OX App Suite Engineering Services Plugins Technical Documentation for
1.7.2-rev1

2022-07-18

Contents

1 General Information

1.1 Warnings

⚠ **Warning**
Images included in following pages have been attached as a generic visual reference for the theme and should not be considered as the final aspect when installed on production environment. Actual aspect will change based on components/plugins enabled and their configuration.

⚠ **Warning**
It is mandatory to restart the **open-xchange** service on all middleware nodes after performing the update.

⚠ **Warning**
Custom configuration or template files are potentially not updated automatically. After the update, please always check for files with a **.dpkg-new** or **.rpmnew** suffix and merge the changes manually. Configuration file changes are listed in their own respective section below but don't include changes to template files. For details about all the configuration files and templates shipped as part of this delivery, please read the relevant section of each package.

1.2 Delivery Comment

This delivery was requested with following comment:

Plugins 1.7.2 Feature Delivery for Core 7.10.6

1.3 Install Package Repository

This delivery is part of a restricted software repository:

https://software.open-xchange.com/components/plugins/stable/1.7.2/DebianStretch
https://software.open-xchange.com/components/plugins/stable/1.7.2/DebianBuster
https://software.open-xchange.com/components/plugins/stable/1.7.2/DebianBullseye
https://software.open-xchange.com/components/plugins/stable/1.7.2/RHEL7

1.4 Build Dependencies

This delivery was build and tested with following dependencies:

```
AppSuite:node-10,frontend-7.10.6-rev12,backend-7.10.6-rev16
```

1.5 Notice

ℹ **Info**
Some configurations can be changed without restarting the service, please call following command for getting a list of supported settings.

```
/opt/open-xchange/sbin/listreloadables
```

Please use following command to enable capable and changed configurations on a running system.

```
/opt/open-xchange/sbin/reloadconfiguration
```

1.6 Release Announcements

We created a dedicated mailing list for on-prem customers using the OX App Suite Engineering Services Plugins. You should subscribe to this mailing list here if you are using one of those plugins in your installation and want to get relevant updates: https://lists.open-xchange.com/mailman/listinfo/plugins-announce

2 Anti Phishing Framework

Bundle Identifier	com.openexchange.plugins.antiphishing, com.openexchange.plugins.antiphishing.json, com.openexchange.plugins.antiphishing.connector.vadesecure
Package(s)	open-xchange-plugins-antiphishing, open-xchange-plugins-antiphishing-vadesecure
Required capabilities	none
Available since	1.6.4-rev2

The anti-phishing implementations supports a “connector” framework. Using this model, any number of custom anti-phishing connector implementations can register with the connector framework. The decision on which implementation to use is determined at runtime via a connector identifier and config-cascade. As such, the connector implementation can be configured on the Global, Brand, Context or User level. However, brand will be the most likely scenario.

The base package open-xchange-plugins-antiphishing provides the **AntiPhishingInterface** service which acts as a container for all registered **AntiPhishingConnector** services and as an adapter between the **PluginsAntiPhishingActionfactory** servlet and each individual AntiPhishingConnector instance. During bundle activation a ServiceTracker is registered to track, collect and map all registered AntiPhishingConnector service instances by brand. When invoked, the **AntiPhishingInterface** looks up an appropriate **AntiPhishingConnector**, if one exists, the request is forwarded to the specific connector.

2.1 Framework Configuration

The configuration needs to be done within plugins-antiphishing.properties.

```
# Setting to control the used connector for a specific user
# This setting is config-cascade aware to support different implementations for each user.
# Default is <none> which means that the feature is disabled for a user
# To enable vade secure com.openexchange.plugins.antiphishing.connector=
    plugins_antiphishing_vadesecure
com.openexchange.plugins.antiphishing.connector=

# Setting to enable/disable the antiphishing capability
# This setting is config-cascade aware to support different implementations for each user.
# Default is false which means that the feature is disabled for a user
com.openexchange.plugins.antiphishing.enabled=false

# Setting to enable/disable the antiphishing mta_capability
# If true, the user has the ability to choose antiphishing at the MTA level
# This setting is config-cascade aware to support different implementations for each user.
# Default is false which means that the feature is disabled for a user
com.openexchange.plugins.antiphishing.mta_capability=false

# Setting to enable/disable the antiphishing at the mta level
# If true, an antiphishing check will take place at the MTA level
# This setting is config-cascade aware to support different implementations for each user.
# Additionally, this property can be set by the user in the UI
# Default is false which means that the feature is disabled for a user
com.openexchange.plugins.antiphishing.mta_antiphishing=false
```

2.2 Anti Phishing API

The Anti-phishing servlet is registered at http://localhost/api/plugins/antiphishing and supports following actions:

Action	Description
`config`	Returns a boolean result telling whether MTA level anti-phishing is supported or not. If it is supported, the UI makes a followup request to determine the configured status of the MTA level anti-phishing.
`get`	Returns true if MTA level anti-phishing is enabled for a given user, false if not.
`update`	Enables/disables the MTA level anti-phishing, returns the new status.
`linkcheck`	Calls the connector framework to make the anti-phishing call on a list of one or more URIs and/or mailto addresses. Returns `SUCESS` or `ERROR` with an appropriate message.

2.2.1 Example Config Request

```
curl 'http://localhost/appsuite/api/plugins/antiphishing?action=config&session=
    df33d98d72914f1c96b26d6827deee3e'
```

Example Config Response

```
{"data":{"STATUS":"OK","mta_capability":true}}
```

2.2.2 Example Get Request

```
curl 'http://localhost/appsuite/api/plugins/antiphishing?action=get&session=
    df33d98d72914f1c96b26d6827deee3e'
```

Example Get Response

```
{"data":{"STATUS":"OK","mta":false}}
```

2.2.3 Example Update Request

```
curl 'http://localhost/appsuite/api/plugins/antiphishing?action=update&session=
    df33d98d72914f1c96b26d6827deee3e' -X 'PUT' --data-binary '{"mta":true}'
```

Example Update Response

```
{"data":{"STATUS":"OK","mta":true}}
```

2.2.4 Example Linkcheck Request

```
curl 'http://localhost/appsuite/api/plugins/antiphishing?action=linkcheck&session=
    df33d98d72914f1c96b26d6827deee3e' -X 'PUT' --data-binary '{"mta":true}'
```

Example Linkcheck Response

```
{"data":{"STATUS":"PHISHING"}}
```

2.3 The Anti Phishing Connectors

The anti-phishing framework requires an anti-phishing connector implementation.

2.3.1 The VadeSecure Connector

The package open-xchange-plugins-antiphishing-vadesecure provides the concrete implementation of a **VadeSecureAntiPhishingConnector** and registers it as a **AntiPhishingConnector** service and comes with a plugins-antiphishing-vadesecure.properties

```
# The customer name as provided by VadeSecure; required to access Phishing API
# Default: NONE
# Config-cascade aware: true
# Lean: false
com.openexchange.plugins.antiphishing.vadesecure.name.passcrypt=<Customer name provided by VadeSecure>

# The customer license provided by VadeSecure; required to access Phishing API
# Default: NONE
# Config-cascade aware: true
# Lean: false
com.openexchange.plugins.antiphishing.vadesecure.license.passcrypt=<Customer license provided by VadeSecure>

# Setting to change the VadeSecure IsItPhishing API URL
# Default: https://iip.eu.vadesecure.com/api/v2/url
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.antiphishing.vadesecure.phishing_url=https://iip.eu.vadesecure.com/api/v2/url

# Setting to change the VadeSecure GRAPH Authentication API URL
# Default: https://api.vadesecure.com/oauth2/v2/token
# Config-cascade aware: false
com.openexchange.plugins.antiphishing.vadesecure.graph_url=https://api.vadesecure.com/oauth2/v2/token

# Setting to change the VadeSecure connector identifier referenced in plugins-antiphishing.properties / com.openexchange.plugins.antiphishing.connector
# Default: "plugins_antiphishing_vadesecure"
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.antiphishing.vadesecure.identifier=plugins_antiphishing_vadesecure

# If set to true, the URL will always be crawled and analyzed, even if it can trigger collateral damages (such as unsubscribing a user, canceling an order, etc.).
# If set to false, the service checks whether the URL may cause collateral damage to the end user (unsubscribe, order confirmation, etc.). If so, the URL is not crawled and NOT_EXPLORED is returned in the response.
# Default: false
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.antiphishing.vadesecure.force=false

# Vade Secure IsItPhishing Smart mode enables URL anonymization. Typically, this is meant to
# replace any unique-ID like tokens in a URL by random characters, to prevent side effects when crawling certain URLs, which if visited, could trigger unwanted actions: unsubscription, cancelation, etc.
# Set to true to enable the smart mode. If set to false, URLs will be crawled in the way they were originally provided. If argument randomization fails, the URL is not crawled and NOT_EXPLORED is returned.# Default: "plugins_antiphishing_vadesecure"
# NOTE: Vade Secure strongly recommends enabling the smart parameter to true, so that the API can trigger token anonymization, to try and prevent any collateral damages.
# Default: false
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.antiphishing.vadesecure.smart=true

# Timeout in milliseconds, with a minimum value of 1000. Once timeout is reached, TIMEOUT response is returned.
# Default: 3000
# Config-cascade aware: true
# Lean: true
```

```
com.openexchange.plugins.antiphishing.vadesecure.timeout=3000
```

2.3.1.1 Custom OXaaS LDAP Extension The cloud-plugins component provides a `open-xchange-cloudplugins-antiphishing-vadesecure-ldap` package to extent the VadeSecure connector with customized OXaaS LDAP support. Please check the external documentation for further details.

3 Block Allow List Middleware

Bundle Identifier	`com.openexchange.plugins.blackwhitelist,`
	`com.openexchange.plugins.blackwhitelist.connector.sieve,`
	`com.openexchange.plugins.blackwhitelist.clt,`
	`com.openexchange.plugins.blackwhitelist.json`

Package(s)	`open-xchange-plugins-blackwhitelist,`
	`open-xchange-plugins-blackwhitelist-sieve`

Required capabilities	none
Available since	1.2.1-rev2

The middleware component consists of a registry and possible available plugins that are registered and can be assigned to a user. The registry can be installed with the package `open-xchange-plugins-blackwhitelist`. This package holds the JSON layer and also computes the available capabilities for the user.

3.1 Connectors

3.1.1 Sieve Connector

The package `open-xchange-plugins-blackwhitelist-sieve` povides a middleware component that is responsible for the blackwhitelist handling on a sieve level. The configuration can be set on a config-cascade aware level for better control. Even though the name is blackwhitelist, the package only supports the blacklist part of the interface. This connector can be configured via the `plugins-blacklist-sieve.properties` file.

```
# Identifier of this blackwhitelist connector: plugins_blackwhitelist_sieve
# Setting to control the rulename to be set and checked as a antispam value inside the sieve rules
# Default: Blacklist
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.blackwhitelist.connector.sieve.rulename=Blacklist

# Setting to control wether the blacklisted mails should be moved to SPAM or deleted directly
# If set to true, mails are moved to SPAM
# If set to false, mails are deleted
# Default: true
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.blackwhitelist.connector.sieve.moveToSpam=true

# Setting to check if memory backed test System should be started
# This connector is identified by plugins_blwl_test
# Default: false
com.openexchange.plugins.blackwhitelist.connector.sieve.test=false
```

3.1.2 LDAP Connector

This connector is part of the **cloud-plugins** component and shipped via the package `open-xchange-cloudplugins-blackwhitelist-ldap`.

3.2 Commandline Tools

The following commandline tools are available:

Tools	Description
`listblackwhitelist`	List black and whitelist entries of a user
`searchblackwhitelist`	Search for black and whitelist entries of a user
`deleteblackwhitelist`	Delete entries for black and whitelist of user
`addblackwhitelist`	Add entries for black and whitelist of user

Please use the `--help` parameter to see further details like e.g.:

```
./addblackwhitelist --help
usage: addblackwhitelist
 -A,--adminuser <arg>          Admin username
 -c,--context <arg>            A valid context identifier
 -e,--entry <arg>              The entry to be added to the list.
 -h,--help                     Prints a help text
 -i,--userid <arg>             A valid user identifier
 -P,--adminpass <arg>          Admin password
 -p,--port <arg>               The optional RMI port (default:1099)
    --responsetimeout <arg>    The optional response timeout in seconds
                               when reading data from server (default: 0s;
                               infinite)
 -s,--server <arg>             The optional RMI server (default: localhost)
 -t,--type <arg>               The type, can either be blacklist or
                               whitelist.
```

3.3 Configuration

Possible connectors to be used in `com.openexchange.plugins.blackwhitelist.connector` are:

Identifier	Description
`plugins_blackwhitelist_sieve`	The sieve connector is shipped by the **plugins** package `open-xchange-plugins-blackwhitelist-sieve`.
`cloudplugins_blackwhitelist_ldap`	The customized OXaaS LDAP connected shipped by the **cloud-plugins** package `open-xchange-cloudplugins-blackwhitelist-ldap`.

