Configuration Guide

DS.OUD REST Configuration

Contents

[Configuring the OUD Environment for Data Management REST API 1](#_Toc184198544)

[Roadmap 1](#_Toc184198545)

[Prerequisites 1](#_Toc184198546)

[Resources 1](#_Toc184198547)

[Preparations 1](#_Toc184198548)

[Configuring Connection Handlers for an Existing OUD Instance 2](#_Toc184198549)

[Verify connection handlers 2](#_Toc184198550)

[Create connection handler 2](#_Toc184198551)

[Enable connection handler 3](#_Toc184198552)

[Configure REST endpoints 4](#_Toc184198553)

[Enable the REST Server extension 4](#_Toc184198554)

[Enable the directory endpoint 5](#_Toc184198555)

[Testing 5](#_Toc184198556)

[Access Control 6](#_Toc184198557)

[ACI Template 6](#_Toc184198558)

[Import ACI 7](#_Toc184198559)

[ACI ZIMP 7](#_Toc184198560)

[Drop password file 8](#_Toc184198561)

## Configuring the OUD Environment for Data Management REST API

OUD environment must be configured to use the Oracle Unified Directory Data Management REST API.

### Roadmap

The instance is configured on each node according to the following procedure:

* Preparations
* Configuring HTTP Connection Handlers
* Configure REST endpoints
* Testing and cleanup

### Prerequisites

An OUD instance must exist.

### Resources

Managing Directory Data Using Data Management REST API

<https://docs.oracle.com/en/middleware/idm/unified-directory/12.2.1.4/oudag/managing-directory-data-using-data-management-rest-api.html#GUID-7DCC4DFE-19E7-496F-B06B-B3EE34F45264>

Using JKS Key Manager Provider

<https://docs.oracle.com/en/middleware/idm/unified-directory/12.2.1.4/oudag/configuring-security-clients-and-servers.html#GUID-8A2E7CCD-0E6F-4604-9745-922C4EB60789>

Using the JKS Trust Manager Provider

<https://docs.oracle.com/en/middleware/idm/unified-directory/12.2.1.4/oudag/configuring-security-clients-and-servers.html#GUID-A1CCCC44-D418-46F3-8E5E-F99FB46AAF8F>

### Preparations

The password of the directory administrator must be stored in a file which can be used as a reference during the configuration.

|  | **Command** |
| --- | --- |
| echo *my-password* > /path/to/my/password/file/my-password-file.txt  echo Password123 > .password |

|  | **Important** |
| --- | --- |
| The password must belong to the directory administrator who will be used during the configuration like "cn=Directory Manager". |

| D:\tt\icon-warning-16.png | **Warning** |
| --- | --- |
| Do not forget to clean up the password file after the configuration! |

##### Parameters to use during the configuration:

| **Name** | **Value** |
| --- | --- |
| **Hostname** | The hostname of the oud server, e.g.: IAM-DEV-STG1010.pap.bka.bund.de |
| **Port** | The administration port of the oud instance, e.g.: 7441 |
| **Username** | Directory administrator, e.g.: "cn=Directory Manager" |
| **Password-file** | Path of the file created to store the admin password what belongs to the Directory administrator. E.g.: .password |

Store the parameter values in shell variables to reuse them during the configuration:

|  | **Command** |
| --- | --- |
| OUD\_HOSTNAME=IAM-DEV-STG1010.pap.bka.bund.de  OUD\_ADMIN\_PORT=7441  OUD\_ADMIN\_USER="cn=Directory Manager"  OUD\_ADMIN\_PASSWD=.password |

### Configuring Connection Handlers for an Existing OUD Instance

#### Verify connection handlers

First it should be checked if any handlers are already exposed:

|  | **Command** |
| --- | --- |
| ./dsconfig list-connection-handlers \  --hostname $OUD\_HOSTNAME \  --port $OUD\_ADMIN\_PORT \  --portProtocol LDAP \  --bindDN "$OUD\_ADMIN\_USER" \  --bindPasswordFile $OUD\_ADMIN\_PASSWD \  --no-prompt |

