Oracle® Retail Xstore Office Cloud Service

Security Guide

Release 18.0

F13698-03

November 2020

ORACLE®

Oracle® Retail Xstore Office Cloud Service Security Guide, Release 18.0

F13698-03

Primary Author: Nishant Rao

Contributing Author:

Contributor:

Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.

Contents

Send Us Your Comments vii

Preface ix

Audience ix
Documentation Accessibility ix
Related Documents ix
Customer Support ix
Improved Process for Oracle Retail Documentation Corrections x
Oracle Retail Documentation on the Oracle Technology Network xi
Conventions xi

1 Overview

Product Overview 1-1
Identity Cloud Service 1-2
Cloud Deployment Overview 1-2

2 Security Features

Security Model 2-1
Xstore Office Cloud Service Provisioning 2-1
Authentication 2-1
Multi-Factor Authentication (MFA) 2-2
Access Control 2-2
Security Audit 2-2
Credential Rotation 2-2

3 Security Considerations for Developers

Creation of Users 3-1
IDCS 3-1
Manual Creation 3-1
Bulk Import 3-1
REST APIs 3-1
User Access AppRole 3-2
Xadmin 3-2
Xoffice OAuth Client AppRoles 3-2

User Access 3-2
Xstore Access 3-2
Data Privacy Access 3-3
Service Access 3-3
Cloud Enrollment of Xstore Clients 3-3
Xadmin 3-3
Xenvironment 3-3
Creation of OAuth Clients in IDCS 3-4
Prerequisites 3-4
Required Data 3-4
Tools 3-5
Xstore Office Setup App 3-5
Xstore Office Data Migration App 3-8
Xstore Office Data Privacy App 3-11
RTLog Generator Client App 3-14
Additional Apps 3-16
Deletion of OAuth Clients in IDCS 3-17

Send Us Your Comments

Oracle® Retail Xstore Office Cloud Service Security Guide, 18.0

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

> **Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network Web site. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at http://www.oracle.com.

Preface

The *Oracle® Retail Xstore Office Cloud Service Security Guide* describes the security features and security considerations for the Oracle Retail Xstore Office Cloud Service application.

Audience

This guide is for technical personnel who configure, maintain and support, or use Oracle Retail Xstore Office.

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.

Access to Oracle Support

Oracle customers that have purchased support 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.

Related Documents

For more information, see the Oracle Retail Xstore Point- of-Service and Oracle Retail Xstore Office documentation set.

http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

https://support.oracle.com

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received

- Screen shots of each step you take

Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times not be attached to a numbered software release; instead, the Oracle Retail document will simply be replaced on the Oracle Technology Network Web site, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

This process will prevent delays in making critical corrections available to customers. For the customer, it means that before you begin installation, you must verify that you have the most recent version of the Oracle Retail documentation set. Oracle Retail documentation is available on the Oracle Technology Network at the following URL:

http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-02 is an updated version of a document with part number E123456-01.

If a more recent version of a document is available, that version supersedes all previous versions.

Oracle Retail Documentation on the Oracle Technology Network

Oracle Retail product documentation is available on the following web site:

http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html

(Data Model documents are not available through Oracle Technology Network. You can obtain them through My Oracle Support.)

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Overview

This chapter provides a product and cloud deployment overview of Xstore Office Cloud Service.

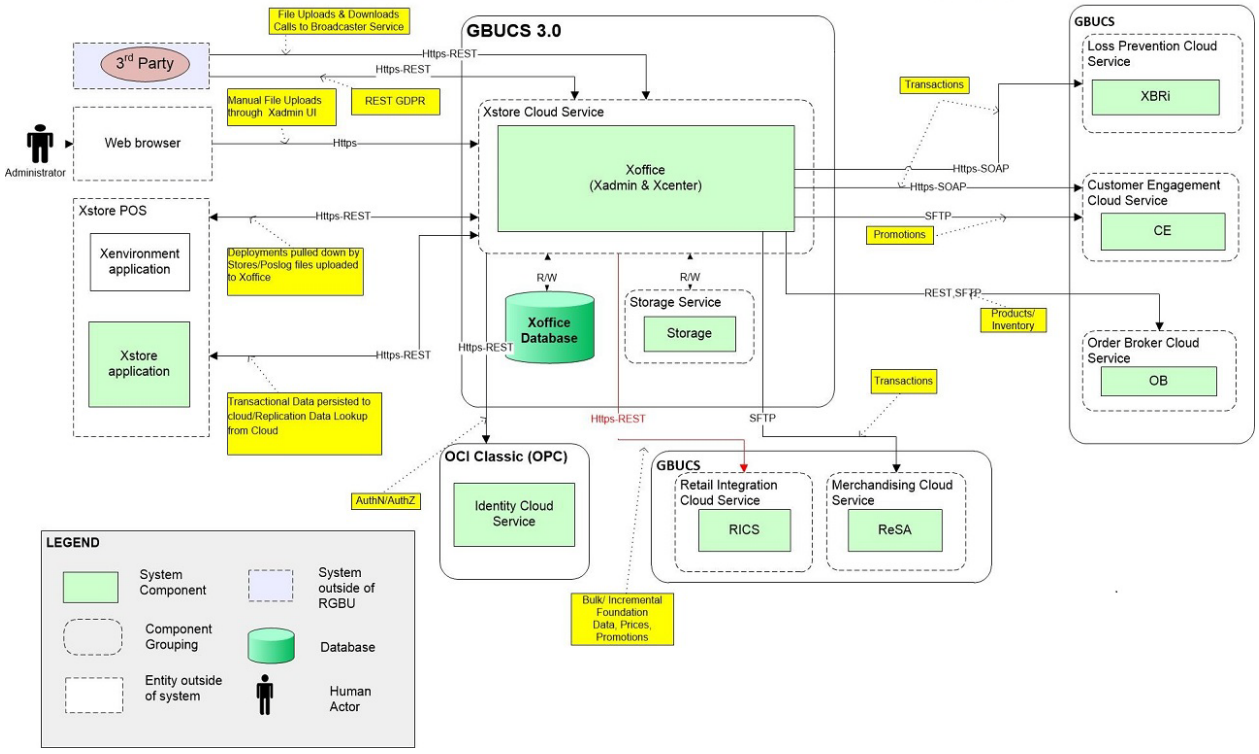
Product Overview

Oracle Retail Xstore Office Cloud Service is a web-based application used to administer corporate based functions such as Oracle Retail Xstore Point-of-Service configuration setup and maintenance, file management for the Oracle Retail Xstore Suite, viewing the electronic journal, viewing store reports, monitoring Oracle Retail Xstore Point-of-Service versions in use at the store and register levels, and monitoring alerts.