This plugin needs to be configured via the `plugins-blackwhitelist.properties` file:

```
# Setting to control the used connector for a specific user
# This setting is config-cascade aware to support different implementations for each user.
# Default is <none> which means that the feature is disabled for a user
com.openexchange.plugins.blackwhitelist.connector=

# Setting to check if memory backed test System should be started
# This connector is identified by plugins_blwl_test
# Default: false
com.openexchange.plugins.blackwhitelist.test=false
```

4 Contact Whitelist Sync

Bundle Identifier	`com.openexchange.plugins.contacts.whitelist,`
	`com.openexchange.plugins.contacts.whitelist.migration,`
	`com.openexchange.plugins.contacts.whitelist.migration.clt,`
	`com.openexchange.plugins.contacts.whitelist.rdb`

Package(s)	`open-xchange-plugins-contact-whitelist-sync`

Required capabilities	none
Available since	1.4.5-rev1

The plugin provides a pluggable solution to sync contacts into an external database. This is done by hooking into contact **create/update/delete** events and forwarding those into a registry of:

```
com.openexchange.plugins.contacts.whitelist.service.PluginsContactWhitelistConnector
```

Those Connectors are monitored and provide an identifier which can be configured on a config-cascade level to enable the plugin. The plugin is also able to ignore the contact-collect folder if that is configured to only sync contacts that are in any user folder, but not the contact-collect folder. In the current version a SQL connector is provided.

4.1 Configuration

To enable the plugin, an admin has to set the following property to a currently supported value:

```
com.openexchange.plugins.contacts.whitelist.connector
```

As of now, this is only `rdb`. In later versions, `ldap` or something else might be supported. All of the following `/opt/open-xchange/etc/plugins-contacts-whitelist.properties` properties are **config-cascade** aware:

```
# This setting enables or disables special handling for the ContactCollectionFolder
# If set to true, the contactCollectFolder is ignored and contacts in this folder
# are not added to the whitelist. Contacts moved to this folder are also removed from the whitelist
# If set to false, the contactCollectFolder is handled like any other folder.
# config-cascade aware
# Default: true
com.openexchange.plugins.contacts.whitelist.ignoreContactCollectFolder=true

# This setting is used to set the connector for the contact sync.
# Currently available options are:
#  <not-set> (this will disable the sync for the user)
#  rdb
# Default: <not-set>
com.openexchange.plugins.contacts.whitelist.connector=
```

The rdb component can also be configured on a **config-cascade** level. First, a pool has to be enabled. This is done by using the SQL Client Library which is also part of the plugins repository. In addition, the rdb layer supports two strategies, one beeing **normal** and **tombstone**.

Mode	Create contact event	Update contact event	Delete contact event
`normal`	New contact mails are written into the database table.	Contacts in the database are read, missing ones are removed, new mails are added.	All mails of that contactId are removed.
`tombstone`	New contact mails are written into the database table where the current timestamp is added to the `updatedAtColumnName`.	Contacts in the database are read. Missing ones are not removed but updated with the current timestamp in the `deletedAtColumnName`. All current mails are updated with the current timestamp in the `updatedAtColumnName`.	All mails of that contactId are read. All mails are updated with the current timestamp in the `deletedAtColumnName`.

Furthermore, an admin can define the different table and column names used by the plugin via /opt/open-xchange/etc/plugins-contacts-whitelist-rdb.properties:

```
# Pool to be used
com.openexchange.plugins.contacts.whitelist.rdb.pool=contact-whitelist-pool

# normal or tombstone
com.openexchange.plugins.contacts.whitelist.rdb.strategy=normal

# table name
com.openexchange.plugins.contacts.whitelist.rdb.tableName=senderwl

# Name of the column used for the primary mail
com.openexchange.plugins.contacts.whitelist.rdb.primaryAddressColumnName=rcpt

# Name of the column used for the contact mails
com.openexchange.plugins.contacts.whitelist.rdb.contactMailColumnName=sender

# Name of the column used for the individual contactIds
com.openexchange.plugins.contacts.whitelist.rdb.contactIdColumnName=contactid

# Name of the deleted_at column if tombstone is enabled
com.openexchange.plugins.contacts.whitelist.rdb.tombstone.deletedAtColumnName=deleted_at

# Name of the updated_at column if tombstone is enabled
com.openexchange.plugins.contacts.whitelist.rdb.tombstone.updatedAtColumnName=updated_at
```

It is further possible to define a migration strategy at login time which will be executed by a LoginHandler via /opt/open-xchange/etc/plugins-contacts-whitelist-migration.properties:

```
# Defines the strategy of the automatic migration
# Can be either
#     <not-set> which disables the automatic migration
#     once
#     time:<timeinmillis>
# Default: <not-set>
#
# Examples
# If sync should happen once a day:
# com.openexchange.plugins.contacts.whitelist.migration.strategy=time:86400000
# If sync should happen once a week
# com.openexchange.plugins.contacts.whitelist.migration.strategy=time:604800000
com.openexchange.plugins.contacts.whitelist.migration.strategy=

# Setting, if a warning should appear in the logs, if a user has more than configured contacts in one folder.
# Default: 10000
com.openexchange.plugins.contacts.whitelist.migration.warningSize=10000
```

5 Group Contact Storage

Bundle Identifier	com.openexchange.plugins.contact.storage.group
Package(s)	open-xchange-plugins-contact-storage-group
Required capabilities	none
Available since	1.5.3-rev5

The Group Contact Storage enables virtual contact folders for members of internal user groups. Once installed and activated, the folders will be created dynamically for each group in a context. Via a permission entry for the represented group, these folders will be visible to those users who are themselves member of the corresponding group. Doing so, it is possible to categorize internal user contacts in structured views based on the group membership, especially in contexts with many users where the global addressbook would become too large, hence would better be hidden in clients. Possible use cases could be departments, offices or teams in large organizations or authorities, that can be represented as different user groups in the groupware.

5.1 Installation and Configuration

The group contact storage plugin is available through the package `open-xchange-plugins-contact-storage-group`. After installation, the storage still needs to be enabled explicitly for those contexts it should be used in by following setting the property to `true` via the config-cascade:

```
com.openexchange.plugins.contact.storage.group.enabled
```

Upon the next reload of the configuration, when the contact storage is first accessed in an enabled context, the group contact folders are dynamically created as needed for each group found in the context. For the system groups “All Users”, “All Guests” and the “Standard Group”, no folders are created of course. Additionally, it is possible to exclude further groups where no contact folder should be created for using the property:

```
com.openexchange.plugins.contact.storage.group.excludedGroups
```

It takes a comma-separated list of group identifiers and can also be defined through the config-cascade. Please see the `/opt/open-xchange/etc/plugins-contact-storage-group.properties` file which define those settings:

```
# Configures whether the group contact storage is enabled for a context or not.
# Default: false
com.openexchange.plugins.contact.storage.group.enabled=false

# Defines an optional list of those groups for which no group contact folder should
# be used, as a comma-separated string of the identifiers of those groups that should
# be excluded. The groups "All Users", "All Guests" and the "Standard Group" are
# always excluded.
# Default: <empty>
com.openexchange.plugins.contact.storage.group.excludedGroups=

# Defines if the display name of the groups should be used to create the folder
# names in the folder tree.
# If set to <true>, the displayname is used
# If set to <false>, the group name is used
# The Group Names are limited by the property CHECK_GROUP_UID_REGEXP
com.openexchange.plugins.contact.storage.group.useDisplayName=true
```

5.2 Group Folders

The group contact folders will be created below the system public folder (folder identifier 2), using the display names of the groups as folder name.

Info

If an equally named folder already exists at that location, it will be re-used implicitly, making its previous contents inaccessible as long as the plugin is enabled.

Groups with duplicate display names are skipped. The group contact folders will get two permission entries assigned: one administrative permission for the context admin, and one for the corresponding group entity, so that each member of the group will see the folder and all contained contacts, and is able to edit his “own” contact details. Fine-tuning of these inserted standard permissions can still be performed by the context administrator, however it is required that the group folder permission is not removed, otherwise it’ll get re-inserted again automatically during the next initialization.

Any changes of groups within a context that has group contact folders enabled leads to a reinitialization of the mapped contact folders, so that the changes are reflected automatically. This includes new contact folders for newly created groups, updated folder names for updated group display names, and deletions of folders when the corresponding group gets deleted. Changes of the group members will also directly lead to changes in the visibility of the corresponding group contact folder through the assigned group permission entry.

5.3 Global Address Book

Basically, it is still possible to use the default global addressbook folder in parallel. However, espacially in scenarios with many users within a single context a huge global address book folder is not really useful, both from the end user's experience as well as performance-wise. Here, the group contact storage delivers an alternative solution where users rather see their peers in one or more group contact folders, e.g. representing the members of their department in a company or office location. Here, access to the global addressbook can be switched off by setting the corresponding module permission `globaladdressbookdisabled`.

Info
In order to disable the global address book for non-PIM users, a rather historic permission check needs to be disabled by setting `com.openexchange.admin.bypassAccessCombinationChecks` to `true`.

With the global address book disabled, users can still collaborate with other users in the context, even if a user contact does not appear within a visible group contact folder. E.g. it is still possible to share folders, check free/busy times or create meetings with all other users, independently of their group membership. Although not all user contacts will appear in addressbooks or during autocomplete operations, they can still be addressed directly by their mail address. The middleware will then take care and recognize that there's an internal user entity behind the mail address implicitly.

6 Imap Util Library

Bundle Identifier	`com.openexchange.util.imap`
Package(s)	`open-xchange-util-imap`
Required capabilities	none
Available since	1.3.0-rev3

This package provides a library for common IMAP operations that are not available or usable in the AppSuite middleware core. It currently only includes an IMAP authentication feature. As it is a library and can be used for many different projects, the names of the configuration properties are configurable, at least their prefix is.

6.1 Usage

A project that wants to use the library needs to register an instance of the service JavaMailImapAuthenticator with an optional OSGi property to inform the IMAP library of which property name prefix to use to look up its configuration, e.g. like this when using util-custom's ActivatorTemplate:

Example `ProjectActivator.java`

```

public final class ProjectActivator extends ActivatorTemplate {
    @Override
    protected void registerServices(final Registrar registrar) {
        registrar
        .service(JavaMailImapAuthenticator.class, properties(
            ImapAuthenticator.CONFIG_PREFIX_PROPERTY, "com.openexchange.my.project.name.
                imap"
        ))
        // ...
    }
}
```

The OSGi property name to define is ImapAuthenticator.CONFIG_PREFIX_PROPERTY, or config.prefix, and defaults to com.openexchange.imap.auth. if not specified.

6.2 Configuration

The following configuration properties are supported:

Property	Description
<prefix>host	Hostname or IP address to connect to.
<prefix>port	Port to connect to, defaults to 143 (imap) or 993 (imaps), depending on the value of com.openexchange.<customer>.imap.secure being false or true, respectively.
<prefix>secure	Whether to use SSL/TLS (true) or not (false), defaults to false.
<prefix>starttls	Whether to use STARTTLS (when true, requires com.openexchange.<customer>.imap.secure to be set to false), defaults to false.
<prefix>secure.protocols	Which SSL/TLS protocols to use for the handshake, defaults to the protocols that supported by the JDK.
<prefix>secure.ciphersuites	Which SSL/TLS cipher suites to use for the handshake, defaults to the ones that are supported by the JDK.
<prefix>connection.timeout.millis	The timeout for connections to the IMAP server, in milliseconds, defaults to 4000 (4 seconds).
<prefix>login.timeout.millis	The timeout for logins to the IMAP server, in milliseconds, defaults to 4000 (4 seconds).

7 Ldap Client Library

Bundle Identifier	com.openexchange.ldap.client
Package(s)	open-xchange-ldap-client
Required capabilities	none
Available since	1.3.0-rev3

This package contains a library for easy, flexible and high-performance LDAP client operations. It contains and provides the open source UnboundID LDAP SDK and adds a YAML based configuration scheme on top, as well as managing pools of LDAP connections centrally for the AppSuite middleware. Bundles that need to perform LDAP operations can access the LdapClientService to retrieve a pool of LDAP connections by an identifier which is typically defined through a configuration property or hard-coded in the source.

The goals are:

- To have a central configuration in /opt/open-xchange/etc/ldap-client.d/*.yaml of all LDAP client connection pools
- To minimize the amount of code to implement in order to perform LDAP operations in custom bundles: LdapClientService.getPoolFor("xyz").getReadPool().search(...)
- To provide flexible configuration capabilities to accommodate most if not all scenarios (pools, read/write pools, round-robin, failover, ...)

It does not provide any API besides an internal one for other bundles and, as such, is to be understood as a service for other custom bundles. It can also be used standalone instead of in an OSGi container, for example for integration tests or for command-line tools, by using LdapClientConfigParser with a YAML configuration that can come from a file, or be constructed into a string, and then invoke .materialize() on the resulting configuration object.

The included UnboundID LDAP SDK also contains a very useful in-memory LDAP server that can and is being used for integration tests.

7.1 Configuration

LDAP connection pools are defined in YAML files that have a filename ending in ".yml" or ".yaml" and are located under the directory `/opt/open-xchange/etc/ldap-client.d/` like e.g. `/opt/open-xchange/etc/ldap-client.d/customer-xyz.yaml`

The bundle will read all the files that are located under that directory on startup and will register for discovering changes and new files upon reload, automatically reconfiguring and replacing pools that are affected by the changes, which is why bundles that use this service should always use `LdapClientService.getPoolFor(...)` to perform LDAP operations, and not retrieve and hold an `LdapClientPool` indefinitely (at least not beyond the scope of a "transaction").

