilities will then verify the source with the corresponding public key. Assertions that fail this verification process are rejected. This mechanism ensures that only assertions originating from the proper utility client are accepted. Furthermore, data is encrypted via HTTPS during transfer. In addition, the `RelayState` parameter does not include a full URL when it is passed from Oracle Utilities to the utility and then back, but it is a reference to the desired URL which is stored on the Oracle Utilities federation server. This prevents unauthorized parties from tampering with the destination URL during transit.

SAML Single Sign-On User Experience

The user experience can be customized when implementing SAML SSO by controlling how the utility website initiates SSO and how users are directed to the Oracle Utilities website from the Oracle Utilities Opower Home Energy Reports (HERs). Some of the most widely used options are described below. The URL that is displayed in the HERs can point directly to the Energy Efficiency Web Portal or the utility website.

Visiting Utility Website First

In this scenario, the URL that is displayed in the HERs sends users directly to the utility website. After users log in, they can be presented with links that would take them to the Energy Efficiency Web Portal using SSO. These links can be customized to link to any particular page on the Energy Efficiency Web Portal. The disadvantage of this approach is that the user may never navigate to the Energy Efficiency Web Portal and only conduct other activities (like bill pay) based on the functionality of the existing utility website.

Another option is for the URL on the HERs to take the user directly to a landing page on the utility website that is specific for the Oracle Utilities Opower program. Users would be directed to the Energy Efficiency Web Portal using SSO after they log in. The advantage of this technique over using the overall utility homepage on the HER is that users accessing the website on the HER URL are taken directly to the Energy Efficiency Web Portal after logging in. This helps to maximize the use of the Energy Efficiency Web Portal. Users can still go the main utility website for other necessary functions and applications, such as paying bills.

Both these options use IdP-initiated SSO. The disadvantage of these options is that users cannot browse content on the Energy Efficiency Web Portal prior to logging in.

Visiting Energy Efficiency Web Portal First

Another option is for the Home Energy Report (HER) URL to take the user directly to the Energy Efficiency Web Portal. Consumers can then browse content that is available to non-logged in users. If users click on a link that requires that they be logged in to view the content, they are directed to the utility website to log in under SP-initiated SSO. After logging in, they are directed back to the Energy Efficiency Web Portal to the specific page they were trying to access.

The Energy Efficiency Web Portal home page also presents the user with links allowing them to sign in or register. The default configuration is to have these links invoke SP-initiated SSO and have the user directed back to the Energy Efficiency Web Portal after they complete the sign in process. The utility can elect to send the user to any arbitrary page on the Energy Efficiency Web Portal. These links can be configured to point to any specific URL on the utility (or any other website).

If the method described above is used, customers can browse content on the Energy Efficiency Web Portal that is accessible for users that are not logged in. The advantage of having the Home Energy Report contain the URL of the Energy Efficiency Web Portal is that this ensures users will land on the Energy Efficiency Web Portal after logging in.

SAML Single Sign-On Configuration Information

When implementing SSO, most utilities choose to contract with a federation server provider and configure settings through the provider’s interface. Configuration details are provided below.

Oracle Utilities Opower SAML Information

Oracle Utilities provides the utility with SAML metadata for production and staging servers. The metadata provided by Oracle Utilities includes the following information:

- **Oracle Utilities Opower SAML Entity ID**
- **Oracle Utilities Opower Assertion Consumer Service URL**
- **Default Target URL (RelayState Value)**

Information Required by Oracle Utilities from the Utility

Oracle Utilities requires that a Utility defines their SAML specification or extracts a SAML metadata definition, and provides either resource to Oracle Utilities. Refer to your IdP third-party documentation for steps on completing a SAML metadata extraction. The information in the specification or metadata file must include the following:

- **Utility SAML Entity ID**: The URL to the client IdP SAML endpoint, which is the client-side counterpart to the Oracle Utilities Opower entity ID.
- **Utility Public Key**: Oracle Utilities requires the public key for the corresponding private key the utility is using to sign their SAML assertions. SAML requires the IdP to sign all assertions submitted via POST with a private key. Oracle Utilities needs the public keys to verify the assertions were sent by the utility.
- **SAML Single Sign-On Service URL**: Required for SP-initiated SSO, in which the user visits the URL for the Energy Efficiency Web Portal before logging in at the client utility website. Oracle Utilities needs to redirect users to the utility to begin the sign-in process and afterwards they will be returned to the URL on the Energy Efficiency Web Portal they were trying to access. This is done by sending SAML messages to the partner’s federation server to begin a user's SSO process. This value is the URL Oracle Utilities will use to begin SP-initiated SSO. Oracle Utilities uses redirect binding to access this URL.
- **Logout Redirect URL**: This is an optional parameter. Oracle Utilities redirects the user to after they click the logout link.

