



Documentation

Installation and Configuration Guide

Ldap Rest API

Version: 1.1
created:: 11.05.2017
File: Ldap Rest API manual 1.docx

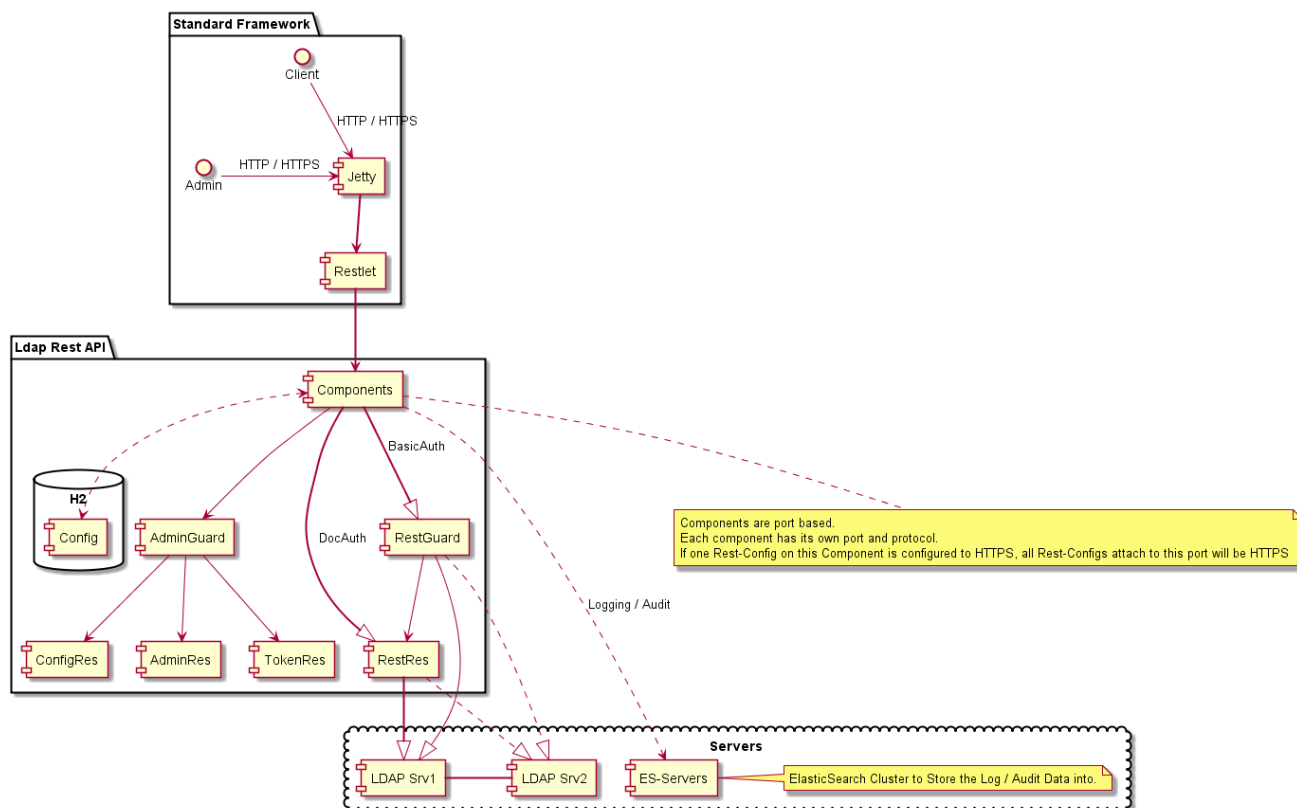
Contents

1	Service Design Overview	3
2	Installation	4
3	Startup Options / Recovery	6
4	Configuration	7
4.1	Configs	8
4.2	Scripting	15
	Scripts Folder	15
	Script Invoke.....	15
	Get Started	15
5	Service Flow Diagramme	16
6	Error Codes / Handling	18
7	Request / Response Documents.....	22

Installation and Configuration Guide

1 Service Design Overview

Components



Installation and Configuration Guide

2 Installation

1 To install the Ldap Rest API, go to our web site <http://www.skypro.ch/en/products> and download the complete installation file *LdapRestAPI.zip*.

This file contains all the needed components. After downloading, please unpack it.

2 Extract the ZIP file into a new Directory

3 Create a Start.sh / start.cmd or start.command file.

The command to start the service is:

```
java -jar eDirRest-*-jar-with-dependencies.jar
```

4 Start the service

Start the service via FireDaemon on Windows or any Wrapper available for your System.

5 First steps

Connect to the Service via <http://localhost:8182>

You should be presented a Page like this:

Read the Page, and do as stated there.

Installation and Configuration Guide

The Basic Configuration may look like:

Ldap

RestApi

Admin

Tokens

Configs

Services

Service / Object configurations

Load an already defined API Configuration or create a new one

Changes here need a restart of the Service.

Loadtesten

NewSaveValid

Service

Edit JSON

Config

testen

Name of this Configuration / Service

☒

Enable this Config

Check this to have this Service Loaded after restarts

Port the Service should start on

4567

the Service will start on this TCP Port

☐

Enable HTTPS encryption

If Checked, the Service will run as HTTPS. Caution: You need to provide a valid keystore file

☐

Enable Basic HTTP Authentication

If Checked, the Service will do Basic HTTP Authentication. If not Auth Block has to be provided in Requests

☒

Use preconfigured Tokens instead of Passwords

If Checked, the Server will accept Tokens, and Lookup matching Tokens in his List. And replace Token at Runtime with the Password

☒

Allow JSON Requests

Allow consumption of JSON Requests. If JSON is provided, response Documents will also be in JSON Format

☒

Allow XML Requests

Allow consumption of XML Requests. If XML is provided, response Documents will also be in XML Format

☒

Allow Partial Updates

If the Request would fail, the API tries to Apply as much as possible.