Note that configuration files may include placeholders, using the syntax `${placeholder}`, for example `${com.openexchange.customerxyz.brand}`, and those will be replaced with their corresponding property value using the `ConfigurationService`. This may be useful for usernames and passwords in the case of authenticated connection pools.

When using LDAP to authenticate using the "search-and-bind" approach (e.g. if the username is an email address that first needs to be resolved to a uid attribute to then perform a bind operation to verify the password), the best approach should be to use two connection pools: one with authenticated connections for the search (first operation), and then one with unauthenticated connections for the bind operations. The UnboundID library does provide an operation that binds and then guarantees the "un-bind" though (which is an operation that doesn't exist in LDAP and actually means performing another bind on a different user, or doing an anonymous bind (= bind on username=="" and password==""), so using a single pool for both operations is an option too.

`/opt/open-xchange/etc/ldap-client.d/ldap-client-pools.yaml.example`

```
# The top-level key is the identifier of the pool, which can be
# any string of text and is being used by the bundles and applications
# to access that pool configuration.
# Typically, those are fixed or need to be configured in the bundles
# that use this library.
#
# When Java Security Manager support is enabled, files that are referenced
# in these configuration files must be in a directory that is already
# whitelisted, or in a subdirectory thereof, such as
# /opt/open-xchange/etc/
#
# A good candidate would be something along the lines of
# /opt/open-xchange/etc/ldap-files/
#
# Otherwise, the filename or its directory must be put into a new .list
# file in the folder
# /opt/open-xchange/etc/security/
# with e.g. the following content:
#
# file:/etc/trust.jks
#
pool1:
  trust-store:
    # SSL: path to the JKS trust store file that contains the anchors
    file: /etc/trust.jks
    # SSL: indicates whether to reject certificates if the current time
    # is outside the validity window for the certificate
    validity: true
  key-store:
    # SSL: path to the JKS client key store file that contains the key
    file: /etc/private.jks
    # SSL: password to access the keystore and the key
    password: foobar
    # SSL: alias of the key to use
    alias: key
```

```
# Configure a read/write pool with different settings for read operations
# and for write operations (i.e. different pools of LDAP servers).
# Here comes the part for the read operations:
read:
  # Use a failover cluster of two nodes:
  failover:
    - ldap1.example.com
    - ldap2.example.com
  # Pool connection management
  # --------------------------
  # When creating a connection pool, you may specify an initial number of
  # connections (pool-min) and a maximum number of connections (pool-max).
  # The initial number of connections is the number of connections that should
  # be immediately established and available for use when the pool is created.
  # The maximum number of connections is the largest number of unused connections
  # that may be available in the pool at any time.
  # Whenever a connection is needed, whether by an attempt to check out a
  # connection or to use one of the pool's methods to process an operation,
  # the pool will first check to see if there is a connection that has already
  # been established but is not currently in use, and if so then that connection
  # will be used.
  # If there aren't any unused connections that are already established, then
  # the pool will determine if it has yet created the maximum number of
  # connections, and if not then it will immediately create a new connection
  # and use it.
  # If the pool has already created the maximum number of connections, then the
  # pool may wait for a period of time (as configured using 'maxWaitTimeMillis' below,
  # which has a default value of zero to indicate that it should not wait at all)
  # for an in-use connection to be released back to the pool.
  # If no connection is available after the specified wait time (or there should
  # not be any wait time), then the pool may automatically create a new connection
  # to use if 'createIfNecessary' is true (which is the default).
  # If it is able to successfully create a connection, then it will be used.
  # If it cannot create a connection, or if 'createIfNecessary' is set to false,
  # then an error will be thrown.
  # Note that the maximum number of connections specified when creating a pool
  # refers to the maximum number of connections that should be available for use
  # at any given time.
  # If 'createIfNecessary' is set to true, then there may temporarily be more
  # active connections than the configured maximum number of connections.
  # This can be useful during periods of heavy activity, because the pool will
  # keep those connections established until the number of unused connections
  # exceeds the configured maximum.
  # If you wish to enforce a hard limit on the maximum number of connections so
  # that there cannot be more than the configured maximum in use at any time,
  # then set 'createIfNecessary' to false to indicate that the pool should not
  # automatically create connections when one is needed but none are available,
  # and you may also want to set 'maxWaitTimeMillis' to a maximum wait time to allow
  # the pool to wait for a connection to become available rather than throwing
  # an exception if no connections are immediately available.
  pool-min: 10
  pool-max: 50
  maxConnectionAgeMillis: 30000
  maxWaitTimeMillis: 500
  createIfNecessary: true
  # Specifies whether certain operations that should be retried on a newly-created
  # connection if the initial attempt fails in a manner that indicates that the
  # connection used to process the request may no longer be valid.
  # Only a single retry will be attempted for any operation.
  retryFailedOperations: true
# Here comes the part for the write operations:
write:
  host: ldap0.example.com
  pool-min: 1
  pool-max: 10
  maxConnectionAgeMillis: 60000
  maxWaitTimeMillis: 1000
  createIfNecessary: false
  retryFailedOperations: false
# Specifies whether the pool should attempt to abandon any request for which
# no response is received in the maximum response timeout period:
abandonOnTimeout: true
```

```
    # Specifies the maximum length of time in milliseconds that a connection attempt
    # should be allowed to continue before giving up.
    # A value of zero (default) indicates that there should be no connect timeout.
    connectionTimeoutMillis: 3000
    # Specifies the maximum length of time in milliseconds that an operation should
    # be allowed to block while waiting for a response from the server.
    # A value of zero indicates that there should be no timeout.
    responseTimeoutMillis: 5000
    # Specifies whether to use the SO_KEEPALIVE option for the underlying sockets
    # used by associated connections.
    keepAlive: true
    # Specifies whether to use the TCP_NODELAY option for the underlying sockets.
    tcpNoDelay: true
    # Specifies whether to operate in synchronous mode, in which at most one
    # operation may be in progress at any time on a given connection.
    # When using asynchronous mode, a background thread takes care of multiplexing
    # and dispatching all the operations on connections that are shared for
    # multiple operations.
    synchronousMode: true
    # Specifies the length of time in milliseconds between periodic background
    # health checks against the available connections in this pool.
    healthCheckIntervalMillis: 120000
    # Specifies whether associated connections should attempt to follow any
    # referrals that they encounter.
    followReferrals: true
    # Specifies the maximum number of hops that a connection should take when
    # trying to follow a referral, must be greater than zero when 'followReferrals'
    # is true.
    referralHopLimit: 1
    # Specifies the maximum size in bytes for an LDAP message that a connection
    # will attempt to read from the directory server.
    # If it encounters an LDAP message that is larger than this size, then the
    # connection will be terminated.
    # Disabled when not specified or set to 0.
    maxMessageSize: 1024

  pool2:
    # A failover pool that uses the same set of servers for read and for
    # write operations.
    failover:
      - ldap0.example.com
      - ldap1.example.com
    pool-min: 5
    pool-max: 20
    trust-store:
      file: /etc/trust.jks
    key-store:
      file: /etc/private.jks

  pool3:
    # A simple single-host setup
    host: ldap.example.com
    pool-min: 5
    pool-max: 20

  pool4:
    # A load-balancing setup that will use a round-robin algorithm to
    # select the server to which the connection should be established.
    # Any number of servers may be included, and each request will
    # attempt to retrieve a connection to the next server in the list,
    # circling back to the beginning of the list as necessary.
    # If a server is unavailable when an attempt is made to establish
    # a connection to it, then the connection will be established to
    # the next available server in the set.
    round-robin:
      - host: ldap1.example.com
        port: 10389
        responseTimeoutMillis: 5000
      - host: ldap2.example.com
        port: 10389
        responseTimeoutMillis: 12000
    pool-min: 10
    pool-max: 50

  pool5:
    # A DNS RR setup handles the case in which a given hostname may
    # resolve to multiplee IP addresses.
    # Note that while a setup like this is typically referred to as
    # "round-robin DNS", this option does not strictly require DNS (as names
    # may be resolved through alternate mechanisms like a hosts file or an
    # alternate name service), and it does not strictly require round-robin
    # use of those addresses (as alternate ordering mechanisms like
    # 'random' or 'failover' may be used).
    dns-round-robin:
      host: ldap.example.com
      # The selection mode that should be used if the hostname resolves
      # to multiple addresses.
      # Possible values:
      # - random: the order of addresses will be randomized for each attempt
      # - failover: addresses will be consistently attempted in the order
      #       they are retrieved from the name service.
      # - round-robin: connection attempts will be made in a round-robin order
      selection-mode: random
      # Only use DNS if set to 'true'.
      # If set to 'false' then the operating system's hostname resolution
      # service will be used, which may include a hosts file.
      only-dns: false
      # The maximum length of time in milliseconds to cache addresses resolved
      # from the provided hostname.
      # Caching resolved addresses can result in better performance and can
      # reduce the number of requests to the name service.
      # A value that is less than or equal to zero indicates that no caching
      # should be used.
      cache-timeout: 1440000
    pool-min: 5
    pool-max: 20

  pool6:
    # A failover pool that uses the same set of servers for read and for
    # write operations, as well as StartTLS
    failover:
      - ldap0.example.com
      - ldap1.example.com
    pool-min: 5
    pool-max: 20
    starttls: true
    trust-store:
      file: /etc/trust.jks
    key-store:
      file: /etc/private.jks
```

7.2 Metrics

Since version 1.7.2, metrics are provided for each pool (unless turned off by setting the pool configuration property `metrics` to `false`):

Metric (Prometheus syntax)	Description
`appsuite_ldap_pool_connections_available_max`	The maximum number of connections that may be available in the pool at any time
`appsuite_ldap_pool_connections_available`	The number of connections currently available for use in the pool
`appsuite_ldap_pool_connections_closed_defunct`	The number of connections that have been closed as defunct (i.e., they are no longer believed to be valid)
`appsuite_ldap_pool_connections_closed_expired`	The number of connections that have been closed as expired (i.e., they have been established for longer than the maximum connection age for the pool)

Metric (Prometheus syntax)	Description
`appsuite_ldap_pool_connections_closed_unneeded`	The number of connections that have been closed as unneeded (i.e., they were created in response to heavy load but are no longer needed to meet the current load, or they were closed when the pool was closed)
`appsuite_ldap_pool_connections_checkout_failed`	The number of failed attempts to check out a connection from the pool (including connections checked out for internal use by operations processed as part of the pool)
`appsuite_ldap_pool_connections_failed`	The number of failed attempts to create a connection for use in the connection pool
`appsuite_ldap_pool_connections_released_valid`	The number of times a valid, usable connection has been released back to the pool after being checked out (including connections checked out for internal use by operations processed within the pool)
`appsuite_ldap_pool_connections_checkout_succeeded`	The number of successful attempts to check out a connection from the pool (including connections checked out for internal use by operations processed as part of the pool)
`appsuite_ldap_pool_connections_checkout_succeeded_waiting`	The number of successful attempts to check out a connection from the pool that had to wait for a connection to become available
`appsuite_ldap_pool_connections_checkout_succeeded_new`	The number of successful attempts to check out a connection from the pool that had to create a new connection because no existing connections were available
`appsuite_ldap_pool_connections_checkout_succeeded_nowait`	The number of successful attempts to check out a connection from the pool that were able to obtain an existing connection without waiting
`appsuite_ldap_pool_connections_succeeded`	The number of connections that have been successfully created for use in conjunction with the connection pool

Each of those metrics has a tag named `pool` which is set to the identifier of that pool in the YAML configuration. For pools that are configured to have a read-only and a read-write connection pool, metrics are provided for both, separately, with a `pool` tag value set to the identifier of the pool with `.read` and `.write` appended to.

So, for the following example:

```
mypool:
  read:
    host: localhost
    port: 10389
  write:
    host: localhost
    port: 11389
```

metrics will be exported for the following tags:

- `mypool.read`
- `mypool.write`

Example in Grafana:

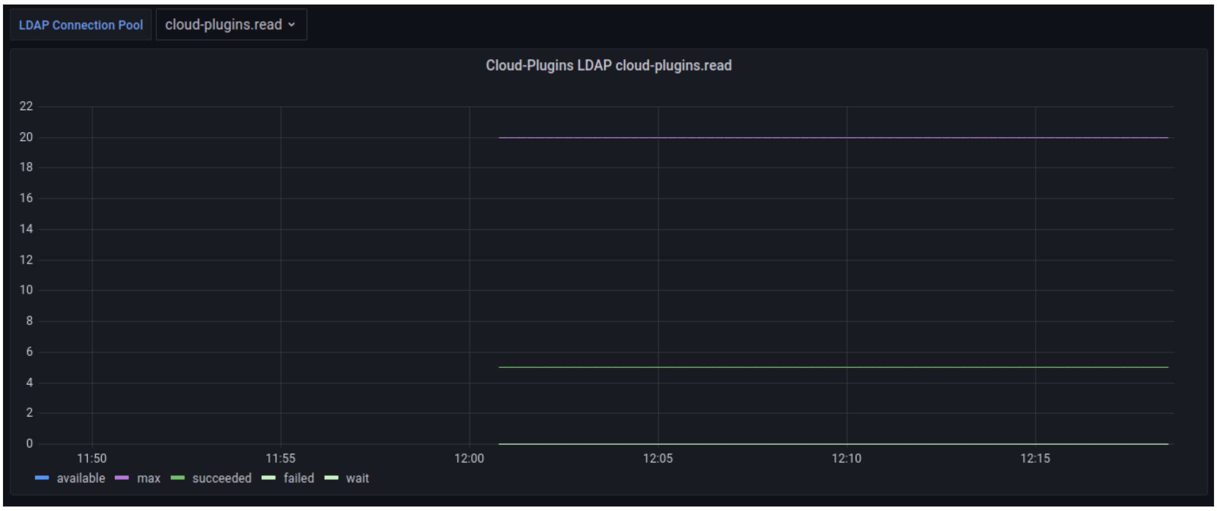


Figure 1: LDAP Client Library Metrics in Grafana

8 Main Authentication Password

Bundle Identifier	`com.openexchange.authentication.masterpassword`
Package(s)	`open-xchange-authentication-masterpassword`
Required capabilities	none
Available since	1.3.0-rev3

Warning
DO NOT USE THIS IN PRODUCTION!