SAML Single Logout (SLO) Configuration

This section describes the requirements for implementing single logout (SLO) with the Oracle Utilities Opower Energy Efficiency Web Portal. When SLO is implemented and a customer logs out of the Energy Efficiency Web Portal or utility site, they are automatically logged out of both sites. When the user clicks to log out of the Energy Efficiency Web Portal, Oracle Utilities can configure the final URL to which users are directed after the logout process is complete. For example, the utility may want the user redirected to the utility home page after the user clicks on the logout link on the Energy Efficiency Web Portal. SSO must be supported in order to implement SLO.

SAML Requirements

Oracle Utilities uses SAML 2.0 to implement SLO with clients. A utility'Oracle Cloud Infrastructure Identity and Access Management environment acts as the Service Provider (SP) and the utility acts as the Identity Provider (IdP). See "Getting Started" on page 1 for more information.

SAML Bindings

Identity Provider to Service Provider Binding

Oracle Utilities accepts SAML logout messages from Identity Providers using the HTTP POST Binding method. This means all SAML logout messages are sent as HTTP POST requests to the Oracle Utilities federation server. Oracle Utilities the use of HTTP POST and the browser transmits the SAML logout message to the Oracle Utilities federation server. For this reason, Oracle Utilities does not support Artifact Binding for SAML 2.0.

Service Provider to Identity Provider Binding

Oracle Utilities supports either HTTP Redirect Binding, or HTTP POST Binding when sending logout requests to the Identity Provider. By default, Oracle Utilities will use HTTP Redirect Binding. This means that when Oracle Utilities begins the SP Initiated SLO process, Oracle Utilities will issue an HTTP Redirect to the user's browser directing them to the Identity Provider. The Identity Provider federation service will then receive an HTTP POST request from the consumer and initiate the Authorization process. Oracle Utilities does not support Artifact Binding on communication from Oracle Utilities to the Identity Provider.

SAML Single Logout Assertion Requirements

The SAML assertion for an SLO implementation requires a SAML subject and security information.

SAML Subject

The SAML Subject must contain a user identifier. Oracle Utilities will describe the exact value required for this field based on the implementation plan. It will be one of the fields passed in the historical and iterative data files about the particular customer. This typically corresponds to the account number printed on a customer's bill or an identifier derived from the billing account number.

Security Requirements

Security for SAML is achieved through several mechanisms. First, SAML logout requests sent using POST Binding from the Identity Provider must be digitally signed with the Identity Provider's Private Key using XML Signature. This is a requirement per the SAML specifications. Oracle Utilities will then verify the source with the corresponding Public Key. Requests that fail this verification process will be rejected. This mechanism ensures that only requests originating from the proper utility client are accepted. Furthermore, data is encrypted via HTTPS during transfer.

SAML Single Logout Configuration Information

When implementing SLO, most clients choose to contract with a federation server provider and configure settings through the provider's interface. Configuration details are provided below.

Oracle Utilities Opower SAML Information

Oracle Utilities provides the utility with SAML metadata for production and staging servers. The metadata provided by Oracle Utilities includes the following information:

- **Oracle Utilities Opower SAML Entity ID**
- **Oracle Utilities Opower SLO URL**

Information Required by Oracle Utilities from the Utility

Oracle Utilities requires that a Utility defines their SAML specification or extracts a SAML metadata definition, and provides either resource to Oracle Utilities. Refer to your IdP third-party documentation for steps on completing a SAML metadata extraction. The information in the specification or metadata file must include the following:

- **Client SAML Entity ID**: Same concept as the Oracle Utilities Opower entity ID.
- **Client Public Key**: Oracle Utilities requires the Public Key for the corresponding Private Key the utility is using to sign their SAML requests. SAML requires the Identity Provider to sign all requests submitted via POST with a Private Key. Oracle Utilities needs the Public keys to verify the requests were sent by the utility client.
- **Client SAML SLO URL**: This is required for SAML SLO. It is the URL to which Oracle Utilities will send SAML SLO messages.
- **Logout Redirect URL**: This is an optional parameter. It is the URL Oracle Utilities will redirect the user to after they click the Logout link or make requests to the Oracle Utilities Logout URL.