☒

HTTP Status Codes

The API Will provide HTML Status Codes on Top of the Status Element in the Response Documents

Ldap Servers

Use this IP address for LDAP Connection. (All Servers should contain the same data and need the same Serviceport)

IP

192.168.1.240

Delete

Add IPDeleteLast IP

Ldap Server Port

389

The LDAP Port we use to connect to the Servers. Default is 389 for unencrypted or TLSv1 connections. 636 for SSL (deprecated)

☐

Ldap Use SSL

Use SSL for LDAP. If possible use TLS

☒

Ldap Use TLS

Use TLSv1 for LDAP.

Users

Users that are allowed to use this Service (Full LDAP DN. Example: cn=myuser,ou=users,o=data)

UserDN

cn=admin,ou=s,a,o=system

Delete

Add UserDNDeleteLast UserDN

☐

Audit to Elasticsearch

Enable Auditing to an Index of an Elasticsearch Cluster

☐

Login to Elasticsearch

Enable Logging to Elasticsearch

Cluster Name

3 Startup Options / Recovery

If the Service does not start anymore, it may be due to the following two reasons.

- 1 One of the service has a Config, that does not start and is configured to run on the same port as the Adminserver
- 2 There is a keystore, that does not work with the given Configuration

To solve such problems, you can pass arguments to the service.

- You can move the AdminServer Port to an new unused port.
- You can disable HTTPS encryption for all the Configured Components / Services

Usage: java -jar LdapRestApi.jar

- h,--help print this message
- i,--insecure Force disable of HTTPS on the service
- p,--port <arg> Override port number of Admin Component

Installation and Configuration Guide

4 Configuration

Service configuration

After installation, you should be able to connect to <http://localhost:8182>

Go there to Admin and set Credentials for the Admin Gui of this service.

Admin Settings
Admin dependent Settings, like User/Password to protect this Admin-Panel

User Settings

Username: admin
Password: ****

Service Language

Language: English

Server Settings

Adminport: 4567
Enable SSL: ☐
Debug SSL: ☐

CAREFULL!! You need to provide a keystore File in Service root, else you cannot Access Service anymore
Debug All SSL/TLS Activity

Save

After Changes here you need to Restart the Service!!

After you set the Adminport, Language, Username and Password you should restart the service and Connect to the new port.

You will be prompted for Username / Password, as soon as you access relevant Data APIs.

Note: Admin GUI / Rest Services may run on the same port. For simplicity and security, we suggest you run this on a different port.

Keystore for HTTPS encryption

To create a Keystore file, you need the Java keytool or similar like OpenSSL.
The keystore file should be place in the root-directory of the Service (where the jar and db files are).

To create a Self-Signed keystore do:
keytool -keystore keystore -alias jetty -genkey -keyalg RSA

The Password for the keystore and the jetty server is "changeme" without " ".

Installation and Configuration Guide

For more information check:

http://wiki.eclipse.org/Jetty/Howto/Configure_SSL

<http://seppinho.github.io/restlet/webservice/2015/08/31/restlet/>

Cacerts for LDAP

For the LDAP encryption we need a cacerts file, including the CA for the Certificate used by the LDAP.

Make sure, both (the CA and the Cert) Certificates are valid.

Add the CA by the following command:

```
keytool -importcert -trustcacerts -file cert.b64 -keystore cacerts -alias ldapserver
```

The cert.b64 is the CA Public Key. You do not need any private Keys.

Use "changeme" as Password for the cacerts file.

4.1 Configs

In the Configs section we configure Service / Object configurations. These are used to define the REST APIs, we present to the User/Customer/Client.

A Config contains a Service Config which consists of:

- Port / Security Config
- LDAP Servers for Auth and Datamanipulation
- Users
- Logging
- Object configuration
- Object specific settings
- Attributes

Installation and Configuration Guide

Port / Security Config

Port the Service should start on

the Service will start on this TCP Port

☐ Enable HTTPS encryption

If Checked, the Service will run as HTTPS. Caution: You need to provide a valid keystore file

☐ Enable Basic HTTP Authentication

If Checked, the Service will do Basic HTTP Authentication. If not Auth Block has to be provided in Requests

☒ Use preconfigured Tokens instead of Passwords

If Checked, the Server will accept Tokens, and Lookup matching Tokens in his List. And replace Token at Runtime with the Password

☐ Allow JSON Requests

Allow consumption of JSON Requests. If JSON is provided, response Documents will also be in JSON Format

☒ Allow XML Requests

Allow consumption of XML Requests. If XML is provided, response Documents will also be in XML Format

☒ Allow Partial Updates

If the Request would fail, the API tries to Apply as much as possible.

☒ HTTP Status Codes

The API Will provide HTML Status Codes on Top of the Status Element in the Response Documents

Name	Good Startvalue	Description
Port the Service should start on	4567	The Port this Configs runs on. All Configs with the same port, run on the same component. All Services on a component run HTTP or HTTPS.
Enable HTTPS encryption	false	Run this Port as HTTPS? You need to create a keystore file in the service root folder. See documentation
Enable Basic HTTP Authentication	true/false	If checked, the Service requires Basic HTTP Authentication. If not, you have to pass Auth parameters in the Request document
Use preconfigured Tokens instead of Passwords	true/false	If checked, you need to provide a token instead of a password. the service will replace the token with password stored in the token section.
Allow JSON Requests	true	If checked, you can pass requests as JSON to the service. You will receive a JSON file as response. Use Content-Type: application/json
Allow XML Requests	true	If checked, you can pass requests as XML to the service. You will receive a XML file as response. Use Content-Type: application/xml
Create more Generic Instance Output	false	Instead of simple XML / JSON, you can enable this Option to get a more Generic Document. In and Out
Allow Partial Updates	true/false	If checked, the Service tries to apply as many changes as possible. It will split up the operation to its atoms, and try to apply every value separately. This happens only, if the Change fails as whole
HTTP Status Codes	true/false	If checked, the Status Codes provided in the answer documents will also be applied to the HTML Response.