The result should look like:

|  | **Result** |
| --- | --- |
| Connection Handler : Type : enabled : listen-port : use-ssl  -------------------------:------:---------:-------------:--------  HTTP Connection Handler : http : false : 8080 : false  HTTPS Connection Handler : http : false : 10443 : true  JMX Connection Handler : jmx : false : 1689 : false  LDAP Connection Handler : ldap : true : 1389 : false  LDAPS Connection Handler : ldap : true : 1636 : true  LDIF Connection Handler : ldif : false : - : -  SNMP Connection Handler : snmp : false : 161 : - |

If the result doesn’t contain HTTP/HTTPS Connection Handler(s), then handler(s) must be created: [Create connection handler](#_Create_connection_handler)

If the result contains HTTP/HTTPS Connection Handler(s), but the enabled flag is false, then the handler(s) must be enabled: [Enable connection handler](#_Enable_connection_handler)

If the result contains HTTP/HTTPS Connection Handler(s) and the enabled flag is true, then this step can be skipped, continue here: [Configure the REST endpoints](#_Configure_the_REST)

#### 

#### Create connection handler

Create only that handler what is required, it is not necessary to create both.

##### HTTP Connection handler

##### Parameter

| **Name** | **Meaning** |
| --- | --- |
| **http-listening-port** | The port to expose for HTTP communication |

|  | **Command** |
| --- | --- |
| ./dsconfig create-connection-handler \  --handler-name "HTTP Connection Handler" \  --type http \  --set enabled:true \  --set listen-port:{http-listening-port} \  --hostname $OUD\_HOSTNAME \  --port $OUD\_ADMIN\_PORT \  --portProtocol LDAP \  --bindDN "$OUD\_ADMIN\_USER" \  --bindPasswordFile $OUD\_ADMIN\_PASSWD \  --no-prompt |

##### HTTPS Connector handler

##### Parameter

| **Name** | **Meaning** |
| --- | --- |
| **https-listening-port** | The port to expose for HTTPS communication |

|  | **Command** |
| --- | --- |
| ./dsconfig create-connection-handler \  --handler-name "HTTPS Connection Handler" \  --type http \  --set enabled:true \  --set listen-port:{https-listening-port} \  --set use-ssl:true \  --set trust-manager-provider:JKS \  --set key-manager-provider:JKS \  --hostname $OUD\_HOSTNAME \  --port $OUD\_ADMIN\_PORT \  --portProtocol LDAP \  --bindDN "$OUD\_ADMIN\_USER" \  --bindPasswordFile $OUD\_ADMIN\_PASSWD \  --no-prompt |

Next step: [Configure the REST endpoints](#_Configure_the_REST)

#### Enable connection handler

This method can also be used to modify other parameters like listening port.

##### HTTP Connection handler

##### Parameter

| **Name** | **Meaning** |
| --- | --- |
| **http-listening-port** | The port to expose for HTTP communication |

|  | **Command** |
| --- | --- |
| ./dsconfig set-connection-handler-prop \  --handler-name "HTTP Connection Handler" \  --set enabled:true \  --set listen-port:{http-listening-port} \  --hostname $OUD\_HOSTNAME \  --port $OUD\_ADMIN\_PORT \  --portProtocol LDAP \  --bindDN "$OUD\_ADMIN\_USER" \  --bindPasswordFile $OUD\_ADMIN\_PASSWD \  --no-prompt |

##### HTTPS Connector handler

##### Parameter

| **Name** | **Meaning** |
| --- | --- |
| **https-listening-port** | The port to expose for HTTPS communication |

|  | **Command** |
| --- | --- |
| ./dsconfig set-connection-handler-prop \  --handler-name "HTTPS Connection Handler" \  --set enabled:true \  --set listen-port:{https-listening-port} \  --set use-ssl:true \  --set trust-manager-provider:JKS \  --set key-manager-provider:JKS \  --hostname $OUD\_HOSTNAME \  --port $OUD\_ADMIN\_PORT \  --portProtocol LDAP \  --bindDN "$OUD\_ADMIN\_USER" \  --bindPasswordFile $OUD\_ADMIN\_PASSWD \  --no-prompt |

Next step: [Configure the REST endpoints](#_Configure_the_REST)