SAML Testing Procedures

Oracle Utilities follows thorough testing procedures for SSO and SLO implementations. Oracle Utilities has separate instances of the Energy Efficiency Web Portal specifically for integration testing. This is known as our staging environment.

Before going live with a utility, the Oracle Utilities staging infrastructure is configured to accept SSO assertions from the corresponding utility testing environment. The client application and federation server must similarly be configured to send SSO assertions to the Oracle Utilities federation server.

To verify a successful connection and assist with troubleshooting, Oracle Utilities needs the ability to log in to the utility’s staging environment. This may require VPN access if the utility stage environment is located behind a firewall. Oracle Utilities also requires at least one valid login on the utility’s stage environment.

After testing is complete, the configurations are migrated to the production applications for both Oracle Utilities and the utility. 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 and SLO functionality.

SAML Assertion XML Schemas

The XML schemas in this section specify the structure and elements that Oracle Utilities expects in the SAML assertions for both the single and multiple account implementations.

XML Schema for Single Account SSO

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    id="sso_user_properties"
    elementFormDefault="qualified"
    version="v0.1.0">
    <xs:annotation>
        <xs:documentation xml:lang="en">
        </xs:documentation>
    </xs:annotation>
    <xs:element name="property">
        <xs:annotation>
            <xs:documentation xml:lang="en">
                An arbitrary property.
            </xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element
name="name" type="xs:string" minOccurs="1" maxOccurs="1"/>
                <xs:element
name="value" type="xs:string" minOccurs="1" maxOccurs="1"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="error" type="xs:string">
        <xs:annotation>
            <xs:documentation xml:lang="en">
                If there was a problem providing a valid response, this
should be set to contain an explanation of the problem.
            </xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="sso_user_properties">
        <xs:annotation>
            <xs:documentation xml:lang="en">
```

```
                Root element.
            </xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:choice>
                <xs:sequence>
                    <xs:element
ref="property" minOccurs="1" maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:element ref="error" />
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

XML Schema for Multiple Account SSO

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    id="authorized_accounts"
    elementFormDefault="qualified"
    version="v0.1.0">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            This describes the expected response to an Oracle Utilities
multiple account sso request.  The main purpose of this is to provide
additional authorization information about the user who has been
authenticated. The current version of this schema provides for a list of
accounts (in the domain of the utility company) to which the authenticated
user should have access.  Preferably, the response should provide an
"initial account" representing the account that the user should see upon
successful completion of authentication and authorization.
        </xs:documentation>
    </xs:annotation>

    <xs:element name="user">
        <xs:annotation>
            <xs:documentation xml:lang="en">
                Information about the user.
            </xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="display_name" type="xs:string" />
                <xs:element name="language_
preference" type="xs:string" minOccurs="0" maxOccurs="1"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="account">
        <xs:annotation>
            <xs:documentation xml:lang="en">
               An account (in the domain of the utility company) that the
user should have access to.
            </xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="name" type="xs:string" />
            </xs:sequence>
            <xs:attribute name="id" type="xs:NMTOKEN" use="required" />
        </xs:complexType>
    </xs:element>

    <xs:element name="accounts">
        <xs:annotation>
            <xs:documentation xml:lang="en">
                A list of accounts
            </xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element
ref="account" minOccurs="1" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="error" type="xs:string">
        <xs:annotation>
            <xs:documentation xml:lang="en">
                If there was a problem providing a valid response, this
should be set to contain an explanation of the problem
```

```
            </xs:documentation>
        </xs:annotation>
    </xs:element>

    <xs:element name="initial_account">
        <xs:annotation>
            <xs:documentation xml:lang="en">
                The initial account the user should be viewing after SSO
has completed.
            </xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:attribute name="id" type="xs:NMTOKEN" use="required" />
        </xs:complexType>
    </xs:element>