Installation and Configuration Guide

LDAP Servers for Auth and Datamanipulation

Ldap Servers
Use this IP address for LDAP Connection. (All Servers should contain the same data and need the same Serviceport)

IP

192.168.1.240 [Delete](#)

192.168.1.241 [Delete](#)

[Add IP](#) [Delete Last IP](#) [Delete All](#)

Ldap Server Port
389 The LDAP Port we use to connect to the Servers. Default is 389 for unencrypted or TLSv1 connections. 636 for SSL (deprecated)

☐ **Ldap Use SSL**
Use SSL for LDAP. If possible use TLS

☐ **Ldap Use TLS**
Use TLSv1 for LDAP.

Name	Good Startvalue	Description
Ldap Servers	A List of your LDAP Servers	Add the IPs / DNS Names here of your LDAP server. The Rest Service will Auth against this Server and run its Operations against those Servers. It will allways use the first Server. If that is not available, it will the try to connect to the second one.
Ldap Server Port	389	The LDAP Port your Server responses to
Ldap Use SSL	false	Use SSL for LDAP. If possible use TLS
Ldap Use TLS	false	If checked, the Service tries to use TLSv1 for LDAP. (This runs on 389 as Extension / 636 is not required for security)

Users

Users
Users that are allowed to use this Service (Full LDAP DN. Example: cn=myuser,ou=users,o=data)

UserDN

cn=admin,ou=sa,o=system [Delete](#)

[Add UserDN](#) [Delete Last UserDN](#)

Name	Good Startvalue	Description
UserDN	cn=admin,ou=sa,o=system	a fqdn of a user, you wanna give access to this API. NOTE: If no User / Users is provided, every available User of the LDAP Directory can use this API. This is Usefull, if you like to use this API in a Webpage or similar to change some personal Data like mobiles or whatever

Installation and Configuration Guide

Logging

☒ **Audit to Elasticsearch**
 Enable Auditing to an Index of an Elasticsearch Cluster
Es Audit Indexname

Index Name in Elasticsearch to write to. Allowed are SimpleDateFormat parts in square brackets. Example: audit[YYYYMM] -> audit201605. see <https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html>
☒ **Enable Advanced Audit**
Add more information passed from Request to the Auditing
AuditFields
Add more information to the Audit

Fieldname	Alias for the Field	isMandatory
test	testAlias	☒ true Delete

Add Fields Delete Last Fields

☒ **Log to Elasticsearch**
 Enable Logging to an Index of an Elasticsearch Cluster
Es Log Indexname
 Index Name in Elasticsearch to write to. see Es Audit Indexname
Cluster Name
 Name of the ES Cluster we connect to.
Es Transport Port
 Transport Port to Communicate with the Es Cluster. Default 9300
EsNodes
IP Addresses / DNS Nodenames of the ES Cluster to Connect to. Example: localhost

IP
192.168.1.10 Delete

Add IP Delete Last IP

Elasticsearch Loglevel
 Loglevel to apply to ES logging
☒ **Log to a File**
Log to Files
Logfile Path
 Path to Log to. Try 'logs'
File Loglevel
 Loglevel to apply to Filelogging

Name	Good Startvalue	Description
Audit to Elasticsearch	true/false	If this box i checked, we try to log to an es cluster (v 1.x)
Es Audit Indexname	audit[YYYY]	The name of the Index we store our Audit Informations into. This Field can contain [] with the SimpleDateFormat to add dynamic to it. Examples: audit[YYYY] -> audit2016 audit[YYMM] -> audit1608

Installation and Configuration Guide

Enable Advanced Audit	true/false	If this is checked, we add Additional Information to the Auditobjects. See AuditFields
AuditFields	Requestor	If a Field is Added, we can have a Alias for it. That will be used in the Requestobject. If the Field is marked as Mandatory (isMandatory = true), the Service will Fail with Error 412, if no value is provided. The Content in the Request defined by: <pre>"Audit": [{ "Requestor": { "value": "Hans Muster[USR007]" } }]</pre> will be stored in the Field "Requestor" with the Value "Hans Muster[USR007]"
Log to Elasticsearch	true/false	The Log from the Service will be sent to the ES Cluster The Entry looks like: <pre>"_source": { "message": "Check if User allowed for this Component: cn=admin,ou=sa,o=system", "postDate": "2016-0829T08:34:02.952Z", "data": null, "severity": "INFO", "severitycode": 3 }</pre>
Es Log Indexname	log[YYYY]	The same as "Es Audit Indexname". Except this is for Logging purposes
Cluster Name	elasticsearch	The Clustername, the Transport client tries to connect to. The default is "elasticsearch"
Es Transport Port	9300	The Port the Transport client tries to reach the Cluster on (Transport not REST api !)
EsNodes	escluster	Ip or Hostname of the ClusterNodes. Need at least one valid, to get the Transport Client into the running Cluster. (Bootnodes)
Elasticsearch Loglevel	Warning	The Loglevel we use for ES Logging.
Log to File	true/false	Create Service Logfiles
Logfile Path	logs	Path to place the Logfiles into
File Loglevel	Info	The Loglevel we use for File Logging

Installation and Configuration Guide

Object configuration

Object
Add item Delete Last item Delete All

0 - Users
1 - Groups

0 - Users
Delete item
Edit JSON

Servicename
Users
The Name for the Service on This Object

BaseDN
ou=users,o=data
LDAP BaseDN the Service Operates in

createDN
ou=users,o=data
LDAP DN, where new Objects has to be placed in

Filter
objectClass=*
Additional LDAP Filter

☒ Force Creates in Create DN
Force all new Objects to be in container CreateDN or a subcontainer of it

☒ Readable
Is this Object readable

☒ Writable
Is this Object writable

☒ Deleteable
Is this Object deletable

☒ Addable
Is this Object addable

☒ Allow Multiple Operations
If we define a Modify or Delete via Search, can we modify / delete multiple Objects at once

InputScript
JS Script to Apply to Input Documents

OutputScript
JS Script to apply to Output Documents

Attributes
Users that are allowed to use this Service (Full LDAP DN. Example: cn=myuser,ou=users,o=data)

Attributename	Alias for the Attribute	isReadable	isWritable	isSearchable	isNaming	Default Values for Add
cn	Commonname	true	false	true	true	Add Value Delete
givenName	Vorname	true	true	true	false	Add Value Delete
SN	Nachname	true	true	true	false	Add Value Delete
telephoneNumber	telephonenumber	true	true	true	true	Add Value Delete