### Configure REST endpoints

#### Enable the REST Server extension

|  | **Command** |
| --- | --- |
| ./dsconfig set-extension-prop \  --Extension-name "REST Server" \  --set enabled:true \  --trustAll \  --hostname $OUD\_HOSTNAME \  --port $OUD\_ADMIN\_PORT \  --portProtocol LDAP \  --bindDN "$OUD\_ADMIN\_USER" \  --bindPasswordFile $OUD\_ADMIN\_PASSWD \  --no-prompt |

If the REST Server extension setup fails with the following error:

The REST Server Extension could not be modified because of the following reason:

\* The End Point "Token Generator" referenced in property "end-point" is disabled

Then the Token Generator end point should be enabled first

|  | **Command** |
| --- | --- |
| ./dsconfig set-end-point-prop \  --point-name 'Token Generator' \  --set enabled:true \  --hostname $OUD\_HOSTNAME \  --port $OUD\_ADMIN\_PORT \  --portProtocol LDAP \  --trustAll \  --bindDN "$OUD\_ADMIN\_USER" \  --bindPasswordFile $OUD\_ADMIN\_PASSWD \  --no-prompt |

#### Enable the directory endpoint

|  | **Command** |
| --- | --- |
| ./dsconfig set-directory-end-point-prop \  --set enabled:true \  --hostname $OUD\_HOSTNAME \  --port $OUD\_ADMIN\_PORT \  --portProtocol LDAP \  --bindDN "$OUD\_ADMIN\_USER" \  --bindPasswordFile $OUD\_ADMIN\_PASSWD \  --trustAll \  --no-prompt |

| D:\tt\icon-note-16.png | **Note** |
| --- | --- |
| If the LDAPS connection handler is not configured for the OUD instance, then you must configure *cn=JKS,cn=Key Manager Providers,cn=config* and *cn=JKS,cn=Trust Manager Providers,cn=config* before you set the HTTPS connection handler. See Using JKS Key Manager Provider and Using the JKS Trust Manager Provider |

### Testing

When the configuration is done, the applied changes can be tested by executing a simple curl query.

##### Parameters

| **Name** | **Value** |
| --- | --- |
| **protocol** | http or https |
| **http-port** | The port that was exposed to the protocol |
| **dn** | The full DN of the directory entry to lookup |

|  | **Command** |
| --- | --- |
| curl -k -i \  -X GET \  -u {Username}:{Password} \  “{protocol}://{Hostname}:{http-port}/rest/v1/directory/{dn}” |

|  | **Command** |
| --- | --- |
| curl -k -i \  -X GET \  -u “cn=Directory Manager”: Password123 \  “https: IAM-DEV-STG1010.pap.bka.bund.de:1081/rest/v1/directory/ ou=Groups,ou=ZIMP,cn=Services,dc=bka,dc=bund,dc=de?scope=sub” |

### Access Control

The appropriate privileges must be granted to the application service users so they can manage their LDAP subtree and maintain the lifecycle of their entitlements.

It can be achieved by using Access Control Instructions (ACI).

The privileges to grant:

* Create groups
* Delete groups
* Read groups
* Modify the description attribute on groups

| D:\tt\icon-warning-16.png | **Warning** |
| --- | --- |
| The service users must exist to configure access controls. This document doesn’t provide any solution to create users. |

The access control can be set up in many ways like via OUDSM, here a command line tool solution will be presented.

#### ACI Template

ACI definitions can be imported as an LDIF file, what means an LDIF file must be created for each system. First create a file like *<system>-aci.ldif* copy the template code from below into the file and replace the parameters defined in the table:

| D:\tt\icon-warning-16.png | **Warning** |
| --- | --- |
| Be aware of the special rules of LDIF format! |