    <xs:element name="authorized_accounts">
        <xs:annotation>
            <xs:documentation xml:lang="en">
                Root element.
            </xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:choice>
                <xs:sequence>
                    <xs:element ref="user" minOccurs="0" maxOccurs="1"/>
                    <xs:element ref="initial_
account" minOccurs="0" maxOccurs="1"/>
                    <xs:element
ref="accounts" minOccurs="1" maxOccurs="1"/>
                </xs:sequence>
                <xs:element ref="error" />
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

OpenID Connect Single Sign-On (SSO) Configuration

You can implement OpenID Connect for SSO purposes to support embedded widgets integrated using custom elements. To implement SSO with OpenID Connect, the utility website acts as the Relying Party (RP) and must have an integrated authentication server that conforms to the OpenID Connect specification. This documentation refers to these authentication servers as OpenID Providers.

Note: Standalone Oracle Utilities products rely on SAML for SSO implementation. A utility must use OpenID Connect to implement SSO for embedded widgets. Refer to the Oracle Utilities Opower Digital Self Service - Energy Management Embeddable Widgets Integration Guide for information on embedding widgets using custom elements.

Utility Configuration Checklist

SSO relies on standards-based communication between an OpenID Provider authentication server managed by the utility and the server managed by Oracle Utilities. The following steps are required to set up and configure SSO for OpenID Connect.

Note: Oracle Utilities supports configuration with multiple authorization servers. In this scenario, the following configuration checklist must be completed for each authorization server.

- Set up two OpenID Providers, one stage server and one production server.
- Implement customer "Access Tokens" on page 20 which must be passed in through an Opower API for webpages that embed widgets. For information on the separate embedding process, see the Oracle Utilities Opower Digital Self Service - Energy Management Embeddable Widgets Integration Guide.
- Complete the steps for "Creating a UserInfo Endpoint with Custom Claims" on page 21.
- Provide the UserInfo Endpoint URI, the applicable public key URI, and an optional subscription key to the Oracle Utilities Delivery Team. The public key URI is made available by the OpenID Provider, commonly through the use of a discovery endpoint. Confirm this URI with the OpenID Provider if required. Refer to "Access Tokens" on page 20 for information on UserInfo Endpoint URI requirements based on your setup.
- Enable VPN access and test login accounts for the Oracle Utilities Delivery Team to your stage environment. This access is used to complete required testing and troubleshooting. If your test site is behind a firewall, ensure that you add the Oracle Utilities IP address to your allowlist. Contact your Delivery Team to retrieve the Oracle Utilities IP address value.

Access Tokens

Your web portal system must be able to supply an access token of the currently logged in user to Oracle Utilities embedded widgets. Oracle Utilities does not support refreshed access tokens that have expired. Your web portal system must support subsequent access token requests to retrieve a new access token when the previous access token has expired (see "Authenticated Connection Storage" on page 23).

JSON Web Token Access Tokens

Oracle Utilities recommends supporting access tokens as JSON Web Token (JWT), which conform to the JWT standard, as it allows for additional security validations as compared to opaque access tokens.

You must define the following JWT registered payload claims:

- `aud`: The list of Audiences, which are authorized to call an Oracle Utilities Opower API endpoint. Audiences are defined by the OpenID Provider. Confirm the audience values with the OpenID Provider and provide the values to the Oracle Utilities Delivery Team.
- `iat` and `exp`: These two claims define the access token expiration timeout. The `iat` claim defines the time the access token was issued, and the `exp` claim defines when the access token should expire. Be aware that Oracle Utilities enforces a maximum access token expiration timeout of 24 hours. If the difference between `iat` and `exp` claims in the token exceeds 24 hours, the 24 hour maximum from `iat` time is enforced.

Warning: Failure to provide valid `aud`, `exp`, or `iat` claims results in an authentication error.

Opaque Access Tokens

Oracle Utilities supports the use of opaque access tokens but discourages their use as they do not benefit from the security advantages of JWT access tokens.

Opaque access tokens do not provide a uniform method to describe access token expiration. Therefore, Oracle Utilities allows for the configuration of a token expiration. If no expiration is specified, Oracle Utilities enforces a default expiration of 15 minutes.