This package provides an authentication implementation that verifies user passwords against a globally configured password. This implementation is only meant for migration scenarios.

8.1 Configuration

`/opt/open-xchange/etc/masterpassword-authentication.properties`

```
# Configuration file for the master password authentication plugin
#
# DO NOT USE IN PRODUCTION !
#

# The clear text password to authenticate all users.
# Mandatory.
# Example:
# com.openexchange.authentication.masterpassword.password=supersecret
com.openexchange.authentication.masterpassword.password=

# The default value for the context when it is not specified.
# Optional and defaults to using the "defaultcontext" mapping.
#com.openexchange.authentication.masterpassword.default.context=

# Whether the username portion of the login should be lowercased
# before being looked up in the user database.
# Optional and defaults to false
#com.openexchange.authentication.masterpassword.lowercase=false

# Whether the context name portion of the login should be lowercased
# before being looked up in the context database.
# Optional and defaults to false
#com.openexchange.authentication.masterpassword.lowercase.context=false

```

```
# Whether to use the complete login string as the username,
# e.g. login "foo@bar.com" -> user name "foo@bar.com" and
# context name "bar.com"
# Optional and defaults to false
#com.openexchange.authentication.masterpassword.use.full.login.info=false

# Whether to use the complete login string for the context name,
# e.g. login "foo@bar.com" -> context name "foo@bar.com"
# Optional and defaults to false
#com.openexchange.authentication.masterpassword.use.full.login.info.for.context=false
```

9 Minimal API for user data

9.1 Minimal API Framework

Bundle Identifier	`com.openexchange.plugins.minimal.api,`
	`openexchange.plugins.minimal.api.consent.rdb,`
	`com.openexchange.plugins.minimal.api.consent.clt,`
	`com.openexchange.plugins.minimal.api.json,`
	`com.openexchange.plugins.minimal.api.ws.security,`
	`com.openexchange.plugins.minimal.api.auth.jwt,`
	`com.openexchange.plugins.minimal.api.ws.mail,`
	`com.openexchange.plugins.minimal.api.ws.calendar`

Package(s)	`open-xchange-minimal-api`, `open-xchange-minimal-api-jwt`, `open-xchange-minimal-api-mail`, `open-xchange-minimal-api-calendar`
Required capabilities	none
Available since	1.3.7-rev2

The so called minimal API is meant for a temporary access to a limited dataset of the user. Temporary, because the token is only active as long as the user session is not removed from the cluster. The API is not open by default and always needs user consent to be accessed, which is tracked by date and client name. Furthermore, the API is meant to only provide readable data. The main focus is to provide data to external clients that show this data to the user while limiting the fetched fields. A widget on a second tab/iframe could be a use-case. Using a full session for this type of access would give the external client to much control over the user data which we can't track and are not able to control afterwards. By giving the user and the admin the control over the consent status, we can always restrict access again.

9.1.1 Concept

- The access is be token based.
- The API is designed for external clients.
- These tokens are only allowed to access a specific set of data via scopes/claims.
- These tokens are bound to existing OX Sessions and are invalidated as soon as the session is not in use / is removed.
- Endpoints are Restfull whenever possible.
- Either the user or the admin are able to set consent on the token handling.

Therefore so called **claims** are introduced, which represents a defined number of endpoints:

Claim	Description
`readMail`	Enables the client to read basic mail data
`readCalendar`	Enables the client to read basic calendar data

9.1.2 Security Layer

The package `open-xchange-minimal-api-jwt` contains the security layer based on signed **JWT (JSON Web Token)**. The security handling is based on a key which is saved inside the session and put into the SessionStorage. Upon security validation, the Session is fetched from the SessionStorage and the key is validated. This package also contains a Rate limiter which limits the access to the API.

9.1.3 Authentication

For authentication tokens are required which need to be fetched by the Get Token JSON endpoint like e.g:

```
GET http://{server}/appsuite/api/minimal?session={sessionId}&client={clientName}&action=
    token
```

The example JSON **GET token** response for `sessionId` **03a6bd595320422288e1767222603bce** and `clientName` **example** looks like:

```
{"data":{"token":"eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIwM2E2Y...LCJyZWFkTWFpbCI6dHJ1ZX0.9
    xQ6_YXDxZIZsZWAZzxLFRqgTptYERTl_-Qma8OUsWY"}}
```

This example **tokenResponse** translates into the following **JWT (JSON Web Token)**:

```
HEADER:ALGORITHM & TOKEN TYPE
{
  "alg": "HS256"
}

PAYLOAD:DATA
{
  "sub": "03a6bd595320422288e1767222603bce",
  "client": "example",
  "readMail": true
}

VERIFY SIGNATURE
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),

) secret base64 encoded
```

Please see in the example JSON Web Token `PAYLOAD`:

- The `sub` key contains the initial `sessionId`.
- The `client` key contains the used `clientName`.
- The claim is shown as `readMail` provided as key.

9.1.4 User Consent

As the API enables external clients(widgets) to fetch data, access to user data without consent is forbidden by default. The consent should be done for an individual claim and a client combination. It must be possible to decline this combination again. It is possible to track such changes individually.

9.1.5 Database

The user consent state is saved in 2 Database tables, which are automatically created on the user db.

9.1.6 JSON API for Client, Consent and Token Handling

Parameter	Description
`server`	Hostname or IP of the server.
`sessionId`	The session id of the user.
`clientName`	The client name configured for the user.
`claims`	Comma separated list of claims to be set. A provided claim will be written to the database as `true`. A not set claim will be written to the database as `false`. If not set at all, the default claims of the client are used and saved as `true`. If send as "" (empty string) all claims are written to the database as `false`.

9.1.6.1 Parameter

Please use the URL prefix below for following actions:

```
GET http://{server}/appsuite/api/minimal?session={sessionId}&action=
```

Action URL Suffix	Descripton
`token&client={clientName}`	Request a token for a configured client.
`set&client={clientName}&claims={claims}`	Set consent for a configured client.
`getDefaultConsent&client={clientName}`	Get default consent for a configured client.
`get`	Get all clients.
`getByName&client={clientName}`	Get a specific client.
`reset&client={clientName}`	Reset a specific client.

Please note, resetting a specific client requires following config file setting, otherwise the request will return just a 403 error:

`/opt/open-xchange/etc/minimal-api.properties`

```
com.openexchange.plugins.minimal.api.{clientName}.resetallowed=true
```

9.1.6.1.1 Example Default Consent Response

```
{
  "data": {
    "client": "example",
    "defaultConsent": true // or false
  }
}
```

9.1.6.1.2 Example Client Response

```
{
  "data": [
    {
      "client": "example",
      "name": "Special Example Client",
      "configuredClaims": [
        "readMail"
      ],
      "configuredClaimsButNotInStorage": [
      ],
      "configuredAndActiveClaims": [
        "readMail"
      ],
      "configuredAndDisabledClaims": [
      ]
    }
  ]
}
```

Error Code	Description
MIN_API-0003	A SQL error occurred.
MIN_API-0004	The client is not configured.
MIN_API-0005	The client is not allowed because the user has not yet decided to allow access.
MIN_API-0006	The client is not allowed due to not allowed claims.
MIN_API-0007	No session found for sessionid / No key found for sessionid.
MIN_API-0008	The token is missing.

9.1.6.2 JSON Error Codes

9.1.7 Accessing User Data via REST API

With the base path `/minimal/` with the token of the user as an authorization header. This header needs to be send with every request:

```
Authorization: Bearer {token}
```

Parameter	Description
`amount`	The amount from `1` to `100`.
`folder`	The folder name as string.

9.1.7.1 Parameter

9.1.7.2 General Handling When the underlying session is removed from the session storage, the token will not be able to access any data and also be considered invalid and an `MIN_API-0007` exception will be thrown.

9.1.7.3 Claim `readMail` This claim is part of the `open-xchange-minimal-api-mail` package and provides following `GET` functionality:

Path	Description
`/mail/v1/quota`	Info about the current mail quota.
`/mail/v1/info`	Info about the current mail of the user which internally is handled.
`/mail/v1/stats`	Status of total and unread count in inbox.
`/mail/v1/stats/{folder}`	Status of total and unread count in `folder`.
`/mail/v1/inbox`	Details about the last 5 by date emails in `INBOX`.
`/mail/v1/inbox/{amount}`	Details about the `amount` of mails by date emails in `INBOX`.
`/mail/v1/mails/{folder}`	Details about the last 5 by date emails in `folder`.
`/mail/v1/mails/{folder}/{amount}`	Details about the last `amount` of emails in the `folder` by date.

9.1.7.3.1 Example Quota Info Response

```
{
  "storageQuota": {
    "used": -1,
    "limit": -1
  },
  "messageQuota": {
    "used": -1,
    "limit": -1
  }
}
```

9.1.7.3.2 Example Current Mail Info Response

```
{
  "mail" : "string",
  "displayName" : "string"
}
```

9.1.7.3.3 Example Status of Mail Folder Response

```
{
  "total" : 10,
  "unread" : 5
}
```

9.1.7.3.4 Example E-Mail Details Response

```
{
    "amount" : 5 // the number of returned mails
    "mails" :[
      {
        "id": "string",
        "subject": "string",
        "from": "string",
        "seen": true,
        "arrival": 0
      },...
    ]
}
```

9.1.7.4 Claim `readCalendar` This claim is part of the `open-xchange-minimal-api-calendar` package and provides following `GET` functionality:

Path	Description
`/calendar/v1/events`	Details about the next 10 calendar events.
`/calendar/v1/events/{amount}`	Details about the next `amount` of calendar events.

9.1.7.4.1 Example Calendar Events Response

```
[
    {
        "location": "The Doctor Office",
        "created": 1581457892837,
        "description": "Need to get my ears checked.",
        "attendees": [
            {
                "uri": "mailto:john.doe@ox.com",
                "cn": "Doe, John",
                "email": "john.doe@ox.com",
                "entity": 3,
                "cuType": "INDIVIDUAL",
                "partStat": "ACCEPTED",
                "folder": "31"
            }
        ],
        "modifiedBy": {
            "uri": "mailto:john.doe@ox.com",
            "cn": "Doe, John",
            "email": "john.doe@ox.com",
            "entity": 3
        },
        "attendeePrivileges": "DEFAULT",
        "transp": "OPAQUE",
        "createdBy": {
            "uri": "mailto:john.doe@ox.com",
            "cn": "Doe, John",
            "email": "john.doe@ox.com",
            "entity": 3
        },
        "folder": "cal://0/31",
        "summary": "Doctor",
        "startDate": {
            "tzid": "Europe/Berlin",
            "value": "20200213T210000"
        },
        "organizer": {
            "uri": "mailto:john.doe@ox.com",
            "cn": "Doe, John",
            "email": "john.doe@ox.com",
            "entity": 3
        },
        "class": "PUBLIC",
        "lastModified": 1581542567786,
        "uid": "88f822e7-2f98-4c8c-8ff0-96c2f6aaf265",
        "timestamp": 1581542567786,
        "endDate": {
            "tzid": "Europe/Berlin",
            "value": "20200213T213000"
        },
        "calendarUser": {
            "uri": "mailto:john.doe@ox.com",
            "cn": "Doe, John",
            "email": "john.doe@ox.com",
            "entity": 3
        },
        "id": "3",
        "sequence": 1,
        "flags": [
            "organizer",
            "accepted"
        ]
    }
]
```

Response	Description
200	The request was successful.
400	The amount is not in range from 1 to 100.
403	The token is not alllowed to access the mail component.