Add Attribute
Delete Last Attribute
Delete All

Name	Good Startvalue	Description
Servicename	Users	This is the Servicename. The Configname will set the URI Part. This is the Reference you need to set in your Request <Service>
BaseDN	ou=users,o=data	This is the BaseDN used for all the Operations. You can leave it empty, for no restrictions. But we suggest, you enter a DN here
CreatedDN	ou=users,o=data	This DN defines, in which Container / Subcontainer new Objects should be placed in.
Filter	objectClass=*	You need to provide a Filter here. Without (). It will restrict Search Operations / Multi Operation Finds

Installation and Configuration Guide

Allow Options to be used	false	If you want to change BaseDN / CreatedDN or Filter on the fly through the Request enable this Option
Force Creates in Create DN	true	Use the Create DN for Adds instead of the BaseDN
Readable	true	This Objectconfig allows read. Needed for Search
Writeable	true	This Objectconfig allows write. Needed for Modifies
Deleteable	false	This Objectconfig allows delete. Needed for Deletes / Renames / Moves
Addable	true	Allows Adds. Also needed for Moves /Renames
Allow Multiple Operations	true	If this is checked. You can pass "Searches" for Modifies and Deletes, that will match multiple Objects. Operation is then performed on all finds If not checked, it will only perform an Operation, if only one Object has matched Search criteria.
InputScript/OutputScript		If this is provided, it will run this Script on the Object received/sent from/to a client. The scripts provided has to be Javascript. the simplest possible script would be: result=data; this will copy the data to the result. Which result in no change at all. For deeper insight. Check the Scripting documentation.

Attributes

Name	Good Startvalue	Description
Attributename		The name of the Attribute in the LDAP directory. Eg: CN / SN / givenName
Alias for the Attribute		The name this Attribute is presented in the REST API as. Can be left blank
isReadable	true	Attribute should be readable of the API
isWritable	true/false	Attribute can be written over the API
isSearchable	true/false	This Attribute can used as Searchattribute
isNaming	false	This attribute is a naming Attribute, can be used to auto generate DNS
Default Values for Add		Add default Values to this Attribute on add /create of Objects. For example create an attribute objectClass, set Readable / Writeable / Searchable / Naming to false and set the Default values like Person / inetOrgPerson etc here.

4.2 Scripting

Scripts Folder

In the main folder of the Service, there is a scripts directory.
In that Directory there should be at least a helper.js Javascript file.
You can add as many js file to that directory as you like. They will get loaded when the Script Engine starts.

Script Invoke

To use a function of any of those scripts file, you simply include the method / function in the Config by Input / Output Script.

Get Started

Create your Hello World.

Paste the following Snipped into one of your Configs Input Script:

```
debug('hello World!!'); result=data;
```

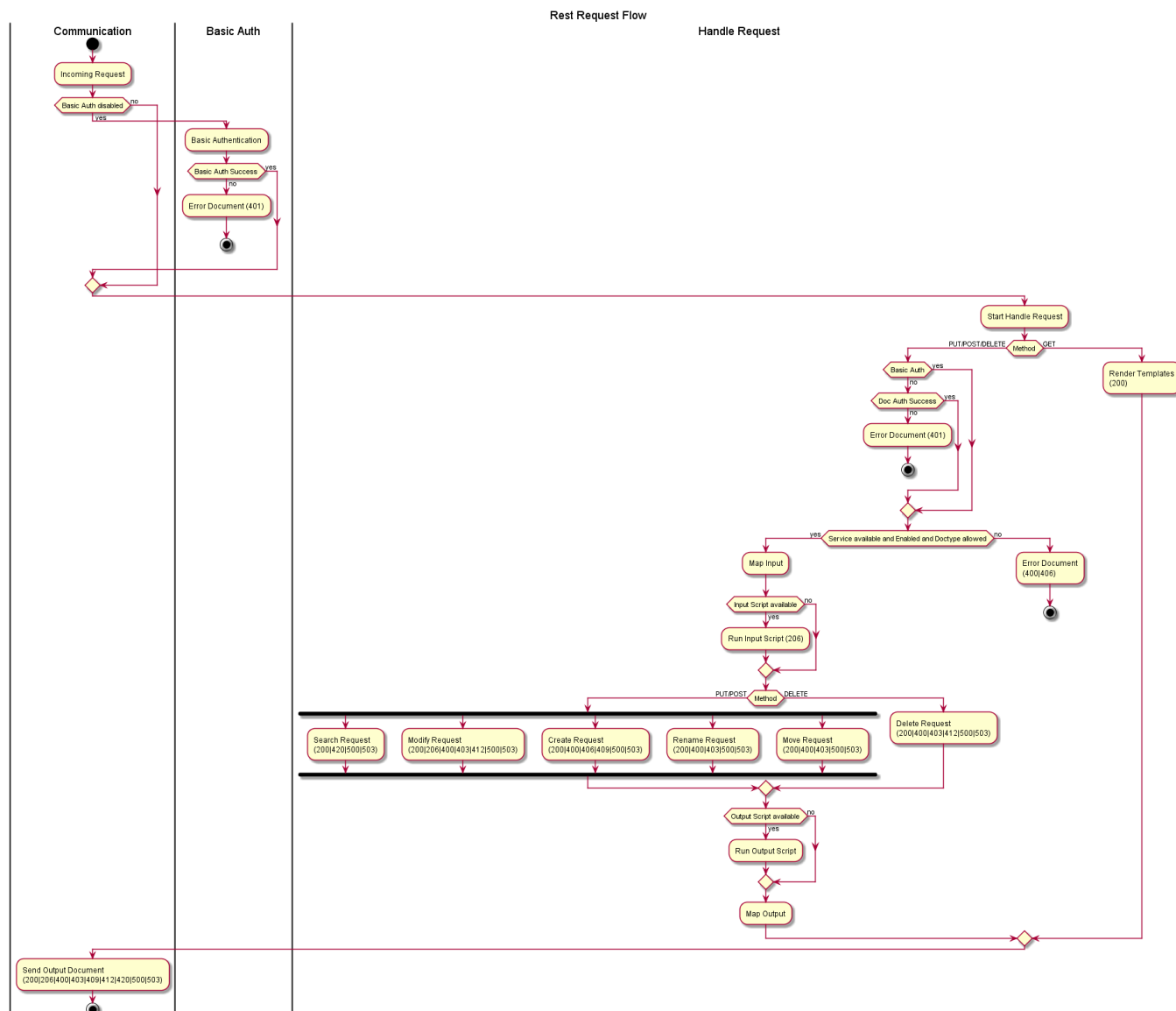
This will print a line on the Server Console with 'hello World!!' and copy the input data to the output object. This means no changes are made to the data at all.