Creating a UserInfo Endpoint with Custom Claims

You must create a UserInfo API endpoint, which is used by Oracle Utilities to retrieve additional user information. After validating a JWT access token (see "Access Tokens" on page 20), Oracle Utilities sends an HTTPS `GET` request with the `Authorization: Bearer` *`OP_issued_access_token`* header to the URI that you have provided. Oracle Utilities calls this endpoint to:

- Retrieve a list of all utility accounts associated with a web user.

  If the web user has multiple premises and you intend to support account switching in your web portal, the UserInfo endpoint must respond with all valid accounts for the web user. For additional information on account switching, see the Oracle Utilities Opower Digital Self Service - Energy Management Embeddable Widgets Integration Guide.
- Retrieve a unique user identifier using the registered token claim `sub`. This endpoint must conform to the OpenID Connect specification about UserInfo Endpoints and support custom claims.

Note: While the OpenID Connect standards require a POST method, Oracle Utilities does not require a POST method to support the OpenID Connect workflow. Therefore its definition is out of scope of this documentation.

Be aware that this secure `GET` request is a service-to-service request which uses the access token in the header section. No data or information is sent as part of the `GET` query parameters through the requested URL. The `GET` request must use HTTPS and thus SSL verification is also required for the UserInfo Endpoint.

Note: In addition to the security provided by the access token, you can also request a subscription key to be provided as part of the `GET` request. The subscription key must be retrieved from your OpenID Provider and then must be sent to your Oracle Utilities Delivery Team using the methods described in "Contacting Your Delivery Team" on page 27.

A sample UserInfo `GET` Endpoint is provided below, along with a table of possible responses.

```
GET /userinfo
Authorization: Bearer OP_issued_access_token
```

Status Code	Response	Comments

200	`{`
	`"sub": "248289761001",`
	`"user_accounts": [`
	`{`
	`"id": "434343",`
	`"display_name": "My Home"`
	`},`
	`{`
	`"id": "1212121",`
	`"display_name": "My Office"`
	`}`
	`]`
	`}`
		`sub`: Required. This is any unique identifier for the logged in web user. Oracle Utilities recommends that you keep this value consistent with the web user. Oracle Utilities does not use this identifier for customer lookup.
		`user_accounts`: A minimum of one account is required for this custom claim.
		■ `id`: Required. The field or fields passed in this account ID must uniquely identify a customer record for retrieval. The most common value to pass in as the account ID is `customer_id-premise_id`. If multiple fields are required, as is the case with this example, the values are concatenated and separated with a hyphen. This account ID is what is sent to Oracle Utilities through the billing file integration between the utility and Oracle Utilities.
		■ `display_name`: Optional. This is the account name that can be displayed in the user interface.

401	`{`
	`"error": "invalid_token",`
	`"error_description": ""`
	`}`
		The bearer token has expired.

JSON Schemas

The example JSON schema specifies the structure and elements that Oracle Utilities expects in the JSON for the /userinfo endpoint, which is required in support of

OpenID Connect implementation of SSO for embedded widgets. For more information on the /userinfo endpoint, see "Creating a UserInfo Endpoint with Custom Claims" on page 21.

```
{
   "definitions": {},
   "$schema": "http://json-schema.org/draft-07/schema#",
   "$id": "http://example.com/root.json",
   "type": "object",
   "title": "The Root Schema",
   "required": [
       "sub",
       "user_accounts"
   ],
   "properties": {
       "sub": {
           "$id": "#/properties/sub",
           "type": "string",
           "title": "Unique identifier of currently logged in user",
           "default": "",
           "pattern": "^(.+)$"
       },
       "user_accounts": {
           "$id": "#/properties/user_accounts",
           "type": "array",
           "title": "The user_accounts Schema",
           "items": {
               "$id": "#/properties/user_accounts/items",
               "type": "object",
               "title": "The Items Schema",
               "minProperties": 1,
               "required": [
                   "id"
               ],
               "properties": {
                   "id": {
                       "$id": "#/properties/user_accounts/items/properties/id",
                       "type": "string",
                       "title": "account id",
                       "pattern": "^(.+)$"
                   },
                   "display_name": {
                       "$id": "#/properties/user_
accounts/items/properties/display_name",
                       "type": "string",
                       "title": "The display_name Schema",
                       "default": ""
                   }
               }
           }
       }
   }
}
```