Xstore Office Cloud Service consists of a User Interface component called Xadmin and a Web Services component called Xcenter, a Java and JSON-based messaging framework.

Xstore Office Cloud Service integrates with several other products as shown in the following Architecture Model. All incoming web services are RESTful and secured with OAuth 2.0. Outgoing web services are RESTful or SOAP based and secured with OAuth, Basic Auth or Custom Auth.

Figure 1–1 Architecture Model



Identity Cloud Service

The Identity Cloud Service (IDCS) is an Identity Management Service and Authorization Server and has a host of other features and capabilities. For more information on IDCS, see the Oracle Identity Cloud Service documentation set, Get Started portal for IDCS.

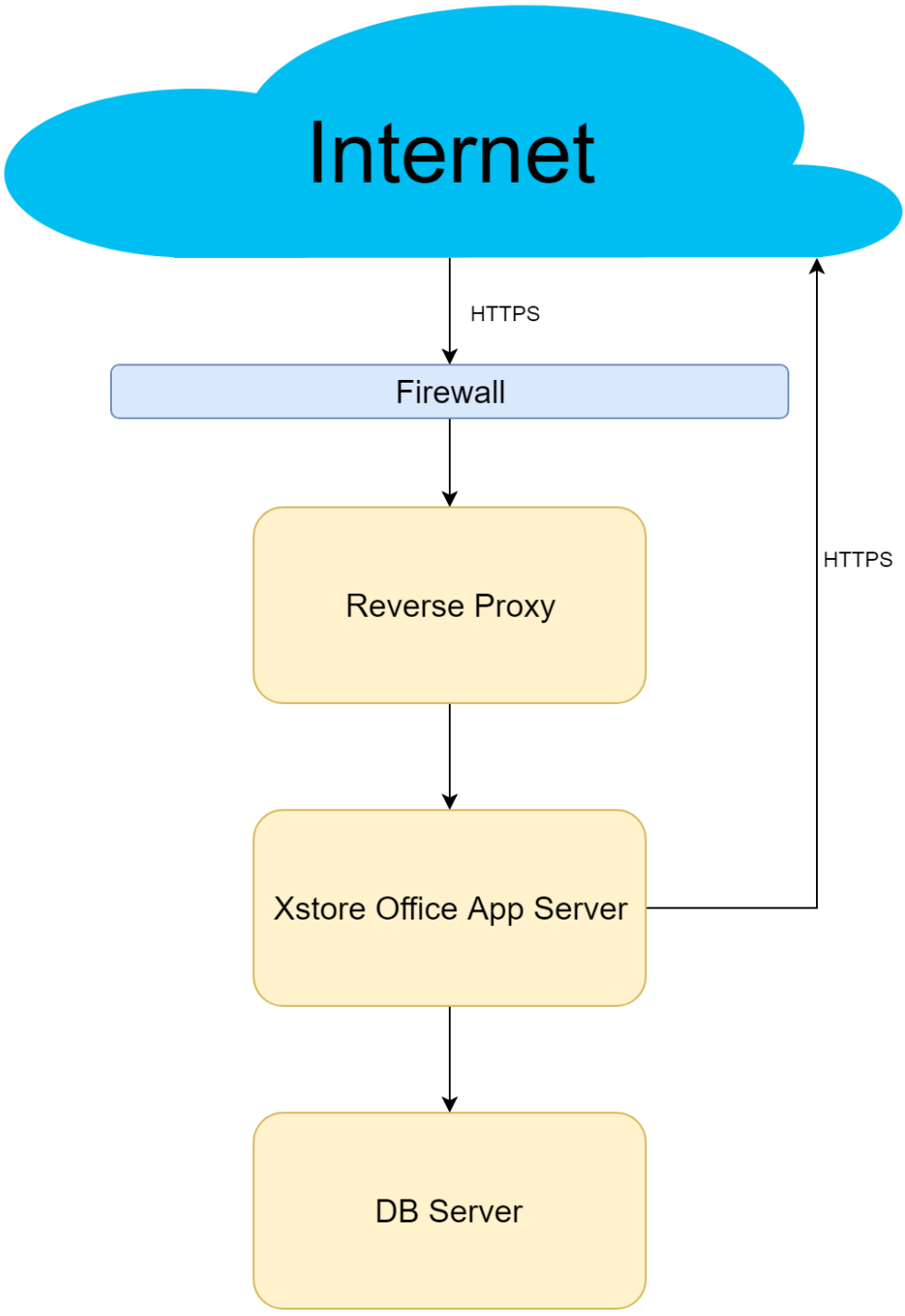
https://docs.oracle.com/en/cloud/paas/identity-cloud/index.html

Xstore Office Cloud Service integrates with Identity Cloud Service (IDCS). For Xstore Office Cloud, IDCS is primarily used for Identity Management (that is, storing user information), for securing REST services using the Open Authorization (OAuth) 2.0 and Login via the OAuth 2.0 and OpenID Connect (OIDC) protocols.

Cloud Deployment Overview

The following diagram describes an Xstore Office Cloud deployment.

Figure 1–2 Xstore Office Cloud Deployment Process



All incoming and outgoing service communication with the Cloud instance requires TLS for transport security.

Reverse Proxy: The Reverse Proxy intercepts all incoming requests to Xstore Office Cloud and authorizes and/or authenticates the requests based on the Xstore Office Cloud Web Tier Policy defined in IDCS.

Xstore Office Application Server: The Application Server hosts the Xstore Office Cloud Service Applications including Xadmin which is the User Interface component and Xcenter which is the Web Services component (a Java and JSON-based messaging framework).

DB Server: The Oracle database server contains the database schemas required by the Xstore Office Cloud Service.

2

Security Features

This chapter describes the available security features of the Xstore Office Cloud Service.

Security Model

Xstore Office Cloud Service integrates with Identity Cloud Service (IDCS) for Identity Management (that is, storing user information), for securing REST services using the Open Authorization (OAuth) 2.0 and Secure User Authentication via the OAuth 2.0 and OpenID Connect (OIDC) protocols.

A Reverse Proxy is in place that intercepts all incoming requests to Xstore Office Cloud Service and authorizes and/or authenticates the requests based on the Xstore Office Cloud Web Tier Policy defined in IDCS.