9.1.7.5 HTTP Response Codes

9.1.8 Command Line Tool for Fetching the History

The package `open-xchange-minimal-api` provides the `listminimalconsenthistory` command line tool:

```
usage: listminimalconsenthistory
 -A,--adminuser <arg>         Admin username
 -c,--context <arg>           A valid context identifier
 -e,--end-time <arg>          End time in seconds since 1970-01-01
                              00:00:00 UTC. Only consent given before this
                              time is shown. If not set, all feedback
                              since -s is shown.
 -h,--help                    Prints a help text
 -i,--userid <arg>            A valid user identifier
 -n,--client-name <arg>       A valid (and used) client identifier. If not
                              set, all clients are returned
 -p,--port <arg>              The optional RMI port (default:1099)
 -P,--adminpass <arg>         Admin password
    --responsetimeout <arg>   The optional response timeout in seconds
                              when reading data from server (default: 0s;
                              infinite)
 -s,--server <arg>            The optional RMI server (default: localhost)
The command-line tool to list given consent of a user for the minimal api
```

With this tool, the admininstrator is able to fetch the given consent for a user like e.g.:

```
/opt/open-xchange/sbin/listminimalconsenthistory -c 1 -i 4
user has the following entries:
     -Claim [client=example, claim=readMail, consent=false, time=1551883129884]
     -Claim [client=example, claim=readMail, consent=true, time=1551883134048]
```

9.1.9 Configuration

/opt/open-xchange/etc/minimal-api.properties

```
# The capability to control whether or not the user is allowed to access the API
# at all
#
# Optional, default value: false
#
# Example:
# com.openexchange.capability.minimalapi=true
com.openexchange.capability.minimalapi=false

# The clients names enabled for a user
# Must be provided as a comma separated list
#
# Optional, default value: ""
#
# Must be provided as a comma separated list
#
# Example:
# com.openexchange.plugins.minimal.api.clients=exampleClient,exampleClient2
com.openexchange.plugins.minimal.api.clients=

# The user-friendly name of a client
#
# Optional, default value: ""
#
# If not set, the client identifier is returned.
#
# Example:
# com.openexchange.plugins.minimal.api.exampleClient.name=Example Preview
com.openexchange.plugins.minimal.api.[client].name=

# The claims assigned to a client
#
# Optional, default value: ""
#
# Must be provided as a comma separated list
#
# Example:
# com.openexchange.plugins.minimal.api.exampleClient.claims=readMail
com.openexchange.plugins.minimal.api.[client].claims=

# Default consent if user has not yet decided on first access
# WARNING: It might be required by law to enforce user consent
#
# Optional, default value: false
#
# Example:
# com.openexchange.plugins.minimal.api.exampleClient.defaultconsent=true
com.openexchange.plugins.minimal.api.[client].defaultconsent=false

# Maximum amount of requests per second per source IP address if the token could not be validated from cache
# May be a decimal number.
#
# Optional, default value: 1.0
# Optional, default for client: 5.0
#
# Example:
# com.openexchange.plugins.minimal.api.ratelimit.requestsPerSecond=10.0
# com.openexchange.plugins.minimal.api.ratelimit.exampleClient.maxRequestsPerSecond=10.0
com.openexchange.plugins.minimal.api.ratelimit.requestsPerSecond=1.0

# Maximal time window, in milliseconds: after a given source IP address has not accessed
# the minimal API, its number of requests per second rate is reset.
#
# Optional, default value: 300000
# Optional, default for client: 300000
#
# Example:
# com.openexchange.plugins.minimal.api.ratelimit.maxRateTimeWindow=60000
# com.openexchange.plugins.minimal.api.ratelimit.exampleClient.maxRateTimeWindow=60000
com.openexchange.plugins.minimal.api.ratelimit.maxRateTimeWindow=300000

# Strategy to use for reacting to the inability to access the API for a given source
# IP address due to surpassing the maxRequestsPerSecond rate.
#
# Format: it must be one of:
# * fail-fast
# * block
# * timeout:...
#
# fail-fast
#   if the rate limit is exceeded, the API will respond with a 401 Unauthorized
# block
#   if the rate limit is exceeded, the API will block infinitely until the rate limit
#   allows for another request to be performed
# timeout:...
#   block until the specified timeout is reached, after which the API responds with a
#   401 Unauthorized
#   if the timeout does not allow to get a new token in time, a 401 Unauthorized is
#   returned
#   The value after "timeout:" consists of a number followed by a time unit, examples:
#   - timeout:400s ---> 400 seconds
#   - timeout:1m ------>   1 minute
#   - timeout:2000ms -> 2000 milliseconds
#
# If the token could be validated and is correct, the API will not return a
# 401 Unauthorized but a 429 Too Many Requests instead.
#
# Optional, default value: timeout:250ms
# Optional, default for client: timeout:500ms
#
# Example:
# com.openexchange.plugins.minimal.api.ratelimit.strategy=timeout:1s
# com.openexchange.plugins.minimal.api.ratelimit.exampleClient.strategy=timeout:5s
com.openexchange.plugins.minimal.api.ratelimit.strategy=timeout:250ms
```

10 MX SPF Record Checker

10.1 MX SPF Record Checker Framework

Bundle Identifier	`com.openexchange.plugins.mx.checker,`
	`com.openexchange.plugins.mx.checker.json`

Package(s)	`open-xchange-plugins-mx-checker`
Required capabilities	none
Available since	1.6.3-rev4

This package contains an adapter framework designed to support multiple MX Checker connector implementations. Once the package is installed and configured, and app node is started, the plugins will be registered with the platform. It provides a json HTTP interface for MX Checker, and is registered at `plugins/mx-checker` and supports the action get.

- A connector is selected based on `com.openexchange.plugins.mx.checker.connector` config-cascade property.
- If a connector is found, mx checker logic is executed based on the incoming JSON request data. An appropriate HTTP response code, provided by the connector, will be returned to the web-browser.
- If a connector can't be found, a `501` HTTP response will be returned.

The package also provides a `MXCheckerConnector` service tracker which acts as the abstraction layer between the json API and each concrete connector service implementation.

10.1.1 Configuration

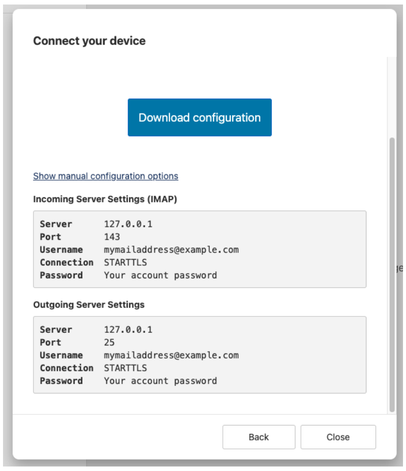
`/opt/open-xchange/etc/plugins-mx-checker.properties`

```
# Determines which connector will be used for a user
# This setting is config-cascade aware to support different implementations for each user.
# Default is <none> which means that the feature is disabled for a user
com.openexchange.plugins.mx.checker.connector=
```

11 Onboarding Wizard Change Login Details

Bundle Identifier	`com.openexchange.plugins.onboarding.maillogin`
Package(s)	`open-xchange-plugins-onboarding-maillogin`
Required capabilities	none
Available since	1.3.3-rev2

This package enables the App Suite core onboarding wizard to show a different login name. The onboarding wizard displays information for users in order to configure their applications and/or devices. It essentially displays the login and protocol specific URI to use to connect. In some cases, there is a need to override the **Username** that is used by default by the wizard. For instance, if the `"<user id>@<context id>"` format is used as login attribute, the wizard will show this incorrectly as login name which is not what users can use in order to actually login.



Connect your device

Incoming Server Settings (IMAP)

```
Server      127.0.0.1
Port        143
Username    mymailaddress@example.com
Connection  STARTTLS
Password    Your account password
```

Outgoing Server Settings

```
Server      127.0.0.1
Port        25
Username    mymailaddress@example.com
Connection  STARTTLS
Password    Your account password
```

Figure 2: example wizard

This package can be configured to use various other attributes as the string of text to display as the login. Most prominently, and what is probably used in most if not all such setups, the so-called *defaultSenderAddress* (the default email address of the user).

11.0.1 Configuration (Middleware)

The configuration settings are config cascade aware and can hence be set on a per-brand, per-context or even per-user level (the latter probably not making much sense though). The plugin covers the four different onboarding protocols that are supported by core:

- CalDAV
- CardDAV
- IMAP
- SMTP

/opt/open-xchange/etc/client-onboarding-maillogin.properties

```
# Default value for overriding the login information displayed
# in the client onboarding.
#
# Possible values:
# email
#   uses the user's defaultSenderAddress
# attr:<name>
#   uses the user's attribute <name>
# login
#   uses the user's login, which is the same as if the
#   onboarding login was not overridden by this plugin
# login_name
#   uses the loginName attribute when possible, which is only the case
#   for session based logins (IMAP, SMTP) and for protocols that do not
#   create a session (CalDAV, CardDAV, EAS), it falls back on the login
#   instead
#
# This property is config cascade aware and must be set globally
# (in this file), and can then be overridden by context and/or by
# user.
#
# Note that for this feature to be enabled, one is also required
# to set one or more the following properties, depending on the
# client onboarding dialogs that need the login information to
# be overridden by this plugin:
# com.openexchange.client.onboarding.caldav.login.customsource=true
# com.openexchange.client.onboarding.carddav.login.customsource=true
# com.openexchange.client.onboarding.mail.imap.login.customsource=true
# com.openexchange.client.onboarding.mail.smtp.login.customsource=true
#
```

```
com.openexchange.plugins.onboarding.login=login
```

12 Send SMS via Twilio

Bundle Identifier	`com.openexchange.sms.twilio`
Package(s)	`open-xchange-sms-twilio`
Required capabilities	none
Available since	1.2.0-rev4

This package is needed to send SMS messages via twilio.

⚠ **Warning**
The `open-xchange-sms-sipgate` package must **not** be installed.

12.1 Requirements

- An active Twilio account (`https://www.twilio.com/`).
- A suitable number of pre-provisioned Twilio long-codes and/or short-codes.

12.2 Configuration

`/opt/open-xchange/etc/twilio.properties`

```
# Twilio accountSID
com.openexchange.plugins.sms.twilio.accountSID.secret=ACCOUNT_SID

# Twilio auth token
com.openexchange.plugins.sms.twilio.authtoken.secret=AUTH_TOKEN

# Twilio Message Service SID
com.openexchange.plugins.sms.twilio.messageservicesid.secret=SERVICE_SID

# Max message length. 1600 characters is Twilio's maximum
com.openexchange.plugins.sms.twilio.maxlength=1600
```

13 Server Metrics

13.1 Metrics for HTTP

Bundle Identifier	`com.openexchange.metrics.http`
Package(s)	`open-xchange-metrics-http`
Required capabilities	none
Available since	1.3.6-rev1

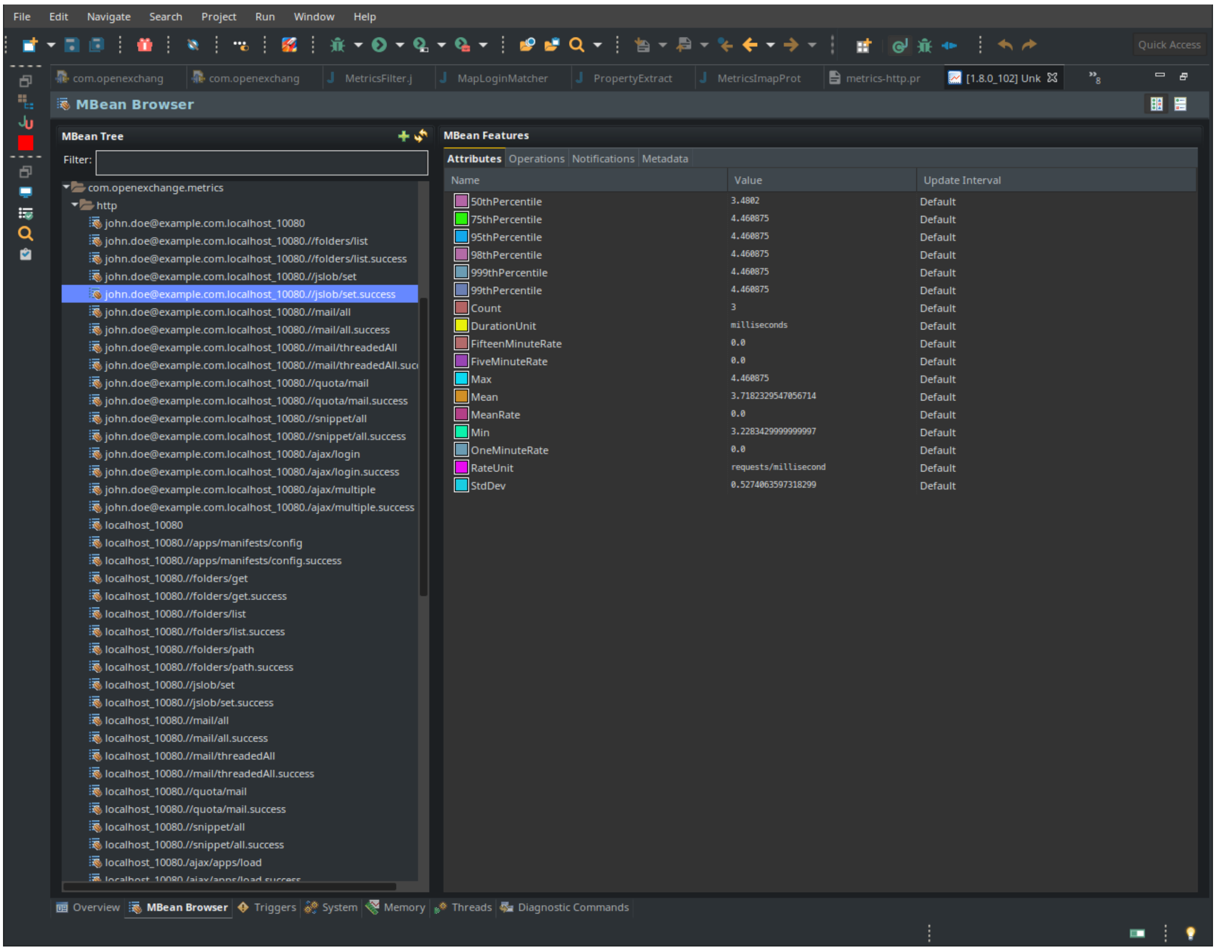
This package contains metrics for any HTTP requests that are sent to the middleware. It is highly configurable and uses the core framework `com.openexchange.metrics`.