Authenticated Connection Storage

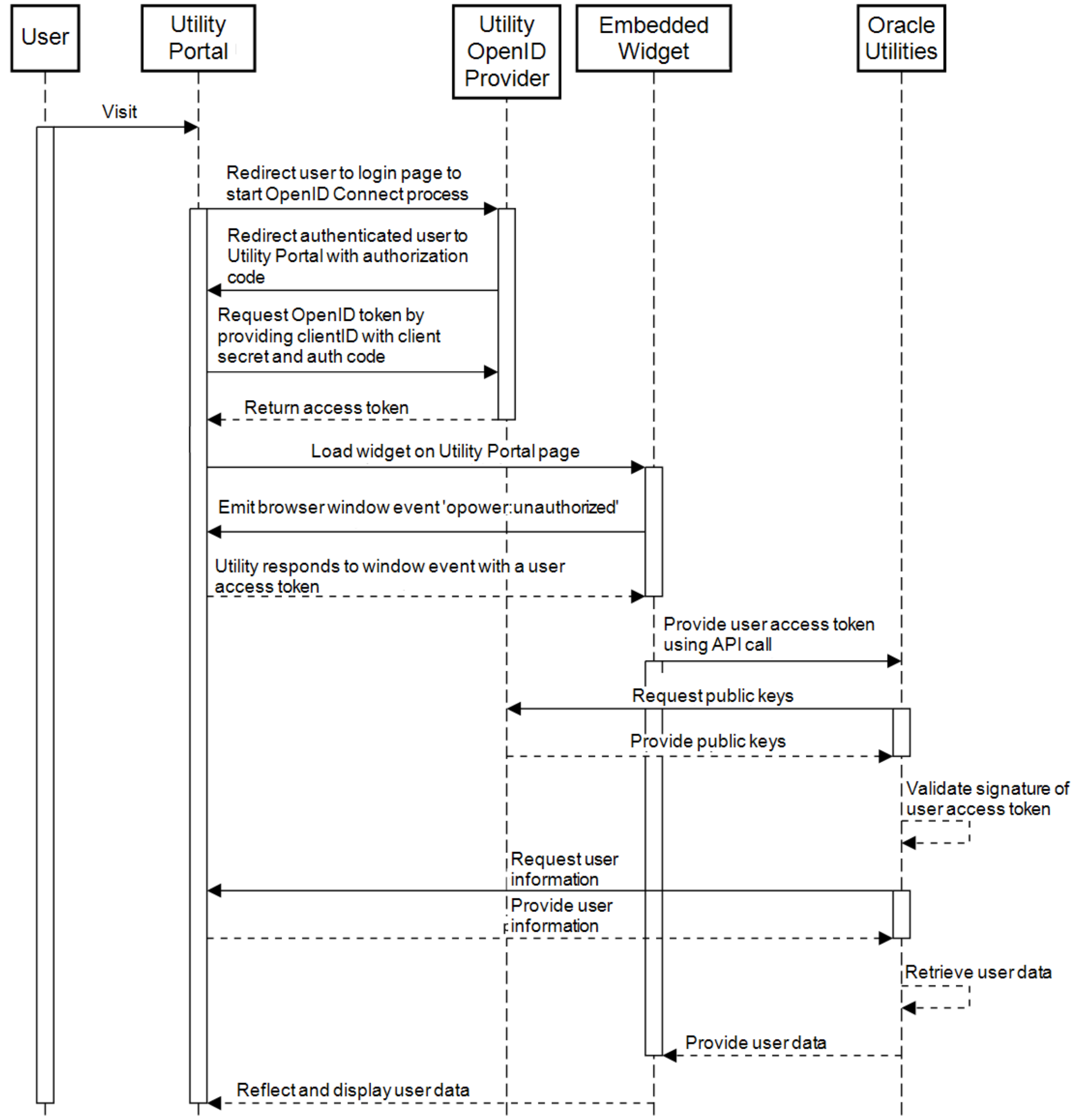
Oracle Utilities stores access tokens and user information after user validation. Any subsequent calls to Oracle Utilities services while the access token is still valid re-use the stored information. After expiration, Oracle Utilities requires a new token to initiate another authenticated connection. For information on supplying access tokens, see the Oracle

Utilities Opower Digital Self Service - Energy Management Embeddable Widgets Integration Guide.