Xstore Office Cloud Service Provisioning

During Xstore Office Provisioning, Xstore Office OAuth Clients (or Apps) are created in IDCS with custom AppRoles. The Custom AppRoles are used to perform additional Application Level authorizations in addition to Application Level Privilege authorizations.

At the time of provisioning, a Customer Administration User is also created, who initially, is the sole user with access to the Xstore Office Cloud Service application. It is the responsibility of the Customer Administration User to create users with the appropriate privileges for functionality that will become available to them. It is recommended that users are granted the least level of access they require to perform their duties.

Authentication

Xadmin delegates the login to IDCS. Therefore, it does not prompt the user to login and does not store any user credentials. Instead, when a user accesses Xadmin, the Reverse Proxy determines whether this user's session already exists in IDCS. If so, it forwards to Xadmin. If this user's session does not exist, then the Reverse Proxy redirects to IDCS prompting the user to enter their credentials. If the user successfully authenticates in IDCS, then the request is forwarded to Xadmin. Once at Xadmin, additional application level authorization is performed to determine the user's role and privileges granted to the user in order to display the appropriate features that the user is authorized to access.

For details on how users are created and provisioned, see the Creation of Users section.

Multi-Factor Authentication (MFA)

IDCS provides the ability to enable Multi-Factor Authentication. For more information on enabling Multi-Factor Authentication, see the *Oracle Cloud Administering Oracle Identity Cloud Service Guide*.

Access Control

Xcenter REST APIs are secured with OAuth 2.0 protocols and use OAuth tokens. When Xcenter REST Services are invoked, the Reverse Proxy intercepts the requests, uses the OAuth 2.0 protocol to authorize the OAuth tokens and forwards the request to Xcenter. Xcenter then performs additional application level authorization by examining the tokens to see if they were requested by an OAuth Client that was granted specific AppRoles defined in IDCS when Xstore Office OAuth Clients were provisioned. If the token contains the necessary AppRole Grants, Xcenter provides access to the endpoint and the appropriate response is returned.

For details on how users are created and provisioned, see the Creation of Users section.

Security Audit

User Identity (account name or IP address) is recorded in the application logs when accessing Xadmin or invoking Xcenter REST APIs. In addition, date, time, information, software or configuration changes are also recorded in the application logs.

IDCS provides several reports that are detailed in the *Oracle Cloud Administering Oracle Identity Cloud Service Guide*.

Credential Rotation

All credentials in use within the Xstore Office Cloud Service will be rotated on a regular schedule.

3

Security Considerations for Developers

This chapter describes the security considerations for developers.

Creation of Users

A Customer Administration User will be created as part of the Xstore Office Cloud Service provisioning process. Before end users can access the Xstore Office Cloud Service application it is necessary to create and provision users. This includes provisioning access to the system, assigning organizations, a role and org nodes to each user to control what functionality will be available to them. This will need to be done by the Customer Administration User.

IDCS

Note: While users can be created using the IDCS UI, it is important to note that they must still be provisioned through the Xadmin UI. This includes provisioning access to the system, assigning organizations, a role and org nodes to each user to control what functionality will be available to them.

Manual Creation

Users can be created manually (that is one at a time) in IDCS by following the instructions on how to create user accounts in the *Oracle Cloud Administering Oracle Identity Cloud Service Guide*.

Bulk Import

Users can be bulk imported into IDCS by following the instructions on how to import user accounts in the *Oracle Cloud Administering Oracle Identity Cloud Service Guide*.

REST APIs

Users can also be imported (either individually or in bulk) into IDCS by invoking IDCS REST APIs. For more information about the REST APIs, see the *Oracle Cloud REST API for Oracle Identity Cloud Service Guide*. This requires the creation of an OAuth Client in order to be able to obtain an OAuth Token to be able to invoke the REST APIs. For more information about creating OAuth Clients, refer to the Creation of OAuth Clients in IDCS section.

User Access AppRole

When a user is created using IDCS (either manually or via bulk import, or via REST APIs), it is the Customer Admin's responsibility to grant the User the *User Access* AppRole. This can be done in the IDCS UI as follows:

- Click on the Xstore Office App.
- Click on the Application Roles tab.
- Select the menu icon on the far right of the User Access AppRole.
- Select **Assign Users** and select the Users in the popup to be granted this AppRole.

Xadmin

Refer to the *Oracle Retail Xstore Office/Xstore Office Cloud Service User Guide* for details on how to create and provision users via the Xadmin User Management UI.

Xoffice OAuth Client AppRoles

AppRoles have been created in Xoffice OAuth Clients in order to perform additional App Level Authorization.

User Access

Typically, the IDCS tenant will represent several applications which are independent of each other. The *User Access* AppRole has been created in the Xstore Office App (which typically has a display name of either RGBU_XSTCS_**PRD**_XOFFICE or RGBU_XSTCS_**UAT**_XOFFICE depending on the environment). The *User Access* AppRole is used to link IDCS users with the Xstore Office Cloud Service Application.

When Xadmin performs a user sync against IDCS, it will do the sync based on the users that have been granted this *User Access* AppRole. Refer to the *Oracle Retail Xstore Office/Xstore Office Cloud Service User Guide* for details on the user sync between Xadmin and IDCS.

Whenever a user is created using the Xadmin UI, the user is automatically granted the *User Access* AppRole. When a user is created using the IDCS UI (either manually or via Bulk Import), it is the Customer Admin's responsibility to grant the user the *User Access* AppRole. This can be done as follows:

- Click on the Xstore Office App.
- Click on the Application Roles tab.
- Select the menu icon on the far right of the User Access AppRole.
- Select **Assign Users** and select the Users in the popup to be granted this AppRole.

Xstore Access

The *Xstore Access* AppRole is used for additional App Level Authorization. This authorization is done when Xstore Office REST APIs are invoked. Therefore, if the Xstore Office REST APIs are to be invoked, then they must be done by using an OAuth Client (App) that has been granted the *Xstore Access* AppRole. For instance, refer to the Xstore Office Setup App or Xstore Office Data Migration App in the Creation of OAuth Clients in IDCS section.

Data Privacy Access

The *Data Privacy Access* AppRole is used for additional App Level Authorization. This authorization is done when the Data Privacy REST APIs are invoked. Therefore, if the Data Privacy REST APIs are to be invoked, they must be done by using an OAuth Client (App) that has been granted the *Data Privacy Access* AppRole. Refer to the Xstore Office Data Privacy App in the Creation of OAuth Clients in IDCS section.