It supports the following features:

- Configure which information is used to construct metrics (request headers, request parameters, sessions, logins, request paths, context identifier, ...).
- Blacklist or whitelist of URL paths for which to collect metrics.
- Optional aggregation of metrics.
- Optional additional metrics for specific users.
- Never collects metrics for the Jolokia servlet (reads out the Jolokia servlet URI and keeps it from collecting metrics).

Metrics are made available through JMX and Jolokia (when enabled) to be collected by various monitoring systems. They are available under the object name:

```
com.openexchange.metrics;type=http
```



```
com.openexchange.metrics
  http
    john.doe@example.com.localhost_10080
    john.doe@example.com.localhost_10080.//folders/list
    john.doe@example.com.localhost_10080.//folders/list.success
    john.doe@example.com.localhost_10080.//jslob/set
    john.doe@example.com.localhost_10080.//jslob/set.success
    john.doe@example.com.localhost_10080.//mail/all
    john.doe@example.com.localhost_10080.//mail/all.success
    john.doe@example.com.localhost_10080.//mail/threadedAll
    john.doe@example.com.localhost_10080.//mail/threadedAll.suc
    john.doe@example.com.localhost_10080.//quota/mail
    john.doe@example.com.localhost_10080.//quota/mail.success
    john.doe@example.com.localhost_10080.//snippet/all
    john.doe@example.com.localhost_10080.//snippet/all.success
    john.doe@example.com.localhost_10080./ajax/login
    john.doe@example.com.localhost_10080./ajax/login.success
    john.doe@example.com.localhost_10080./ajax/multiple
    john.doe@example.com.localhost_10080./ajax/multiple.success
    localhost_10080
    localhost_10080.//apps/manifests/config
    localhost_10080.//apps/manifests/config.success
    localhost_10080.//folders/get
    localhost_10080.//folders/get.success
    localhost_10080.//folders/list
    localhost_10080.//folders/list.success
    localhost_10080.//folders/path
    localhost_10080.//folders/path.success
    localhost_10080.//jslob/set
    localhost_10080.//jslob/set.success
    localhost_10080.//mail/all
    localhost_10080.//mail/all.success
    localhost_10080.//mail/threadedAll
    localhost_10080.//mail/threadedAll.success
    localhost_10080.//quota/mail
    localhost_10080.//quota/mail.success
    localhost_10080.//snippet/all
    localhost_10080.//snippet/all.success
    localhost_10080./ajax/apps/load
    localhost_10080./ajax/apps/load.success
```

Name	Value	Update Interval
50thPercentile	3.4802	Default
75thPercentile	4.460875	Default
95thPercentile	4.460875	Default
98thPercentile	4.460875	Default
999thPercentile	4.460875	Default
99thPercentile	4.460875	Default
Count	3	Default
DurationUnit	milliseconds	Default
FifteenMinuteRate	0.0	Default
FiveMinuteRate	0.0	Default
Max	4.460875	Default
Mean	3.7182329547056714	Default
MeanRate	0.0	Default
Min	3.2283429999999997	Default
OneMinuteRate	0.0	Default
RateUnit	requests/millisecond	Default
StdDev	0.5274063597318299	Default

Figure 3: metrics for http

It is disabled by default for performance reasons, and must be enabled explicitly by modifying the configuration file. All configuration changes can be applied through configuration reloading, there is no need to restart.

13.1.1 Configuration

/opt/open-xchange/etc/metrics-http.properties

```
#
# The following property defines the various elements to use to compose the names of
# the metrics, to determine how to group them and what to see.
#
# The elements are separated by dots (".") and parsed individually, then replaced by
# their respective value for each inbound HTTP request to determine the name of
# the metric to update.
#
# Note that not all elements necessarily always result in a value as some are only
# present for specific types of HTTP requests, and others are optional (for example
# all the user information related ones that are only available when the HTTP request
# is authenticated or used in the context of an established Open-Xchange session).
# Values that are not available are skipped in the resulting name of the metric.
```

```
#
# For each component, here are the possible values to specify in this property:
# status
# ======
# Will be replaced by "success" or "error" depending on the result, for example:
# /api/rest/x/y/z -> success
#
# path
# ====
# If the HTTP is an AJAX API call, it will be replaced by "//module/action", and if not
# (e.g. accessing a servlet instead), it will be replaced with the servlet path.
#
# Examples:
# /ajax/folders?action=get&id=1,2,4 -> //folders/get
# /rest/api/x/y/z -------------------> /rest/api/x/y/z
#
# info
# ====
# Will be replaced with the servlet path info, i.e. the part of the URL that is behind
# the servlet path.
#
# Examples:
# /rest/api/users/john.doe@example.com -> john.doe@example.com
#
# session
# =======
# The value "session", "session_id" or "sessionid" will be replaced by the Open-Xchange session
# identifier, if applicable.
# For HTTP operations that are not authenticated, it will be left out.
#
# context_id
# ==========
# The value "context_id" or "cid" will be replaced by the numeric context identifier of the
# user, if applicable.
# For HTTP operations that are not authenticated, it will be left out.
#
# user_id
# =======
# The value "user_id" or "cid" will be replaced by the numeric user identifier of the
# user within the context, if applicable.
# For HTTP operations that are not authenticated, it will be left out.
#
# login
# =====
# The value "login" will be replaced by the login the user entered to authenticate or the
# user identifier provided by an SSO mechanism, if applicable.
# For HTTP operations that are not authenticated, it will be left out.
#
# property(module)
# ================
# Will be replaced by the AJAX API module, if applicable.
#
# property(action)
# ================
# Will be replaced by the AJAX API module action, if applicable.
#
# header(...)
# ===========
# Will be replaced by the value of an HTTP request header, the name of the header
# being specified between the parentheses.
# Note that header names are case sensitive.
#
# Example:
# header(Host).path -> appsuite01.example.com.//folders/list
#
# parameter(...)
# ==============
# Will be replaced by the value of an HTTP request parameter, the name of the
# parameter being specified between the parentheses.
#
```

```
# Example:
# header(Host).parameter(app).path -> appsuite01.example.com.io.ox/mail.//folders/list
#
# cookie(...)
# ===========
# Will be replaced by the value of a cookie present in the HTTP request, the name of the
# cookie being specified between the parentheses.
#
# session(...)
# ============
# Will be replaced by the value of a parameter present in the user's Open-Xchange session,
# the name of the session parameter being specified between the parentheses.
#
# text(...)
# =========
# Specifies text that will be used as-is.
#
com.openexchange.metrics.http.elements=path.status

# When aggregation is enabled (by setting this value to true), each element as configured
# by the property com.openexchange.metrics.http.elements will be a metric in its own right,
# and aggregated accordingly to its path.
# Without aggregation, each metric is "flat".
#
# For example, with the following configuration
#   com.openexchange.metrics.http.elements=header(Host).path.status
#   com.openexchange.metrics.http.aggregation=true
# each element will be a metric, namely:
# 1. header(Host)
# 2. header(Host).path
# 3. header(Host).path.status
#
# Specifically, results will look along the lines of the following, each being a metric:
# - appsuite01.example.com
# - appsuite01.example.com.//folders/list
# - appsuite01.example.com.//folders/list.success
#
# Each of those metrics except for the last one will be aggregating the measurements
# of their parent metrics.
#
com.openexchange.metrics.http.aggregation=false

# List of logins for which to create specific metrics.
# In order to be able to track and aggregate the metrics of specific users, the
# following property can be set to a (full) login name as entered by the user when
# authenticating or as provided by an SSO system if applicable.
#
# For each of the logins specified through this property, an additional set
# of metrics will be created, prefixing the elements that are defined in
# com.openexchange.metrics.http.elements
# with the login value.
#
# For example, the following configuration
#   com.openexchange.metrics.http.elements=header(host).path.status
#   com.openexchange.metrics.http.aggregation=true
#   com.openexchange.metrics.http.logins=jdoe@example.com
# will result in the following list of metrics:
# 1. header(Host)
# 2. header(Host).path
# 3. header(Host).path.status
# 4. login
# 5. login.header(Host)
# 6. login.header(Host).path
# 7. login.header(Host).path.status
#
# Specifically, results will look along the lines of the following, each being a metric:
# - appsuite01.example.com
# - appsuite01.example.com.//folders/list
# - appsuite01.example.com.//folders/list.success
# - jdoe@example.com
# - jdoe@example.com.appsuite01.example.com
# - jdoe@example.com.appsuite01.example.com.//folders/list
# - jdoe@example.com.appsuite01.example.com.//folders/list.success
#
# Without aggregation, the following configuration
#   com.openexchange.metrics.http.elements=header(host).path.status
#   com.openexchange.metrics.http.aggregation=false
#   com.openexchange.metrics.http.logins=jdoe@example.com
# will result in this list of metrics instead:
# 1. header(Host).path.status
# 2. login.header(Host).path.status
#
# Note that if this property is commented out (not set) or left empty,
# no such additional per-login metrics will be created, which is the default
# behavior.
#
# Multiple logins may be specified, either by separating them with whitespaces
# and/or commas, e.g.:
# com.openexchange.metrics.http.logins=john.doe@example.com, jane.doe@example.com
# or by specifying multiple properties as follows:
# com.openexchange.metrics.http.logins.1=john.doe@example.com
# com.openexchange.metrics.http.logins.2=jane.doe@example.com
# (both may also be combined).
#
# Furthermore, it is possible to use regular expressions and wildcards:
# - if a login contains * or ?, it is understood to be a wildcard
# - if a login is enclosed in /.../ or /.../i (case insensitive), it is understood
#   to be a regular expression
# Examples:
# com.openexchange.metrics.http.logins=*@example.com, /^j(ohn|ane)\.doe@example\.cm$/
#
# Being a wildcard, the following value would match all logins:
# com.openexchange.metrics.http.logins=*
#
com.openexchange.metrics.http.logins=

# List of paths and path patterns for which to maintain metrics.
#
# The following property specifies discrete paths, path wildcard patterns, or
# regular expressions that will be matched against the HTTP request paths, and
# only those that match will have metrics.
#
# If the property value contains * or ?, it will be understood as a wildcard pattern.
# If it starts with / and ends with / or /i (cae insensitive), it will be understood
# as a regular expression.
# If it is neither of those, it will be interpreted as an exact (string comparison) value.
#
# To enable metric collection for all URLs, use the following value:
# com.openexchange.metrics.http.path=*
#
# If the value is not defined or empty, no metrics will be collected:
# com.openexchange.metrics.http.path=
#
# Example:
# com.openexchange.metrics.http.path.1=/^/appsuite/.+/(boot|precore)\.js$/
# com.openexchange.metrics.http.path.2=/appsuite/api/apps/manifests
# com.openexchange.metrics.http.path.3=/appsuite/api/mail
#
com.openexchange.metrics.http.path=

# The behavior of the path matching above can be configured with the following property.
# Possible values:
# - whitelist: any URL path that matches one of the URL patterns configured
#   using com.openexchange.metrics.http.path will be measured with metrics;
#   any URL path that does not, will not be measured with metrics
# - blacklist: any URL path that does not matches one of the URL patterns configured
#   using com.openexchange.metrics.http.path will be measured with metrics
#
# When omitted, left empty or invalid, the default mode is whitelist
#
# Example:
# com.openexchange.metrics.http.path.mode=blacklist
#
```

```
com.openexchange.metrics.http.path.mode=whitelist
```

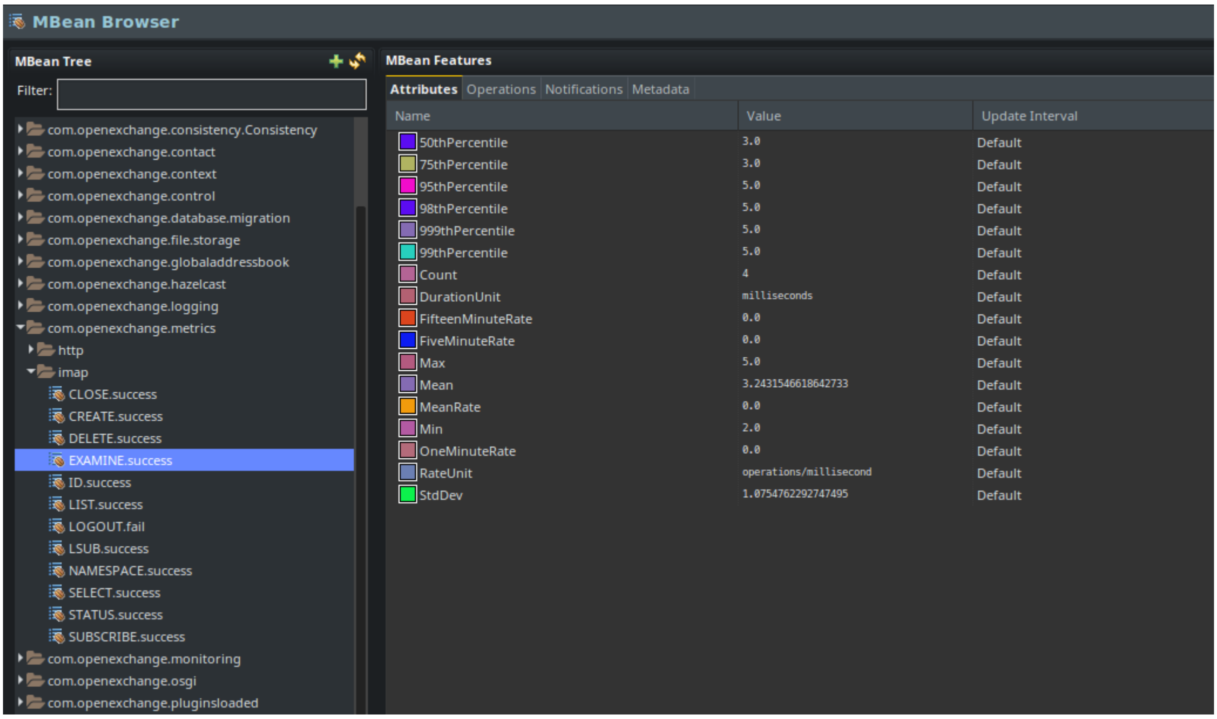
13.2 Metrics for IMAP

Bundle Identifier	`com.openexchange.metrics.imap`
Package(s)	`open-xchange-metrics-imap`
Required capabilities	none
Available since	1.3.6-rev1

This package contains metrics for all IMAP operations that are performed by the middleware. It uses the core framework `com.openexchange.metrics`.