You can now start modify the Input data with normal Javascript and copy the needed Result to the result Object.

Installation and Configuration Guide

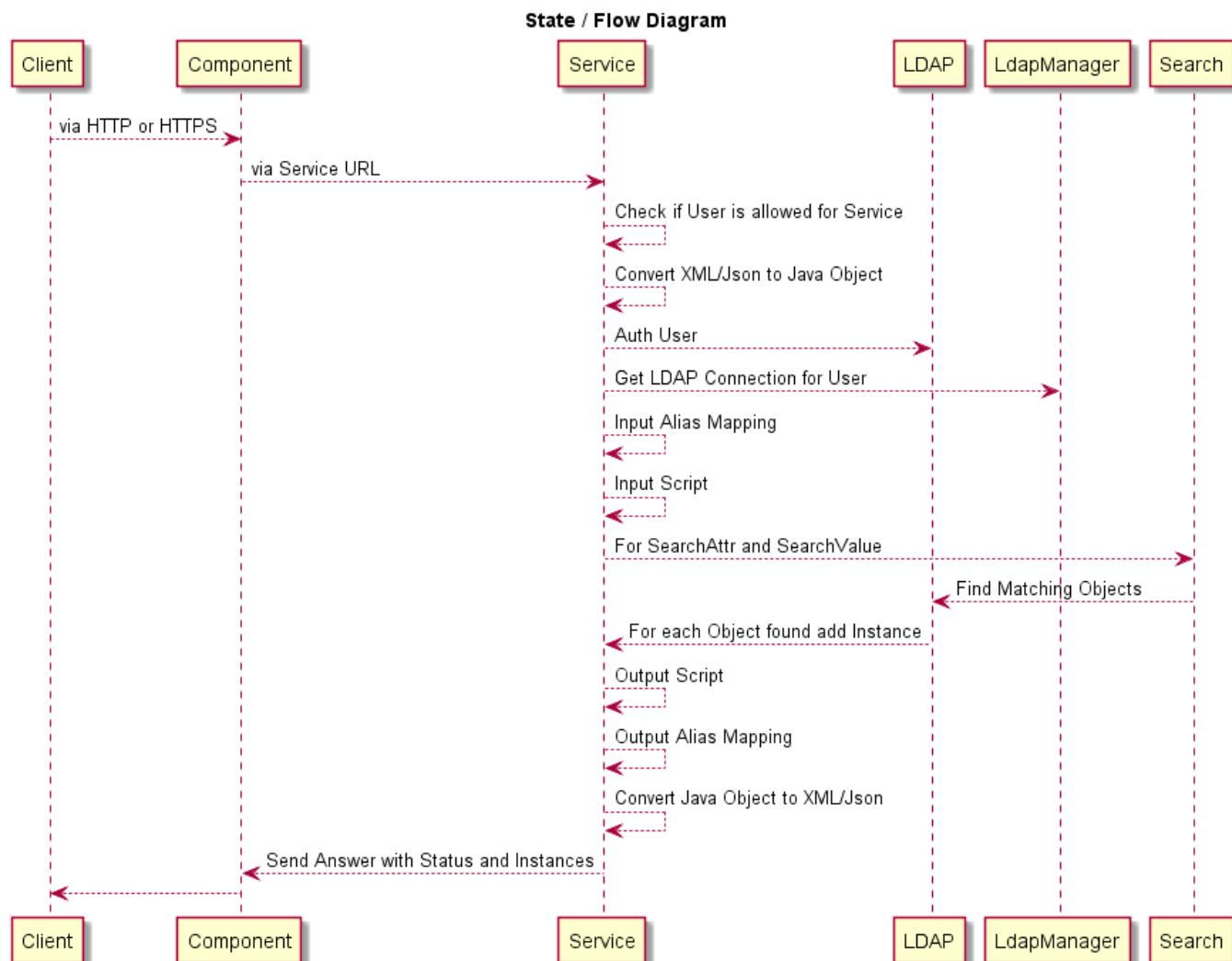
5 Service Flow Diagramme

General Request Flow



Installation and Configuration Guide

Search State / Flow



Installation and Configuration Guide

6 Error Codes / Handling

Error Codes are based on HTML Error Codes

<https://de.wikipedia.org/wiki/HTTP-Statuscode>

We use the following Error Codes:

Code	Message	Meaning
200	OK	All went as expected
206	Partial OK	Some Operations could be performed (On Partial Operations enabled)
400	Bad Request	• The Request could not be processed • The Document could not be parsed
401	Unauthorized	Check your Credentials / Tokens
403	Forbidden	Not allowed by the Config
406	Unacceptable	Not a valid DN
409	Conflict	• Object Exists on Add • Does not Exist by Delete
412	Preconditions Failed	• Search / Filter matched more than one Object, but not enabled or allowed • Advanced Audit Value not provided
420	Method Failed	A given Attribute was not Found / is not allowed to Search on etc.
500	Server Error	Unexpected Error. Should not happen.
503	Ldap Error	No LDAP Connection available / All Servers down or unable to Connect

Error Document / Answers

The Response you receive, has an Status Section.
For Success it simply looks like (JSON)

```
"Status": {  
  "statusCode": 200,  
  "message": "OK"  
}
```