Service Access

The *Service Access* AppRole is used for internal manipulation of the OAuth Clients. This is also needed in case OAuth Clients need to be deleted. See the Deletion of OAuth Clients in IDCS section.

Cloud Enrollment of Xstore Clients

Any Xstore register that communicates with Xstore Office Cloud Service must first be enrolled in IDCS via Xstore Office Cloud Service. This can be done either via Xadmin or Xenvironment. The sections below contain information about the steps to be followed for Cloud Enrollment of Xstore Clients.

Xadmin

Xstore Stores can be enrolled in Xstore Office Cloud Service via Xadmin On-Premise, if the retailer has an existing Xadmin On-Premise application 18.0.1 or higher. Refer to the on-premise *Oracle Retail Xstore Office User Guide* for these steps.

Xenvironment

Xstore stores can be enrolled in the Xstore Office Cloud Service via Xenvironment by following these steps.

Note: Collect the following data prior to starting the Cloud Enroll process via Xenvironment.

1. Xstore Office Cloud Service hostname and port. The Customer Administrator can look this up by logging into their Cloud Service Account.
2. IDCS User credentials: Username and password of any IDCS user belonging to the provisioned IDCS tenant.

---

1. Once the Xenvironment installation is complete, open a web browser (from any system in the store) and go to the following URL: `https://<lead_register_hostname>:9096/cloudenroll`.
2. Log in with the appropriate user. The user must have the `SYSTEM_ADMIN` security privilege. This is the same privilege required to execute secured functions in Xenvironment.
3. In the form that is presented, enter the Xcenter Application Server Settings for Xstore Office Cloud.
   a. Host: Xstore Office Cloud Hostname
   b. Port: Xstore Office Cloud Port
   c. Provisioning ID: Customer's Provisioning ID

d. Username: Username of an IDCS user (This is typically an email address)

e. Password: Password of an IDCS user

4. Click **Enroll Location**. This will validate the user credentials and enroll the location.

5. Once the enrollment is complete the systems will be restarted. When the registers start up again they will be configured for Xstore Office Cloud Service.

Creation of OAuth Clients in IDCS

OAuth Clients (also called Apps) are required in order to invoke REST Services exposed by Xstore Office.

Note: While OAuth Clients can be created via the IDCS User Interface, the resulting OAuth Clients do not have all the needed properties in order to be able to function accurately. Instead, follow the steps detailed below in order to create the OAuth Clients using the IDCS REST APIs.

Prerequisites

- It is very helpful to understand tools and terminologies such as Basic Auth, OAuth, curl, json and their usage.
- For example, knowing that OAuth uses Bearer Tokens in the HTTP Authorization Header whereas Basic Auth uses Base 64 encoded credentials will help you understand the commands below.
- Authorization Header for an OAuth Token would look like this: `"Authorization: Bearer <token>"`
- Authorization Header for a Basic Auth Token would look like this: `"Authorization: Basic <Base64_encode(client_id:client_secret)>"`

Note: 1. OAuth Clients are also called Apps. These terms are used interchangeably.

2. Be sure to use the correct App Client IDs and Client Secrets based on the environment. Typically, there should be one for PRD and one for UAT. They are typically called RGBU_XSTCS_PRD_XOFFICE or RGBU_XSTCS_UAT_XOFFICE. These steps will have to be repeated for each environment. Using the artifacts from one environment in another environment can lead to unexpected results.

3. The steps for each OAuth Client that is created below are all inclusive (so that they can be scripted). Therefore, it is possible that steps are duplicated between OAuth Clients.

Required Data

Collect the following data prior to creation of OAuth Clients.

1. IDCS_TENANT_HOST: The Customer Administrator can look this up by logging into their Cloud Service Account.

2. Xstore Office OAuth Client App ID: The Xstore Office App has a display name that is typically RGBU_XSTCS_**PRD**_XOFFICE or RGBU_XSTCS_**UAT**_XOFFICE.

Clicking on it will display the App Details. The entry called Application ID contains the App ID for this OAuth Client.

3. Xstore Office OAuth Client credentials: The Xstore Office App has a display name that is typically RGBU_XSTCS_**PRD**_XOFFICE or RGBU_XSTCS_**UAT**_XOFFICE. Clicking on it and then clicking the Configuration Tab will display the App's Client ID and Client Secret.
4. IDCS User credentials: Username and password of any IDCS user belonging to the provisioned IDCS tenant. Typically, the user performing the particular function for which these OAuth Clients are needed would be used.

Tools

The following steps are executed using curl. However, any similar tool such as SoapUI or Postman can be used.

Xstore Office Setup App

Before the newly provisioned Xstore Office Cloud Service can be used, some initial setup is required. For instance, the Xstore Office database needs Tax Location data to be present in order to be able to setup new stores or organization hierarchy via the Xadmin UI. This can be achieved by using the Xcenter auto deployment functionality via REST services. In order to utilize the Xcenter REST services, an OAuth Client is required. This client can also be used to insert any other seed Xstore Office data that needs to be present in the database besides the Tax Location data.

Note:

1. This OAuth Client can also be used to configure the Data Migration Utility. Or a separate one can be created as described in the Data Migration section.
2. This OAuth Client can also be used for any additional recurring customer operations where Xcenter REST Services need to be invoked.

1. Request an Access token using the Xstore Office OAuth Client credentials.

```
Replace the <client_id> and <client_secret> with those of the Xstore Office
OAuth Client (App).
curl -i -H "Authorization: Basic <Base64_encode(<client_id>:<client_secret>)>"
-H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8"
https://<IDCS_TENANT_HOST>/oauth2/v1/token -d "grant_type=client_
credentials&scope=urn:opc:idm:__myscopes__"
```