##### Parameters

| **Name** | **Value** |
| --- | --- |
| **\_\_APPLICATION\_\_** | The name of the application in the directory tree. |
| **\_\_ACI\_C\_R\_D \_NAME\_\_** | Name of the ACI what grants access to create, delete  and read data in the directory structure. |
| **\_\_ACI\_U \_NAME\_\_** | Name of the ACI what grants access modify data in the directory structure. |
| **\_\_SERVICEUSER\_\_** | The name of the service user in the directory tree. |

|  | **Command** |
| --- | --- |
| dn: ou=Groups,ou=\_\_APPLICATION\_\_,cn=Services,dc=bka,dc=bund,dc=de  changetype: modify  add: aci  aci: (target="ldap:///ou=Groups,ou=\_\_APPLICATION\_\_,cn=Services,dc=bka,dc=bund,dc=de")  (targetattr="\*")  (targetfilter="(objectClass=groupOfUniqueNames)")  (targetscope="onelevel")  (version 3.0; acl "\_\_ACI\_C\_R\_D\_NAME\_\_"; allow (delete,search,read,add)  userdn="ldap:///cn=\_\_SERVICEUSER\_\_,cn=System,dc=bka,dc=bund,dc=de";)  aci: (target="ldap:///ou=Groups,ou=\_\_APPLICATION\_\_,cn=Services,dc=bka,dc=bund,dc=de")  (targetattr="description")  (targetfilter="objectClass=groupOfUniqueNames")  (targetscope="onelevel")  (version 3.0; acl "\_\_ACI\_U\_NAME\_\_"; allow (write)  userdn="ldap:///cn=\_\_SERVICEUSER\_\_,cn=System,dc=bka,dc=bund,dc=de";) |

#### Import ACI

Steps to import ACI:

1. Make the ldif file available on the server where the OUD is hosted from (copy, scp…)
2. Navigate to the *bin* directory under OUD Instance home *(<OUD Instance>/bin)*
3. Import the ldif file by executing ldapmodify

##### Parameters

| **Name** | **Value** |
| --- | --- |
| **ldap-port** | Ldap port to connect. |
| **filename** | The path where the ldif file is located. |

|  | **Command** |
| --- | --- |
| ./ldapmodify \  -h $OUD\_HOSTNAME \  -p {ldap-port} \  -D "$OUD\_ADMIN\_USER" \  -j $OUD\_ADMIN\_PASSWD \  --filename ../path/to/aci/aci.ldif |

| D:\tt\icon-warning-16.png | **Warning** |
| --- | --- |
| Use the LDAP port here, not the admin port! |

#### ACI ZIMP

Let’s create ACI for ZIMP and import it as an example.

1. Create a file zimp-aci.ldif
2. Copy the command below into the file

|  | **Command** |
| --- | --- |
| dn: ou=Groups,ou=ZIMP,cn=Services,dc=bka,dc=bund,dc=de  changetype: modify  add: aci  aci: (target="ldap:///ou=Groups,ou=ZIMP,cn=Services,dc=bka,dc=bund,dc=de")  (targetattr="\*")  (targetfilter="(objectClass=groupOfUniqueNames)")  (targetscope="onelevel")  (version 3.0; acl "ZIMP CRD Access"; allow (delete,search,read,add)  userdn="ldap:///cn=zimpserviceuser,cn=System,dc=bka,dc=bund,dc=de";)  aci: (target="ldap:///ou=Groups,ou=ZIMP,cn=Services,dc=bka,dc=bund,dc=de")  (targetattr="description")  (targetfilter="objectClass=groupOfUniqueNames")  (targetscope="onelevel")  (version 3.0; acl "ZIMP U Access"; allow (write)  userdn="ldap:///cn=zimpserviceuser,cn=System,dc=bka,dc=bund,dc=de";) |

1. Scp the file to the OUD server (if it wasn’t created there)
2. Navigate to *<OUD\_Instance\_Home>/bin* and execute:

|  | **Command** |
| --- | --- |
| ./ldapmodify \  -h $OUD\_HOSTNAME \  -p {ldap-port} \  -D "$OUD\_ADMIN\_USER" \  -j $OUD\_ADMIN\_PASSWD \  --filename ../path/to/aci/zimp-aci.ldif |

1. The success of the process can be tested by re-executing the steps in [Testing](#_Testing) phase with the actual service user

### Drop password file

After the configuration is done the file which contains the password must be removed!

|  | **Command** |
| --- | --- |
| rm /path/to/my/password/file/my-password-file.txt  rm .password |