Or as XML

Installation and Configuration Guide

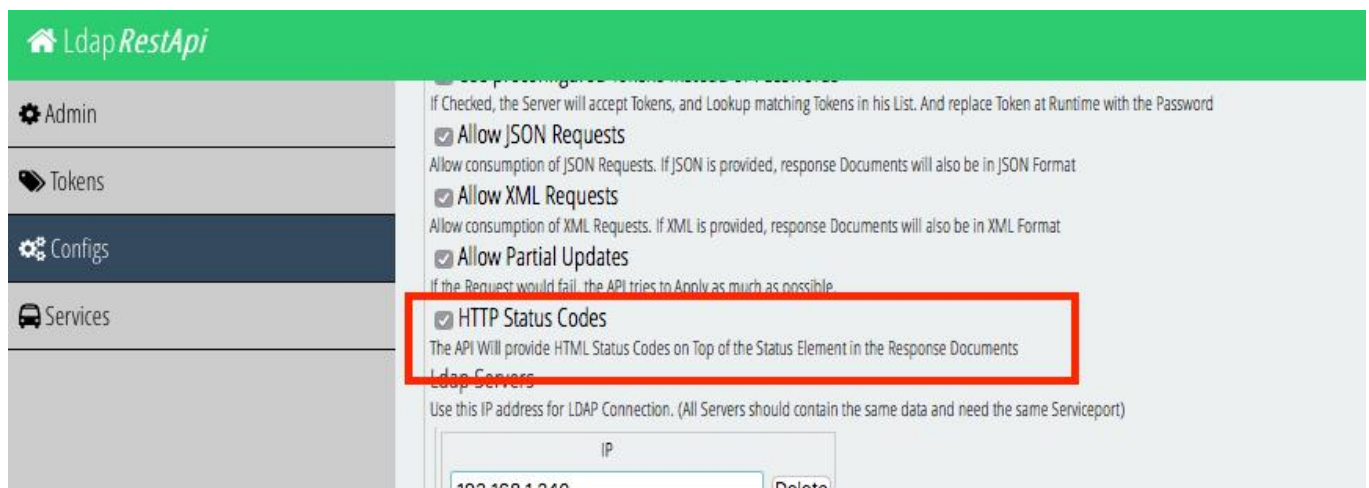
```
<Status>
<statusCode>200</statusCode>
<message>OK</message>
</Status>
```

If an Error happens,
this Section will contain the relevant Data. Also you may enable StackTrace to Doc in Config and
end up with a Section like this:

```
{
  "Output": {
    "Status": {
      "statusCode": 409,
      "message": "An Object with this DN already exists. DN:
cn=todelete,ou=users,o=data -- Error: NDS error:
entry already exists (-606)",
      "errors": [
        "org.apache.directory.api.ldap.model.message.ResultCodeEnum.processResponse(ResultCodeEnum.java:2074)",
        "org.apache.directory.ldap.client.api.LdapNetworkConnection.add(LdapNetworkConnection.java:838)",
        "ch.ktn.edir.rest.RestRestlet.addEntry(RestRestlet.java:821)",
        .....
        .....
        "java.lang.Thread.run(Thread.java:744)"
      ]
    }
  }
}
```

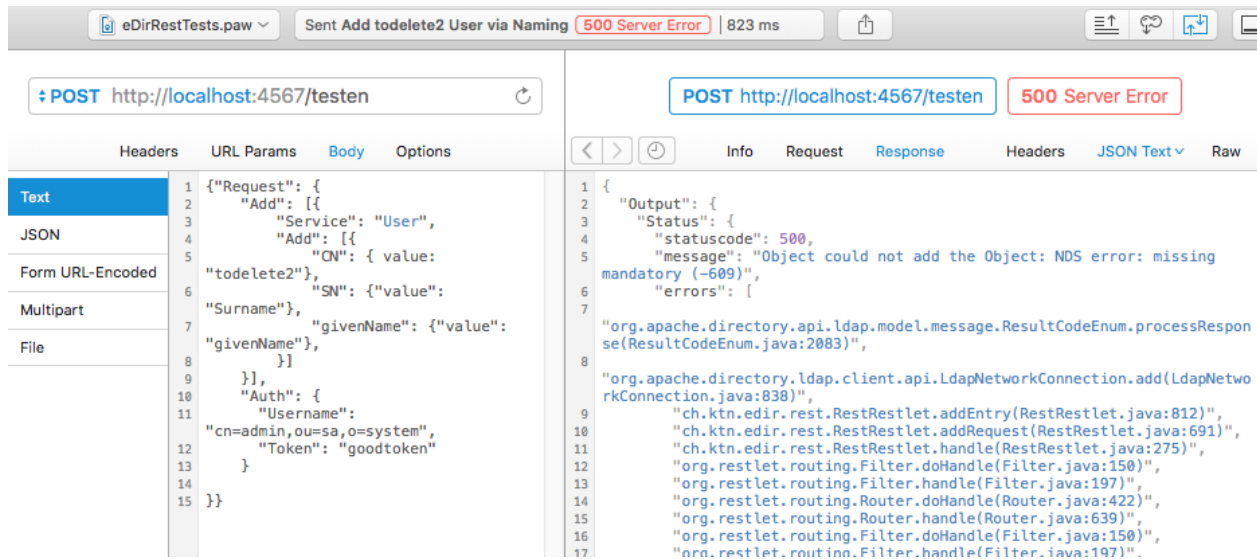
HTTP Status Codes

If you enable HTTP Status Codes



you get this Errors / Statuscodes also as HTTP Codes.