2. Create a Setup OAuth Client (App). The response will contain the Application ID for the Setup App. Save this value since it will be required in a later step.

```
For PRD environment:
curl -i -H "Content-Type:application/scim+json" -H "Authorization: Bearer
<token>" https://<IDCS_TENANT_HOST>/admin/v1/Apps -d @\temp\SetupPRDApp.json

Copy the following contents into a file called SetupPRDApp.json and place it in
\temp (for example).

SetupPRDApp.json
{
    "displayName": "RGBU_XSTCS_PRD_Setup",
    "name": "RGBU_XSTCS_PRD_Setup_APPID",
    "description": "RGBU XSTCS Setup for PRD",
```

```
    "isOAuthClient": true,
    "clientType": "confidential",
    "isOAuthResource": false,
    "allowedGrants": [
        "client_credentials"
    ],
    "allowedOperations": [
        "introspect"
    ],
    "allowOffline": true,
    "allowAccessControl": false,
    "basedOnTemplate": {
        "value": "OPCAppTemplateId"
    },

"urn:ietf:params:scim:schemas:oracle:idcs:extension:opcService:App:serviceInsta
nceIdentifier": "RGBU_XSTCS_PRD_Setup",
    "active": true,
    "schemas": [
        "urn:ietf:params:scim:schemas:oracle:idcs:App"
    ],
    "isUnmanagedApp": true
}

For UAT environment:

curl -i -H "Content-Type:application/scim+json" -H "Authorization: Bearer
<token>" https://<IDCS_TENANT_HOST>/admin/v1/Apps -d @\temp\SetupUATApp.json

Copy the following contents into a file called SetupUATApp.json and place it in
\temp (for example).

SetupUATApp.json
{
    "displayName": "RGBU_XSTCS_UAT_Setup",
    "name": "RGBU_XSTCS_UAT_Setup_APPID",
    "description": "RGBU XSTCS Setup for UAT",
    "isOAuthClient": true,
    "clientType": "confidential",
    "isOAuthResource": false,
    "allowedGrants": [
        "client_credentials"
    ],
    "allowedOperations": [
        "introspect"
    ],
    "allowOffline": true,
    "allowAccessControl": false,
    "basedOnTemplate": {
        "value": "OPCAppTemplateId"
    },

"urn:ietf:params:scim:schemas:oracle:idcs:extension:opcService:App:serviceInsta
nceIdentifier": "RGBU_XSTCS_UAT_Setup",
    "active": true,
    "schemas": [
        "urn:ietf:params:scim:schemas:oracle:idcs:App"
    ],
    "isUnmanagedApp": true
}
```

3. Create a *Service Access* AppRole in the Setup App.

```
Prior to running this command, update the CreateServiceAccessAppRole.json to
replace <App_Id> with the value of the Application Id for the Setup App that
was saved previously.

curl -i -H "Content-Type:application/scim+json" -H "Authorization: Bearer
<token>" https://<IDCS_TENANT_HOST>/admin/v1/AppRoles -d
@\temp\CreateServiceAccessAppRole.json

Copy the following contents into a file called CreateServiceAccessAppRole.json
and place it in \temp (for example).
Replace <App_Id> with the value of the Application Id for the Setup App that
was saved previously.

CreateServiceAccessAppRole.json
{
    "displayName": "Service Access",
    "adminRole": true,
    "description": "Service Access AppRole",
    "public": false,
    "app": {
        "value": "<App_Id>"
    },
    "availableToClients": true,
    "availableToUsers": true,
    "availableToGroups": true,
    "schemas": ["urn:ietf:params:scim:schemas:oracle:idcs:AppRole"]
}
```

4. (If not already retrieved via IDCS) Retrieve the Xstore Office App ID.

```
This can also be obtained by logging into IDCS and clicking on the appropriate
App (typically RGBU_XSTCS_PRD_XOFFICE or RGBU_XSTCS_UAT_XOFFICE). Clicking on
it will display the App Details. The entry called Application Id contains the
App Id for this OAuth Client.

Replace the <client_id> with that of the Xstore Office OAuth Client (App).

curl -i -H "Authorization: Bearer <token>" "https://<IDCS_TENANT_
HOST>/admin/v1/Apps?filter=name+eq+%22<client_id>%22&attributes=id"
```

5. Retrieve the Xstore Office *Xstore Access* AppRole ID.

```
Replace the <App_Id> with the Application Id of the Xstore Office App.

curl -i -H "Authorization: Bearer <token>" "https://<IDCS_TENANT_
HOST>/admin/v1/AppRoles?filter=app.value+eq+%22<App_
Id>%22+and+displayName+eq+%22Xstore%20Access%22&attributes=id"
```

6. Grant the Xstore Office *Xstore Access* AppRole to the Setup OAuth Client (App).

```
curl -i -H "Content-Type:application/scim+json" -H "Authorization: Bearer
<token>" https://<IDCS_TENANT_HOST>/admin/v1/Grants -d
@\temp\XofficeAppRoleToGranteeAppGrant.json

Copy the following contents into a file called
XofficeAppRoleToGranteeAppGrant.json and place it in \temp (for example).
Replace the <Xstore_Office_App_Id> with the Xstore Office Application Id value.
Replace the <AppRole_Id> with the "Xstore Access" AppRole Id value.
Replace the <Grantee_App_Id> with the Setup OAuth Client (App) Application Id
value.

XofficeAppRoleToGranteeAppGrant.json
{
```

```
    "app": {
        "value": "<Xstore_Office_App_Id>"
    },
    "entitlement": {
        "attributeName": "appRoles",
        "attributeValue": "<AppRole_Id>"
    },
    "grantMechanism": "ADMINISTRATOR_TO_APP",
    "grantee": {
        "value": "<Grantee_App_Id>",
        "type": "App"
    },
    "isFulfilled": true,
    "schemas": [
        "urn:ietf:params:scim:schemas:oracle:idcs:Grant"
    ]
}
```

7. Request an Access token using the Setup OAuth Client credentials.

```
Replace the <client_id> and <client_secret> with those of the Setup OAuth
Client (App).

curl -i -H "Authorization: Basic <Base64_encode(<client_id>:<client_secret>)>"
-H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8"
https://<IDCS_TENANT_HOST>/oauth2/v1/token -d "grant_type=client_
credentials&scope=urn:opc:idm:__myscopes__"
```

8. This token can now be used to invoke Xstore Office REST APIs in order to configure Xstore Office Cloud Service.

Xstore Office Data Migration App

The Data Migration OAuth Client is required to configure the Data Migration Utility in order to migrate data from an existing Xstore Office to Xstore Office Cloud Service.