OpenID Connect Workflow

The following diagram describes the overall workflow that is completed to support OpenID Connect authentication. This includes a user's initial log in to the utility portal through the final display of a widget. A step-by-step description of the workflow is provided along with the diagram.

Note: The workflow described below assumes the use of JWT access tokens (see "JSON Web Token Access Tokens" on page 20). Access token validation steps are not applied by Oracle Utilities if opaque access tokens are used, and any opaque access token validation is the responsibility of the utility.


User
Utility Portal
Utility OpenID Provider
Embedded Widget
Oracle Utilities
Visit
Redirect user to login page to start OpenID Connect process
Redirect authenticated user to Utility Portal with authorization code
Request OpenID token by providing clientID with client secret and auth code
Return access token
Load widget on Utility Portal page
Emit browser window event 'opower:unauthorized'
Utility responds to window event with a user access token
Provide user access token using API call
Request public keys
Provide public keys
Validate signature of user access token
Request user information
Provide user information
Retrieve user data
Provide user data
Reflect and display user data

When a user accesses embedded widget content and an active user session is not available, the authentication process using OpenID Connect is started. This process involves the user, the utility portal hosting the embedded widget, the utility OpenID Provider, the embedded widget, and Oracle Utilities services and data.

1. The process begins with a user accessing a utility portal page with an embedded widget. A user access token is required for Oracle Utilities to authenticate the user and display an embedded widget. A user access token can be obtained by following the example steps below for OpenID Connect, or using other means:
   a. The utility portal redirects the user to a login page hosted by the utility OpenID Provider to begin the authorization process.

   b. The utility OpenID Provider redirects the authenticated user to the utility portal with an authorization code.
   c. The utility portal requests an OpenID Connect token from the utility OpenID Provider by providing a clientID and client secret and an authorization code.
   d. The utility OpenID Provider returns an access token to the utility portal. This completes the user's log in to the utility portal.
2. After a valid access token is retrieved by the utility portal, the utility portal makes a request to load the embedded widget for the user.
3. An embedded widget emits browser window event `opower:unauthorized`.
4. The utility portal must respond to this event with a user access token. For more information on these requirements, see "Access Tokens" on page 20.
5. The embedded widget provides the user access token to Oracle Utilities using an API call.
6. Oracle Utilities starts user access token validation for JWT access tokens, which first requests the applicable public keys from the utility OpenID Provider.
7. The utility OpenID Provider public key URI provides the public key information to Oracle Utilities.
8. Oracle Utilities validates the signature of the user JWT access token with the information provided by the utility.
9. Oracle Utilities requests user information from the UserInfo endpoint that is provided by the utility.
10. The utility portal provides the requested user information through a UserInfo Endpoint. For more information on these requirements, see "Creating a UserInfo Endpoint with Custom Claims" on page 21. Be aware that the user information request uses a timeout of 10 seconds.
11. Oracle Utilities retrieves the applicable user data.
12. Oracle Utilities provides the user data to the embedded widget.
13. The embedded widget updates on the utility portal to reflect and display applicable user data.

OpenID Connect Single Sign-On User Experience

The Digital Self Service - Energy Management widgets can be embedded on a utility’s website. Oracle Utilities supports SSO using OpenID Connect to present a consistent view of the customer’s data on the utility website and all embedded widgets. This means that customers can log in on the utility website and access embedded widgets without having to log in again.

OpenID Connect Testing Procedures

Oracle Utilities follows thorough testing procedures for SSO OpenID Connect implementations. Oracle Utilities has a separate staging environment for integration testing. This infrastructure is completely separate from the production Oracle Utilities infrastructure.

Before going live with a utility, the Oracle Utilities staging infrastructure is configured to accept OpenID Connect access tokens from the corresponding utility testing environment. The utility website must similarly be configured to send access tokens to Oracle Utilities, as described in "Creating a UserInfo Endpoint with Custom Claims" on page 21.

In order to verify a successful connection and assist with troubleshooting, Oracle Utilities needs the ability to log in to the utility’s staging environment. This may require VPN access if the utility stage environment is located behind a firewall. Oracle Utilities also requires at least one valid login on the utility’s stage environment.

After testing is complete, the configurations are migrated to the production applications for both Oracle Utilities and the utility. 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.

Contacting Your Delivery Team

Your Oracle Utilities 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. Log 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.