Installation and Configuration Guide



The screenshot shows a REST client interface with a POST request to `http://localhost:4567/testen`. The request body is a JSON object containing user information and authentication details. The response is a 500 Server Error with a detailed message: "Object could not add the Object: NDS error: missing mandatory (-609)".

```

POST http://localhost:4567/testen

{"Request": {
  "Add": [{
    "Service": "User",
    "Add": [{
      "CN": { value:
        "todelete2"},
      "SN": {"value":
        "Surname"},
      "givenName": {"value":
        "givenName"},
    }
  ]
},
  "Auth": {
    "Username":
      "cn=admin,ou=sa,o=system",
    "Token": "goodtoken"
  }
}]
}

{"Output": {
  "Status": {
    "statusCode": 500,
    "message": "Object could not add the Object: NDS error: missing mandatory (-609)",
    "errors": [
      "org.apache.directory.api.ldap.model.message.ResultCodeEnum.processResponse(ResultCodeEnum.java:2083)",
      "org.apache.directory.ldap.client.api.LdapNetworkConnection.add(LdapNetworkConnection.java:838)",
      "ch.ktn.edir.rest.RestRestlet.addEntry(RestRestlet.java:812)",
      "ch.ktn.edir.rest.RestRestlet.addRequest(RestRestlet.java:691)",
      "ch.ktn.edir.rest.RestRestlet.handle(RestRestlet.java:275)",
      "org.restlet.routing.Filter.doHandle(Filter.java:150)",
      "org.restlet.routing.Filter.handle(Filter.java:197)",
      "org.restlet.routing.Router.doHandle(Router.java:422)",
      "org.restlet.routing.Router.handle(Router.java:639)",
      "org.restlet.routing.Filter.doHandle(Filter.java:150)",
      "org.restlet.routing.Filter.handle(Filter.java:197)"
    ]
  }
}

```

Defined Errors

Defined but not all are used at the moment.

Error Codes

HtmlCodes
int INIT_0_NOT_SET
int INFO_100_CONTINUE
int INFO_101_SWITCHING_PROTOCOLS
int INFO_102_PROCESSING
int SUCCESS_200_OK
int SUCCESS_201_CREATED
int SUCCESS_202_ACCEPTED
int SUCCESS_203_NON_AUTHORITATIVE_INFORMATION
int SUCCESS_204_NO_CONTENT
int SUCCESS_205_RESET_CONTENT
int SUCCESS_206_PARTIAL_CONTENT
int SUCCESS_207_MULTI_STATUS
int REDIR_300_MULTIPLE_CHOICES
int REDIR_301_MOVED_PERMANENTLY
int REDIR_302_MOVED_TEMPORARILY
int REDIR_303_SEE_OTHER
int REDIR_304_NOT_MODIFIED
int REDIR_305_USE_PROXY
int REDIR_307_TEMPORARY_REDIRECT
int CLIENT_400_BAD_REQUEST
int CLIENT_401_UNAUTHORIZED
int CLIENT_402_PAYMENT_REQUIRED
int CLIENT_403_FORBIDDEN
int CLIENT_404_NOT_FOUND
int CLIENT_405_METHOD_NOT_ALLOWED
int CLIENT_406_NOT_ACCEPTABLE
int CLIENT_407_PROXY_AUTHENTICATION_REQUIRED
int CLIENT_408_REQUEST_TIMEOUT
int CLIENT_409_CONFLICT
int CLIENT_410_GONE
int CLIENT_411_LENGTH_REQUIRED
int CLIENT_412_PRECONDITION_FAILED
int CLIENT_413_REQUEST_TOO_LONG
int CLIENT_414_REQUEST_URI_TOO_LONG
int CLIENT_415_UNSUPPORTED_MEDIA_TYPE
int CLIENT_416_REQUESTED_RANGE_NOT_SATISFIABLE
int CLIENT_417_EXPECTATION_FAILED
int CLIENT_419_INSUFFICIENT_SPACE_ON_RESOURCE
int CLIENT_420_METHOD_FAILURE
int CLIENT_422_UNPROCESSABLE_ENTITY
int CLIENT_423_LOCKED
int CLIENT_424_FAILED_DEPENDENCY
int SERVER_500_INTERNAL_SERVER_ERROR
int SERVER_501_NOT_IMPLEMENTED
int SERVER_502_BAD_GATEWAY
int SERVER_503_SERVICE_UNAVAILABLE
int SERVER_504_GATEWAY_TIMEOUT
int SERVER_505_HTTP_VERSION_NOT_SUPPORTED
int SERVER_507_INSUFFICIENT_STORAGE

Installation and Configuration Guide

Framework Errors

The underlying Framework Restlet supports/depends on these Errors:

<https://restlet.com/http-status-codes-map>

Could not access the content at the URL because it is not from an allowed source.

<https://restlet.com/http-status-codes-map>

You may contact your site administrator and request that this URL be added to the list of allowed sources.

7 Request / Response Documents

Request as XML

Start Tag

XML Request

```
<Request>
```

Optional Options Block, if enabled in the Config

XML Request

```
<Options>
<BaseDN>ou=users,o=data</BaseDN>
<CreateDN>ou=users,o=data</CreateDN>
<Filter>objectClass=Person</Filter>
</Options>
```

One of the Following:

Search Block with Attr / Value

XML Request

```
<Search>
<SearchAttribute>CN</SearchAttribute>
<SearchValue>Value You Search for</SearchValue>
<Service>User</Service>
<ReadAttribute>givenName</ReadAttribute>
<ReadAttribute>eMail</ReadAttribute>
</Search>
```

Installation and Configuration Guide

Search Block with dn

XML Request

```
<Search>
<dn>uid=admin,ou=system</dn>
<Service>User</Service>
<ReadAttribute>givenName</ReadAttribute>
<ReadAttribute>eMail</ReadAttribute>
</Search>
```

Search Block with Attr / Value - Paged Search

XML Request

```
<Search>
<SearchAttribute>CN</SearchAttribute>
<SearchValue>Value You Search for</SearchValue>
<ObjectPerPage>10</ObjectPerPage>
<PagedSearch>true</PagedSearch>
<Service>User</Service>
<ReadAttribute>givenName</ReadAttribute>
<ReadAttribute>eMail</ReadAttribute>
</Search>
```

This will add a Cookie to the Response

```
<Output>.....<PagedCookie>TheCookie</PagedCookie></Output>
```

Search Block with Attr / Value - Paged Search – FollowUp

Installation and Configuration Guide

XML Request

```
<Search>
<SearchAttribute>CN</SearchAttribute>
<SearchValue>Value You Search for</SearchValue>
<ObjectPerPage>10</ObjectPerPage>
<PagedSearch>true</PagedSearch>
<PagedCookie>CookieFromLastRequest</PagedCookie>
<Service>User</Service>
<ReadAttribute>givenName</ReadAttribute>
<ReadAttribute>eMail</ReadAttribute>
</Search>
```