1. Request an Access token using the Xstore Office OAuth Client credentials.

```
Replace the <client_id> and <client_secret> with those of the Xstore Office
OAuth Client (App).

curl -i -H "Authorization: Basic <Base64_encode(<client_id>:<client_secret>)>"
-H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8"
https://<IDCS_TENANT_HOST>/oauth2/v1/token -d "grant_type=client_
credentials&scope=urn:opc:idm:__myscopes__"
```

2. Create a Data Migration OAuth Client (App). The response will contain the Application ID for the Data Migration App. Save this value since it will be required in a later step.

```
For PRD environment:

curl -i -H "Content-Type:application/scim+json" -H "Authorization: Bearer
<token>" https://<IDCS_TENANT_HOST>/admin/v1/Apps -d
@\temp\DataMigrationPRDApp.json

Copy the following contents into a file called DataMigrationPRDApp.json and
place it in \temp (for example).

DataMigrationPRDApp.json
{
    "displayName": "RGBU_XSTCS_PRD_Data_Migration",
    "name": "RGBU_XSTCS_PRD_Data_Migration_APPID",
    "description": "RGBU XSTCS Data Migration for PRD",
```

```
    "isOAuthClient": true,
    "clientType": "confidential",
    "isOAuthResource": false,
    "allowedGrants": [
        "client_credentials"
    ],
    "allowedOperations": [
        "introspect"
    ],
    "allowOffline": true,
    "allowAccessControl": false,
    "basedOnTemplate": {
        "value": "OPCAppTemplateId"
    },

"urn:ietf:params:scim:schemas:oracle:idcs:extension:opcService:App:serviceInsta
nceIdentifier": "RGBU_XSTCS_PRD_Data_Migration",
    "active": true,
    "schemas": [
        "urn:ietf:params:scim:schemas:oracle:idcs:App"
    ],
    "isUnmanagedApp": true
}
For UAT environment:

curl -i -H "Content-Type:application/scim+json" -H "Authorization: Bearer
<token>" https://<IDCS_TENANT_HOST>/admin/v1/Apps -d
@\temp\DataMigrationUATApp.json

Copy the following contents into a file called DataMigrationUATApp.json and
place it in \temp (for example).

DataMigrationUATApp.json
{
    "displayName": "RGBU_XSTCS_UAT_Data_Migration",
    "name": "RGBU_XSTCS_UAT_Data_Migration_APPID",
    "description": "RGBU XSTCS Data Migration for UAT",
    "isOAuthClient": true,
    "clientType": "confidential",
    "isOAuthResource": false,
    "allowedGrants": [
        "client_credentials"
    ],
    "allowedOperations": [
        "introspect"
    ],
    "allowOffline": true,
    "allowAccessControl": false,
    "basedOnTemplate": {
        "value": "OPCAppTemplateId"
    },

"urn:ietf:params:scim:schemas:oracle:idcs:extension:opcService:App:serviceInsta
nceIdentifier": "RGBU_XSTCS_UAT_Data_Migration",
    "active": true,
    "schemas": [
        "urn:ietf:params:scim:schemas:oracle:idcs:App"
    ],
    "isUnmanagedApp": true
}
```

3. Create a *Service Access* AppRole in the Data Migration App.

```
Prior to running this command, update the CreateServiceAccessAppRole.json to
replace <App_Id> with the value of the Application Id for the Data Migration
App that was saved previously.

curl -i -H "Content-Type:application/scim+json" -H "Authorization: Bearer
<token>" https://<IDCS_TENANT_HOST>/admin/v1/AppRoles -d
@\temp\CreateServiceAccessAppRole.json

Copy the following contents into a file called CreateServiceAccessAppRole.json
and place it in \temp (for example).
Replace <App_Id> with the value of the Application Id for the Data Migration
App that was saved previously.

CreateServiceAccessAppRole.json
{
    "displayName": "Service Access",
    "adminRole": true,
    "description": "Service Access AppRole",
    "public": false,
    "app": {
        "value": "<App_Id>"
    },
    "availableToClients": true,
    "availableToUsers": true,
    "availableToGroups": true,
    "schemas": ["urn:ietf:params:scim:schemas:oracle:idcs:AppRole"]
}
```

4. (If not already retrieved via IDCS) Retrieve the Xstore Office App ID.

```
This can also be obtained by logging into IDCS and clicking on the appropriate
App (typically RGBU_XSTCS_PRD_XOFFICE or RGBU_XSTCS_UAT_XOFFICE). Clicking on
it will display the App Details. The entry called Application Id contains the
App Id for this OAuth Client.

Replace the <client_id> with that of the Xstore Office OAuth Client (App).

curl -i -H "Authorization: Bearer <token>" "https://<IDCS_TENANT_
HOST>/admin/v1/Apps?filter=name+eq+%22<client_id>%22&attributes=id"
```

5. Retrieve the Xstore Office *Xstore Access* AppRole ID.

```
Replace the <App_Id> with the Application Id of the Xstore Office App.

curl -i -H "Authorization: Bearer <token>" "https://<IDCS_TENANT_
HOST>/admin/v1/AppRoles?filter=app.value+eq+%22<App_
Id>%22+and+displayName+eq+%22Xstore%20Access%22&attributes=id"
```

6. Grant the Xstore Office *Xstore Access* AppRole to the Data Migration OAuth Client (App).

```
curl -i -H "Content-Type:application/scim+json" -H "Authorization: Bearer
<token>" https://<IDCS_TENANT_HOST>/admin/v1/Grants -d
@\temp\XofficeAppRoleToGranteeAppGrant.json

Copy the following contents into a file called
XofficeAppRoleToGranteeAppGrant.json and place it in \temp (for example).
Replace the <Xstore_Office_App_Id> with the Xstore Office Application Id value.
Replace the <AppRole_Id> with the "Xstore Access" AppRole Id value.
Replace the <Grantee_App_Id> with the Data Migration OAuth Client (App)
Application Id value.

XofficeAppRoleToGranteeAppGrant.json
```

```
{
    "app": {
        "value": "<Xstore_Office_App_Id>"
    },
    "entitlement": {
        "attributeName": "appRoles",
        "attributeValue": "<AppRole_Id>"
    },
    "grantMechanism": "ADMINISTRATOR_TO_APP",
    "grantee": {
        "value": "<Grantee_App_Id>",
        "type": "App"
    },
    "isFulfilled": true,
    "schemas": [
        "urn:ietf:params:scim:schemas:oracle:idcs:Grant"
    ]
}
```

7. The Client ID and Client Secret of the Data Migration OAuth Client (App) can now be used to update the `idp.properties` in the Data Migration Utility and the Data Migration Utility is ready for use.

Xstore Office Data Privacy App

The Data Privacy OAuth Client is required in order to be able to invoke the Data Privacy REST API.

> **Note:** While some OAuth Clients can be used for different purposes (for instance, the same OAuth Client can be used for both the initial Cloud Setup as well as configuring the Data Migration Utility), it is important to remember that those cannot be used to invoke the Data Privacy REST API. The Data Privacy OAuth Client must be created as described below due to specific Data Privacy related authorizations that are performed.

1. Request an Access token using the Xstore Office OAuth Client credentials.

```
Replace the <client_id> and <client_secret> with those of the Xstore Office
OAuth Client (App).

curl -i -H "Authorization: Basic <Base64_encode(<client_id>:<client_secret>)>"
-H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8"
https://<IDCS_TENANT_HOST>/oauth2/v1/token -d "grant_type=client_
credentials&scope=urn:opc:idm:__myscopes__"
```

2. Create a Data Privacy OAuth Client (App). The response will contain the Application ID for the Data Privacy App. Save this value since it will be required in a later step.

```
For PRD environment:

curl -i -H "Content-Type:application/scim+json" -H "Authorization: Bearer
<token>" https://<IDCS_TENANT_HOST>/admin/v1/Apps -d
@\temp\DataPrivacyPRDApp.json

Copy the following contents into a file called DataPrivacyPRDApp.json and place
it in \temp (for example).

DataPrivacyPRDApp.json
```

```
{
    "displayName": "RGBU_XSTCS_PRD_Data_Privacy",
    "name": "RGBU_XSTCS_PRD_Data_Privacy_APPID",
    "description": "RGBU XSTCS Data Privacy for PRD",
    "isOAuthClient": true,
    "clientType": "confidential",
    "isOAuthResource": false,
    "allowedGrants": [
        "client_credentials",
        "password"
    ],
    "allowedOperations": [
        "introspect"
    ],
    "allowOffline": true,
    "allowAccessControl": false,
    "basedOnTemplate": {
        "value": "OPCAppTemplateId"
    },

"urn:ietf:params:scim:schemas:oracle:idcs:extension:opcService:App:serviceInsta
nceIdentifier": "RGBU_XSTCS_PRD_Data_Privacy",
    "active": true,
    "schemas": [
        "urn:ietf:params:scim:schemas:oracle:idcs:App"
    ],
    "isUnmanagedApp": true
}
For UAT environment:

curl -i -H "Content-Type:application/scim+json" -H "Authorization: Bearer
<token>" https://<IDCS_TENANT_HOST>/admin/v1/Apps -d
@\temp\DataPrivacyUATApp.json

Copy the following contents into a file called DataPrivacyUATApp.json and place
it in \temp (for example).

DataPrivacyUATApp.json
{
    "displayName": "RGBU_XSTCS_UAT_Data_Privacy",
    "name": "RGBU_XSTCS_UAT_Data_Privacy_APPID",
    "description": "RGBU XSTCS Data Privacy for UAT",
    "isOAuthClient": true,
    "clientType": "confidential",
    "isOAuthResource": false,
    "allowedGrants": [
        "client_credentials",
        "password"
    ],
    "allowedOperations": [
        "introspect"
    ],
    "allowOffline": true,
    "allowAccessControl": false,
    "basedOnTemplate": {
        "value": "OPCAppTemplateId"
    },

"urn:ietf:params:scim:schemas:oracle:idcs:extension:opcService:App:serviceInsta
nceIdentifier": "RGBU_XSTCS_UAT_Data_Privacy",
```

```
    "active": true,
    "schemas": [
        "urn:ietf:params:scim:schemas:oracle:idcs:App"
    ],
    "isUnmanagedApp": true
}
```

3. Create a *Service Access* AppRole in the Data Privacy App.

```
Prior to running this command, update the CreateServiceAccessAppRole.json to
replace <App_Id> with the value of the Application Id for the Data Privacy App
that was saved previously.

curl -i -H "Content-Type:application/scim+json" -H "Authorization: Bearer
<token>" https://<IDCS_TENANT_HOST>/admin/v1/AppRoles -d
@\temp\CreateServiceAccessAppRole.json

Copy the following contents into a file called CreateServiceAccessAppRole.json
and place it in \temp (for example).
Replace <App_Id> with the value of the Application Id for the Data Privacy App
that was saved previously.

CreateServiceAccessAppRole.json
{
    "displayName": "Service Access",
    "adminRole": true,
    "description": "Service Access AppRole",
    "public": false,
    "app": {
        "value": "<App_Id>"
    },
    "availableToClients": true,
    "availableToUsers": true,
    "availableToGroups": true,
    "schemas": ["urn:ietf:params:scim:schemas:oracle:idcs:AppRole"]
}
```

4. (If not already retrieved via IDCS) Retrieve the Xstore Office App ID.

```
This can also be obtained by logging into IDCS and clicking on the appropriate
App (typically RGBU_XSTCS_PRD_XOFFICE or RGBU_XSTCS_UAT_XOFFICE). Clicking on
it will display the App Details. The entry called Application Id contains the
App Id for this OAuth Client.

Replace the <client_id> with that of the Xstore Office OAuth Client (App).

curl -i -H "Authorization: Bearer <token>" "https://<IDCS_TENANT_
HOST>/admin/v1/Apps?filter=name+eq+%22<client_id>%22&attributes=id"
```

5. Retrieve the Xstore Office *Data Privacy Access* AppRole ID.

```
Replace the <App_Id> with the Application Id of the Xstore Office App.

curl -i -H "Authorization: Bearer <token>" "https://<IDCS_TENANT_
HOST>/admin/v1/AppRoles?filter=app.value+eq+%22<App_
Id>%22+and+displayName+eq+%22Data%20Privacy%20Access%22&attributes=id"
```

6. Grant the Xstore Office *Data Privacy Access* AppRole to the Data Privacy OAuth Client (App).

```
curl -i -H "Content-Type:application/scim+json" -H "Authorization: Bearer
<token>" https://<IDCS_TENANT_HOST>/admin/v1/Grants -d
@\temp\XofficeAppRoleToGranteeAppGrant.json

Copy the following contents into a file called
XofficeAppRoleToGranteeAppGrant.json and place it in \temp (for example).
```

```
Replace the <Xstore_Office_App_Id> with the Xstore Office Application Id value.
Replace the <AppRole_Id> with the "Data Privacy Access" AppRole Id value.
Replace the <Grantee_App_Id> with the Data Privacy OAuth Client (App)
Application Id value.

XofficeAppRoleToGranteeAppGrant.json
{
    "app": {
        "value": "<Xstore_Office_App_Id>"
    },
    "entitlement": {
        "attributeName": "appRoles",
        "attributeValue": "<AppRole_Id>"
    },
    "grantMechanism": "ADMINISTRATOR_TO_APP",
    "grantee": {
        "value": "<Grantee_App_Id>",
        "type": "App"
    },
    "isFulfilled": true,
    "schemas": [
        "urn:ietf:params:scim:schemas:oracle:idcs:Grant"
    ]
}
```