Modify Block

XML Request

```
<Modify>
<dn>DN of the Object to Modify or a Search</dn>
<SearchAttribute>Attributename to search on if not using DN</SearchAttribute>
<SearchValue>Value You Search for</SearchValue>
<Service>User</Service>
<ProcessMultiMatch>>false</ProcessMultiMatch>
<Add>
<groups>
<value>group 1</value>
<value>group 2</value>
</groups>
</Add>
<Remove>
<groups>
<value>oldgroup 3</value>
</groups>
</Remove>
<Replace>
<givenName>
<value>new name</value>
<value>second new name</value>
</givenName>
</Replace>
</Modify>
```


Installation and Configuration Guide

Delete Block

XML Request

```
<Delete>
<dn>DN of the Object to Delete</dn>
<SearchAttribute>Attributename to search on and delete finds</SearchAttribute>
<SearchValue>Value you search for. Results will be Deleted</SearchValue>
<Service>User</Service>
<ProcessMultiMatch>>false</ProcessMultiMatch>
</Delete>
```

Create Block

XML Request

```
<Create>
<dn>Enter the DN here or Class Default Container and Naming Attribute will be
used</dn>
<Service>User</Service>
<Add>
<cn>
<value>cn - for dn creation if this is checked as naming</value>
</cn>
<SN>
<value>Surname</value>
</SN>
<givenName>
<value>givenName</value>
</givenName>
</Add>
</Create>
```

Installation and Configuration Guide

Rename Block

XML Request

```
<Rename>
<OldDN>cn=oldname,o=demo</OldDN>
<Newname>cn=newname</Newname>
<Service>User</Service>
</Rename>
```

Move Block

XML Request

```
<Move>
<OldDN>cn=oldname,o=demo</OldDN>
<TargetContainer>ou=target,o=newname</TargetContainer>
<Service>User</Service>
</Move>
```

Optional: Auth Block

XML Request

```
<Auth>
<Username>cn=myuser,ou=technicalusers,o=unit</Username>
<Token>Token because Token Authentication is Enabled</Token>
</Auth>
```

Installation and Configuration Guide

XML Request

```
<Auth>
<Username>cn=myuser,ou=technicalusers=o=unit</Username>
<Password>MySecret</Password>
</Auth>
```

Optional: Audit Block

XML Request

```
<Audit>
<advAuditField>
<value>If AdvAudit is enabled, you pass values like this</value>
</advAuditField>
</Audit>
```

Stop Tag

XML Request

```
</Request>
```

Installation and Configuration Guide

A Full example would look like:

XML Request

```
<Request>
<Search>
<SearchAttribute>CN</SearchAttribute>
<SearchValue>Value You Search for</SearchValue>
<Service>User</Service>
<ReadAttribute>givenName</ReadAttribute>
<ReadAttribute>eMail</ReadAttribute>
</Search>
<Auth>
<Username>cn=myuser,ou=technicalusers=o=unit</Username>
<Token>Token because Token Authentication is Enabled</Token>
</Auth>
</Request>
```

Installation and Configuration Guide

The structure is the same in JSON. As an example:

JSON Request

```
{
  "Request": {
    "Modify": {
      "dn": "DN of the Object to Modify or a Search",
      "SearchAttribute": "Attributename to search on if not using DN",
      "SearchValue": "Value You Search for",
      "Service": "User",
      "ProcessMultiMatch": false,
      "Add": [
        {
          "groups": {
            "value": [
              "group 1",
              "group 2"
            ]
          }
        }
      ],
      "Remove": [
        {
          "groups": {
            "value": "oldgroup 3"
          }
        }
      ],
      "Replace": [
        {
          "givenName": {
            "value": [
              "new name",
              "second new name"
            ]
          }
        }
      ]
    },
    "Auth": {
      "Username": "cn=myuser,ou=technicalusers=o=unit",
      "Token": "Token because Token Authentication is Enabled"
    }
  }
}
```

Installation and Configuration Guide

Response Documents

A normal response document in XML would be:

XML Response

```
<Output>
<Data>
<Object>
<dn>cn=anObject,ou=anOrgUnit,o=Org</dn>
<Content>
<cn>
<value>anObject</value>
</cn>
<givenName>
<value>My Name</value>
<value>My Other Name</value>
</givenName>
</Content>
</Object>
</Data>
<Status>
<statuscode>200</statuscode>
<message>OK</message>
</Status>
</Output>
```

If an Error happen, you get the Error in the Status section. It looks like this:

XML Response

```
<?xml version="1.0"?>
<Output>
<Data/>
<Status>
<statuscode>420</statuscode>
<message>The given Attribut is not searchable. Check your config.</message>
</Status>
</Output>
```

Installation and Configuration Guide

If you enable Generic Instance Output, you will get a more Generic Document for Input and Output. If it is checked, you can still send simple Documents to the API

Example of a more Generic Doc:

XML Response

```
<Request>
  <Modify>
    <dn>DN of the Object to Modify or a Search</dn>
    <SearchAttribute>Attributename to search on if not using DN</SearchAttribute>
    <SearchValue>Value You Search for</SearchValue>
    <Service>User</Service>
    <ProcessMultiMatch>>false</ProcessMultiMatch>
    <Add>
      <Attribute>
        <name>groups</name>
        <value>group 1</value>
        <value>group 2</value>
      </Attribute>
    </Add>
    <Remove>
      <Attribute>
        <name>groups</name>
        <value>oldgroup 3</value>
      </Attribute>
    </Remove>
    <Replace>
      <Attribute>
        <name>givenName</name>
        <value>new name</value>
        <value>second new name</value>
      </Attribute>
    </Replace>
    </Modify>
    <Auth>
      <Username>cn=myuser,ou=technicalusers,o=unit</Username>
      <Token>Token because Token Authentication is Enabled</Token>
    </Auth>
    <Audit>
      <Attribute>
        <name>advAuditField</name>
        <value>If AdvAudit is enabled, you pass values like this</value>
        <value>If an Alias is defined</value>
      </Attribute>
    </Audit>
  </Request>
```