7. Request an Access token using the Data Privacy OAuth Client credentials and an IDCS user's userid/password.

```
Replace the <client_id> and <client_secret> with those of the Data Privacy
OAuth Client (App).
Replace the <IDCS_username> and <IDCS_password> with those of an IDCS user.

curl -i -H "Authorization: Basic <Base64_encode(<client_id>:<client_secret>)>"
-H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8"
https://<IDCS_TENANT_HOST>/oauth2/v1/token -d "grant_
type=password&username=<IDCS_username>&password=<IDCS_
password>&scope=urn:opc:idm:__myscopes__"
```

8. Invoke the Data Privacy endpoint (example – replace with appropriate data).

```
Replace <token> with the token from the previous step.

curl -i -H "Authorization: Bearer <token>" "https://<XSTORE_OFFICE_
HOST>/xcenter/rest/privatedata/1000::100?type=employee"
```

RTLog Generator Client App

The steps below create an OAuth Client that can be used to invoke RTLog Generator REST APIs.

1. Request an Access token using the Xstore Office OAuth Client credentials.

```
Replace the <client_id> and <client_secret> with those of the Xstore Office
OAuth Client (App).

curl -i -H "Authorization: Basic <Base64_encode(<client_id>:<client_secret>)>"
-H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8"
https://<IDCS_TENANT_HOST>/oauth2/v1/token -d "grant_type=client_
credentials&scope=urn:opc:idm:__myscopes__"
```

2. Create a RTLog Generator Client OAuth Client (App). The response will contain the Application ID for the RTLog Generator Client App. Save this value since it will be required in a later step.

```
For PRD environment:

curl -i -H "Content-Type:application/scim+json" -H "Authorization: Bearer
<token>" https://<IDCS_TENANT_HOST>/admin/v1/Apps -d
@\temp\RTLogClientPRDApp.json

Copy the following contents into a file called RTLogClientPRDApp.json and place
it in \temp (for example).

RTLogClientPRDApp.json
{
    "displayName": "RGBU_XSTCS_PRD_RTLog_Client",
    "name": "RGBU_XSTCS_PRD_RTLog_Client_APPID",
    "description": "RGBU XSTCS RTLog_Client for PRD",
    "isOAuthClient": true,
    "clientType": "confidential",
    "isOAuthResource": false,
    "allowedGrants": [
        "client_credentials"
    ],
    "allowedOperations": [
        "introspect"
    ],
    "allowOffline": true,
    "allowAccessControl": false,
    "basedOnTemplate": {
        "value": "OPCAppTemplateId"
    },

"urn:ietf:params:scim:schemas:oracle:idcs:extension:opcService:App:serviceInsta
nceIdentifier": "RGBU_XSTCS_PRD_RTLog_Client",
    "active": true,
    "schemas": [
        "urn:ietf:params:scim:schemas:oracle:idcs:App"
    ],
    "isUnmanagedApp": true
}

For UAT environment:

curl -i -H "Content-Type:application/scim+json" -H "Authorization: Bearer
<token>" https://<IDCS_TENANT_HOST>/admin/v1/Apps -d
@\temp\RTLogClientUATApp.json

Copy the following contents into a file called RTLogClientUATApp.json and place
it in \temp (for example).

RTLogClientUATApp.json
{
    "displayName": "RGBU_XSTCS_UAT_RTLog_Client",
    "name": "RGBU_XSTCS_UAT_RTLog_Client_APPID",
    "description": "RGBU XSTCS RTLog_Client for UAT",
    "isOAuthClient": true,
    "clientType": "confidential",
    "isOAuthResource": false,
    "allowedGrants": [
        "client_credentials"
    ],

    "allowedOperations": [
```

```
            "introspect"
        ],
        "allowOffline": true,
        "allowAccessControl": false,
        "basedOnTemplate": {
            "value": "OPCAppTemplateId"
        },

"urn:ietf:params:scim:schemas:oracle:idcs:extension:opcService:App:serviceInsta
nceIdentifier": "RGBU_XSTCS_UAT_RTLog_Client",
        "active": true,
        "schemas": [
            "urn:ietf:params:scim:schemas:oracle:idcs:App"
        ],
        "isUnmanagedApp": true
}
```

3. Create a *Service Access* AppRole in the Setup App.

```
Prior to running this command, update the CreateServiceAccessAppRole.json to
replace <App_Id> with the value of the Application Id for the RTLog Generator
Client App that was saved previously.

curl -i -H "Content-Type:application/scim+json" -H "Authorization: Bearer
<token>" https://<IDCS_TENANT_HOST>/admin/v1/AppRoles -d
@\temp\CreateServiceAccessAppRole.json

Copy the following contents into a file called CreateServiceAccessAppRole.json
and place it in \temp (for example).
Replace <App_Id> with the value of the Application Id for the RTLog Generator
Client App that was saved previously.

CreateServiceAccessAppRole.json
{
    "displayName": "Service Access",
    "adminRole": true,
    "description": "Service Access AppRole",
    "public": false,
    "app": {
        "value": "<App_Id>"
    },
    "availableToClients": true,
    "availableToUsers": true,
    "availableToGroups": true,
    "schemas": ["urn:ietf:params:scim:schemas:oracle:idcs:AppRole"]
}
```

4. The Client ID and Client Secret of the RTLog Generator Client OAuth Client (App) can now be used to obtain a token in order to invoke RTLog Generator REST APIs.

Additional Apps

Additional OAuth Clients can be created if required.

- For invoking Xstore Office REST APIs, please follow the steps as mentioned in the Data Migration OAuth Client section (and make any necessary changes to keep the IDs unique).
- For invoking RTLog REST APIs, follow the steps as mentioned in the RTLog Generator Client section (and make any necessary changes to keep the IDs unique).

Deletion of OAuth Clients in IDCS

In order to delete OAuth Clients in IDCS, follow this process:

1. Log into IDCS.
2. Select the App to be deleted.
3. Click on the Application Roles tab.
4. Select the menu icon on the far right of the *Service Access* AppRole.
5. Select **Assign Users** and select the current user in the popup to be granted this AppRole.
6. Then go back to the list of Applications, select the menu icon to the far right of the App to be deleted. Then click **Deactivate** and click **Deactivate Application** in the pop up dialog.
7. Once again select the menu icon to the far right of the App to be deleted and click **Remove** and click **Remove Application** in the pop up dialog.