Metrics are published through JMX and Jolokia (when enabled) with a metric object for each operation, as it makes little sense to compare LIST with FETCH, for example. They are available under the object name:

```
com.openexchange.metrics;type=imap
```



Name	Value	Update Interval
50thPercentile	3.0	Default
75thPercentile	3.0	Default
95thPercentile	5.0	Default
98thPercentile	5.0	Default
999thPercentile	5.0	Default
99thPercentile	5.0	Default
Count	4	Default
DurationUnit	milliseconds	Default
FifteenMinuteRate	0.0	Default
FiveMinuteRate	0.0	Default
Max	5.0	Default
Mean	3.2431546618642733	Default
MeanRate	0.0	Default
Min	2.0	Default
OneMinuteRate	0.0	Default
RateUnit	operations/millisecond	Default
StdDev	1.0754762292747495	Default

Figure 4: metrics for imap

It is disabled by default for performance reasons, and must be enabled explicitly by modifying the configuration file. To avoid slowing down all IMAP operations performed by the middleware, metrics are computed and updated asynchronously. The number of worker threads in charge of doing so is configurable. All configuration changes can be applied through configuration reloading, there is no need to restart.

13.2.1 Configuration

`/opt/open-xchange/etc/metrics-imap.properties`

```
# Configure whether to enable metrics for IMAP operations.
# When this property is omitted (commented out) or set to false, or empty,
# IMAP metrics will not be collected.
com.openexchange.metrics.imap.enable=false

# The number of threads to use to process IMAP operation results,
# updating metrics.
com.openexchange.metrics.imap.threads=2
```

14 Sql Client Library

Bundle Identifier	com.openexchange.sql.client
Package(s)	open-xchange-sql-client
Required capabilities	none
Available since	plugins-1.4.5-rev3

This package provides SQL pools to any component using them in the middleware. It is not config-cascade aware, but it doesn't need to be. By default, all *.yaml or *.yml files are read and interpreted within /opt/open-xchange/etc/sql-client.d. Internally, the HikariCP is used to manage those pools.

14.1 Configuration

```
# Comma seperated list of drivers to read into the system
# As the sql-client is very early, it may happen that the excpected driver is not yet
    registered.
# To work around this issue, the following list of drivers will be read before any
    connection is
# created.
#
# Default: com.mysql.jdbc.Driver
com.openexchange.sql.client.drivers=com.mysql.jdbc.Driver

# The sql-client.d folder can be changed via
# openexchange.sql.client.dir=
```

14.2 Sample Configuration

/opt/open-xchange/etc/sql-client.d/sql-client-pools.yaml.example

```
# The top-level key is the identifier of the pool, which can be
# any string of text and is being used by the bundles and applications
# to access that pool configuration.
# Typically, those are fixed or need to be configured in the bundles
# that use this library.
#
# When Java Security Manager support is enabled, files that are referenced
# in these configuration files must be in a directory that is already
# whitelisted, or in a subdirectory thereof, such as
# /opt/open-xchange/etc/
#
# A good candidate would be something along the lines of
# /opt/open-xchange/etc/sql-files/
#
# Otherwise, the filename or its directory must be put into a new .list
# file in the folder
# /opt/open-xchange/etc/security/
# with e.g. the following content:
#
# file:/etc/trust.jks
#
# For a complete list of property values, read https://github.com/brettwooldridge/HikariCP
pool1:
  # This is the name of the DataSource class provided by the JDBC driver.
  # Consult the documentation for your specific JDBC driver to get this class name, or see the table below.
  # Note XA data sources are not supported. XA requires a real transaction manager like bitronix.
  # Note that you do not need this property if you are using jdbcUrl for "old-school" DriverManager-based JDBC driver configuration.
  # Default: none
  dataSourceClassName: com.mysql.jdbc.jdbc2.optional.MysqlDataSource
  # This property directs HikariCP to use "DriverManager-based" configuration.
  # We feel that DataSource-based configuration (above) is superior for a variety of reasons (see below), but for many deployments there is little significant difference.
  # When using this property with "old" drivers, you may also need to set the driverClassName property, but try it first without.
  # Note that if this property is used, you may still use DataSource properties to configure your driver and is in fact recommended over driver parameters specified in the URL itself.
  # Default: none
  jdbcUrl: jdbc:mysql://mysql.example.com
  # This property sets the default authentication username used when obtaining Connections from the underlying driver.
  # Note that for DataSources this works in a very deterministic fashion by calling DataSource.getConnection(*username*, password) on the underlying DataSource.
  # However, for Driver-based configurations, every driver is different.
  # In the case of Driver-based, HikariCP will use this username property to set a user property in the Properties passed to the driver's DriverManager.getConnection(jdbcUrl, props) call.
  # If this is not what you need, skip this method entirely and call addDataSourceProperty("username", ...), for example.
  # Default: none
  username: user
  # sets the password of the connection
  password: secret

 pool2:
  jdbcUrl: jdbc:mysql://mysql.example.com
  # This property controls the maximum number of milliseconds that a client (that's you) will wait for a connection from the pool.
  # If this time is exceeded without a connection becoming available, a SQLException will be thrown.
  # Lowest acceptable connection timeout is 250 ms.
  # Default: 30000 (30 seconds)
  connectionTimeout: 30000
  # This property controls the maximum amount of time that a connection is allowed to sit idle in the pool.
  # This setting only applies when minimumIdle is defined to be less than maximumPoolSize. Idle connections will not be retired once the pool reaches minimumIdle connections.
  # Whether a connection is retired as idle or not is subject to a maximum variation of +30 seconds, and average variation of +15 seconds.
  # A connection will never be retired as idle before this timeout.
  # A value of 0 means that idle connections are never removed from the pool.
  # The minimum allowed value is 10000ms (10 seconds).
  # Default: 600000 (10 minutes)
  idleTimeout: 600000
  # This property controls the maximum lifetime of a connection in the pool. An in-use connection will never be retired, only when it is closed will it then be removed.
  # On a connection-by-connection basis, minor negative attenuation is applied to avoid mass-extinction in the pool.
  # We strongly recommend setting this value, and it should be several seconds shorter than any database or infrastructure imposed connection time limit.
  # A value of 0 indicates no maximum lifetime (infinite lifetime), subject of course to the idleTimeout setting.
  # Default: 1800000 (30 minutes)
  maxLifetime: 1800000
  # This property controls the minimum number of idle connections that HikariCP tries to maintain in the pool.
  # If the idle connections dip below this value and total connections in the pool are less than maximumPoolSize, HikariCP will make a best effort to add additional connections quickly and efficiently.
  # However, for maximum performance and responsiveness to spike demands, we recommend not setting this value and instead allowing HikariCP to act as a fixed size connection
```

```
      pool.
  # Default: same as maximumPoolSize
  minimumIdle: 0
  # This property controls the maximum size that the pool is allowed to reach, including
      both idle and in-use connections.
  # Basically this value will determine the maximum number of actual connections to the
      database backend. A reasonable value for this is best determined by your execution
      environment.
  # When the pool reaches this size, and no idle connections are available, calls to
      getConnection() will block for up to connectionTimeout milliseconds before timing
      out.
  # Default: 10
  maximumPoolSize: 10

# The following example shows how to provide additional dataSource properties to the pool
    by using the dataSourceProperties key.
# The DataSource will be started with all key-value pairs added.
pool3:
  jdbcUrl: jdbc:mysql://mysql.example.com
  username: user
  password: secret
  dataSourceProperties:
    useUnicode: true
    characterEncoding: UTF-8
    autoReconnect: false
    useServerPrepStmts: false
    useTimezone: true
    serverTimezone: UTC
    connectTimeout: 15000
    socketTimeout: 15000
    useSSL: false
    requireSSL: false
    verifyServerCertificate: false
    enabledTLSProtocols: TLSv1,TLSv1.1,TLSv1.2
```

15 Trusted Identity for external systems

15.1 Trusted Identity Provider

Bundle Identifier	`com.openexchange.plugins.trustedidentity`
Package(s)	`open-xchange-plugins-trusted-identity`
Required capabilities	none
Available since	1.6.4-rev2

This package extends the OX HTTP API for the creation of encrypted or signed JSON Web Tokens (JWT) containing authenticated user details. With those tokens the browser can link out to a external system while providing authenticated user details in a secure manner. For the cryptography we choose ECDSA keys because it is currently the most fitting standard. The external system must be able to validate the request by using standard protocols.

ℹ **Info**

Note that there is an additional package in Cloud-Plugins that provides support for storing signature keys in LDAP instead of on the filesystem.

15.1.1 JSON Web Token

Signed JSON Web Tokens contain a number of key-value pairs in their header part, as defined by RFC 7517. We include the following headers:

Attribute	Meaning	Description
`kid`	Key ID	References the public key to use to verify the signature of the JWS, in a textual form that has no particular convention, but is supposed to help the peer implementation to pick the correct public key to verify the signature.
`typ`	JWS Type	Always "JWT", mandatory as of the specification.
`alg`	Algorithm	As specified in JSON Web Signature and Encryption Algorithms. We currently support ES256, ES384 and ES512.
`x5t#S256`	SHA-256 thumbprint	Thumbprint of the public key: *base64url-encoded SHA-256 thumbprint (a.k.a. digest) of the DER encoding of an X.509 certificate RFC5280*.

```
{
  "x5t#S256": "BzyJzPOeUJcJH2Csu1fy7ZrLDvSOIbCukwmhgMtIKws",
  "kid": "key01",
  "typ": "JWT",
  "alg": "ES256"
}
```

The payload inside the JWT is JSON and contains "claims" (keys and values) that are either well-defined (in bold) or custom (not in bold):

Attribute	Meaning	Description
sub	subject	The login the user used to authenticate the session.
iss	issuer	A configured string of text, e.g. Open-Xchange (configurable, see below: `com.openexchange.plugins.trustedidentity.issuer`).
`contextId`		The numeric context identifier of the user.
`userId`		The numeric user identifier of the user.
exp	expiry timestamp	Instant in time after which the token should be discarded (configurable, see below: `com.openexchange.plugins.trustedidentity.expiration`).
iat	issued at timestamp	Instant in time at which the token was generated by the middleware.

```
{
  "sub": "three@fourfivesix.example.com",
  "iss": "the issuer",
  "contextId": 456,
  "upsell": ["guard","storage"],
  "source": "upgrade_to_pro_plus",
  "exp": 1617260650,
  "iat": 1617260350,
  "userId": 3
}
```

When the token is signed and encrypted (using the EC-256 algorithm which performs a ECDH to derive a symmetric key that is then used to encrypt with AES key wrap (A256GCM: AES in Galois/-Counter Mode (NIST-800-38D)), hence using a 256-bit key) like e.g.:

```
eyJlcGsiOnsia3R5IjoiRUMiLCJjcnYiOiJQLTI1NiIsIngiOiJpSWF1NU5xRlo3V3JyN1ZnT1JYelZFUH [...]
[...] TkJl-2TygzF8lXMafSqaqy64Cq6n52X_ePZgt3UfZ9WS1nmFx0OO9sbsyb2J.tKZ_5b5LP2YnUr_h2gi6Cw
```

When the tocken is just signed (without encryption) using the EC-256 algorithm which performs a SHA-256 hash like e.g.:

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJFUzI1NiJ9 [...]
[...] n9Sed-CAawcs-t4qTxL8UKZS5hnKFEjb8Fiet27nJwvTfEQdaxAm6SvSiNc3oUBd1PCORpAEEl_ZTVF8A
```

15.1.2 HTTP API

Parameter	Description
`server`	Hostname or IP of the server.
`sessionId`	The session id of the user.
`data`	A JSON Object with key-value pairs that are included as-is in the token response.

Please use the URL prefix below for following actions:

```
PUT http://{server}/appsuite/api/trust?session={sessionId}&
```

Action URL Suffix	Descripton
`action=sign&data={data}`	Create base64url encoded signed JWT.
`action=encrypt&data={data}`	Create base64url encoded signed and encrypted JWT.

15.1.2.1 Example Sign UI Usage for Upsell

```
require(['io.ox/core/http']).then(function (http) { http.PUT({module: 'trust', params: {
    action: 'sign'}, data: {upsell:['guard'], source: 'test'}}).then(function (resp) {
    console.log(resp.token) }) });
```

15.1.2.2 Example Encryt UI Usage for Upsell

```
require(['io.ox/core/http']).then(function (http) { http.PUT({module: 'trust', params: {
    action: 'encrypt'}, data: {upsell:['guard'], source: 'test'}}).then(function (resp) {
    console.log(resp.token) }) });
```

15.1.3 Configuration

/opt/open-xchange/etc/trustedidentity.properties

```
# URI to the private and public key resource to use to sign JWTs.
#
# The format of the URi epends on the scheme and driver.
# The "file" scheme is always supported.
#
# Format: file:<algorithm>:<path>[#<keyid>]
#
# Algorithm may either be "auto" in which case the signing algorithm will be inferred
# from the EC curve OID within the encoded private key part in the file, or be explicitly
# one of the supported values:
# - ES256: ECDSA using P-256 curve and SHA-256 hash algorithm
# - ES384: ECDSA using P-384 curve and SHA-384 hash algorithm
# - ES512: ECDSA using P-521 curve and SHA-512 hash algorithm
#
# Note that for the time being, only ECDSA keys are supported.
#
# The key id may be set as the fragment part of the URI: if set, will be stored as a kid (
    key id)
# claim in the JWT header, which identifies the key in some form that is understandable
    for consumers
# of the JWT token.
# Optional, does not set the kid claim when absent.
#
# The path is a fully qualified filesystem path to the private key PEM file to use for
    signing.
# It should also contain the certificate (public key part) in order to include the
# x5t#S256 (X.509 certificate SHA-256 thumbprint) in the signed token.
#
# Content of the file:
```

```
# -----BEGIN EC PRIVATE KEY-----
# MIGHAgEAMBMGByqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQgyGdEuJcaHla0CDtX
# ...
# Jvb9wIBomk0sFr++dEnvM97Sm3G+c8wkqL0+WFBRwTw79sQioT3V0MVV
# -----END EC PRIVATE KEY-----
# -----BEGIN EC PUBLIC KEY-----
# MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEqxHR/v8D3NktT/EfE5Mq2dvlIZ6H
# QCb2/cCAaJpDrBa/vnRJ7zPe0ptxvnPMJKi9PlhQUcE80/bEIqE91TjFVQ==
# -----END EC PUBLIC KEY-----
#
# The type (specified after BEGIN and END) in the PEM headers must be one of:
# - for the mandatory private key: PRIVATE KEY, EC PRIVATE KEY
# - for the optional public key: CERTIFICATE, PUBLIC KEY, EC PUBLIC KEY
#
# Examples:
# com.openexchange.plugins.trustedidentity.key=file:auto:/opt/open-xchange/etc/trustedidentity.pem#ox-trust-key-2021-1
# com.openexchange.plugins.trustedidentity.key=file:ES256:/opt/open-xchange/etc/trustedidentity.pem
#
# Mandatory, there is no default value.
com.openexchange.plugins.trustedidentity.key=

# The issuer (iss) string to include in the signed JWT.
# Describes this App Suite instance in its role as an authority.
#
# Mandatory, has no default value.
#
# Example:
# com.openexchange.plugins.trustedidentity.issuer=Open-Xchange
com.openexchange.plugins.trustedidentity.issuer=

# Expiration duration: the signed JWT contains a standard claim field
# "exp" that defines when the validity of the JWT should expire.
# The following configuration property configures how long that expiration
# time frame should be, always in addition to the current timestamp as
# of the system clock.
# e.g. "5m" will produce an expiration timestamp that is 5m in the future
#
# Format: <duration>[h|m|s|ms]
#
# Example:
# com.openexchange.plugins.trustedidentity.expiration=30m
#
# Optional, the default value is "5m" (5 minutes)
#
com.openexchange.plugins.trustedidentity.expiration=5m

# Public key file (PEM) location on disk.
#
# This is the public key to use for encrypting JWTs. That public key must be
# provided to us by the peer or customer that will receive the encrypted
# JWT, as they will be able to decrypt it using their private key part.
#
# Note tha this property is config-cascade aware.
#
# Example:
# com.openexchange.plugins.trustedidentity.peer.publicKeyFile=/opt/open-xchange/keys/customer1-pubkey1.pem
#
# This configuration setting is mandatory and has no default value.
# When left empty, it disables encryption.
com.openexchange.plugins.trustedidentity.peer.publicKeyFile=

# Algorithm to use to encrypt the JWT.
#
# The supported algorithms depend on the type of the public key.
#
# For an EC key:
#
# - ECDH-ES: Elliptic Curve Diffie-Hellman Ephemeral Static (RFC 6090) key agreement using the
#            Concat KDF, as defined in section 5.8.1 of NIST.800-56A, with the agreed-upon key
#            being used directly as the Content Encryption Key (CEK) (rather than being used to
#            wrap the CEK).
#
# - ECDH-ES+A128KW: Elliptic Curve Diffie-Hellman Ephemeral Static key agreement per "ECDH-ES",
#            but where the agreed-upon key is used to wrap the Content Encryption Key (CEK) with
#            the "A128KW" function (rather than being used directly as the CEK).
#
# - ECDH-ES+A192KW: Elliptic Curve Diffie-Hellman Ephemeral Static key agreement per "ECDH-ES",
#            but where the agreed-upon key is used to wrap the Content Encryption Key (CEK) with
#            the "A192KW" function (rather than being used directly as the CEK).
#
# - ECDH-ES+A256KW: Elliptic Curve Diffie-Hellman Ephemeral Static key agreement per "ECDH-ES",
#            but where the agreed-upon key is used to wrap the Content Encryption Key (CEK) with
#            the "A256KW" function (rather than being used directly as the CEK).
#
# For an RSA key:
#
# - RSA-OAEP-256: RSAES using Optimal Asymmetric Encryption Padding (OAEP) (RFC 3447), with the
#            SHA-256 hash function and the MGF1 with SHA-256 mask generation function.
#
# Note tha this property is config-cascade aware.
#
# Example:
# com.openexchange.plugins.trustedidentity.peer.algorithm=ECDH-ES+A256KW
#
# The property is optional and defaults to either ECDH-ES for EC keys, or
# to RSA-OAEP-256 for RSA keys.
com.openexchange.plugins.trustedidentity.peer.algorithm=

# Encryption Method to use to encrypt the JWT.
#
# The supported methods are as follows:
#
# - A128GCM: AES in Galois/Counter Mode (GCM) (NIST.800-38D) using a 128 bit key
# - A192GCM: AES in Galois/Counter Mode (GCM) (NIST.800-38D) using a 192 bit key
# - A256GCM: AES in Galois/Counter Mode (GCM) (NIST.800-38D) using a 256 bit key
#
# Note tha this property is config-cascade aware.
#
# Example:
# com.openexchange.plugins.trustedidentity.peer.encryptionMethod=A256GCM
#
# The property is optional and defaults to A256GCM
com.openexchange.plugins.trustedidentity.peer.encryptionMethod=

# Peer public key time-to-live in cache.
#
# Public keys are loaded from PEM files on-demand and are then cached for a configurable
# amount of time before being loaded again.
#
# Format: <duration>[w|d|h|m|s|ms]
#
# Example:
# com.openexchange.plugins.trustedidentity.peer.publicKeyCacheTtl=5d
#
# The property is optional and defaults to 1d (1 day)
com.openexchange.plugins.trustedidentity.peer.publicKeyCacheTtl=
```

15.1.3.1 Example Key Configuration

For a configuration that retrieves the signing key from disk:

- **Filename**: `/opt/open-xchange/etc/trust.pem`
- **Public Key ID**: `ox-trust-example-2021`

The file contains both the public key and the private key in PEM (PKCS#8) format, for example this EC key for ES256 like e.g.:

`trust.pem`

```
-----BEGIN PRIVATE KEY-----
MIGHAgEAMBMGByqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQgyGdEuJcaHla0CDtX
XN3PQq7EN9lxhgUm2D8MlAji0rWhRANCAASrEdH+/wPc2S1P8R8TkyrZ2+UhnodA
Jvb9wIBomk0sFr++dEnvM97Sm3G+c8wkqL0+WFBRwTw79sQioT3VOMVV
-----END PRIVATE KEY-----
-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEqxHR/v8D3NktT/EfE5Mq2dvlIZ6H
QCb2/cCAaJpDrBa/vnRJ7zPe0ptxvnPMJKi9PlhQUcE8O/bEIqE91TjFVQ==
-----END PUBLIC KEY-----
```

The peer public key for encryption is stored on disk, in /opt/open-xchange/etc/peer-pub.pem e.g.:

`peer-pub.pem`

```
-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEh8VpMpNkE15pYX3K0NPsDX2r5MQJ
Ejuj11ALv4WJswjf8t2A4xb7slnPPxG0kC03Le8wnuR05Mi3vRKDH3WqQA==
-----END PUBLIC KEY-----
```

The corresponding settings in `/opt/open-xchange/etc/trustedidentity.properties` should look like e.g.:

```
com.openexchange.plugins.trustedidentity.key=file:auto:/opt/open-xchange/etc/trust.pem#ox-trust-example-2021
com.openexchange.plugins.trustedidentity.issuer=Open-Xchange Cloud

com.openexchange.plugins.trustedidentity.peer.publicKeyFile=/opt/open-xchange/etc/peer-pub.pem
```

15.1.4 Caching

The key information that is in the private key file on disk is kept in memory once it has been read on startup (as referenced by `com.openexchange.plugins.trustedidentity.key`). The same applies to the public keys for encrypting. To apply changes, either reference a different file or, change the content of the same file, but in either case a `reloadconfiguration` is necessary.

16 Unsubscribe Mailing Lists

16.1 Unsubscribe Mailing Lists Framework

Bundle Identifier	`com.openexchange.plugins.unsubscribe,`
	`com.openexchange.plugins.unsubscribe.json,`
	`com.openexchange.plugins.unsubscribe.connector.vadesecure`

Package(s)	`open-xchange-plugins-unsubscribe,`
	`open-xchange-plugins-unsubscribe-vadesecure`

Required capabilities	none
Available since	1.5.2-rev7

Frequently, users subscribe (or are added) to a mailing list and do not wish to receive messages from that mailing list anymore. Many users often report the messages as “spam” in their mail client, training anti-spam services that a legitimately-received email is spam when it is not. The more responsible and effective course of action is instead for the user to unsubscribe from the list.

Currently, only Vade Safe-Unsubscribe exists as a supported provider. However, the Middleware service is constructed so that another adapter could be used instead.

16.1.1 Prerequisites

A MTA service must be configured to receive unsubscribe requests (via API or direct email) in order to process unsubscribe requests. The MTA must also insert an unsubscribe URL or mailto:email address in a header of the mail message.

16.1.2 Safe mode

If an unsubscribe service provider is properly configured, the plugin enters "safe" mode. This enables the `safe-unsubscribe` capability in the config-cascade and exposes a middleware endpoint at `/plugins/unsubscribe` which will relay unsubscribe requests to the provider API.

16.1.3 Unsafe mode

Without a provider configured, the plugin is considered running in "unsafe" mode. The user experience is the same, but instead of relaying unsubscribe requests through the middleware API, a mail is crafted and sent to any `mailto:` address in the List-Unsubscribe header. Non-mail unsubscribe URLs are ignored in this mode.

16.1.4 Configuration

`/opt/open-xchange/etc/plugins-unsubscribe.properties`

```
# Setting to control the used connector for a specific user
# This setting is config-cascade aware to support different implementations for each user.
# Default is <none> which means that the feature is disabled for a user
# To enable vade secure com.openexchange.plugins.unsubscribe.connector=
    plugins_unsubscribe_vadesecure
com.openexchange.plugins.unsubscribe.connector=

# Setting to enable safe_mode capability via config-cascade
# This setting is config-cascade aware to support different implementations for each user.
# Default is false which means that the feature is disabled for a user
com.openexchange.plugins.unsubscribe.safemode=false
```

16.1.5 Vade Connector

The `open-xchange-plugins-unsubscribe-vadesecure` package contains all vendor-specific code for the adapter and serves as a reference implementation for another adapter. Once the package is installed and configured, and app node is started, the plugin will be registered with the platform and discovered by the unsubscribe connector framework. The VadeSecure `UnsubsubscribeConnector` service needs to be registered with the connector identifier within framework and also to enable the `safe_unsubscribe` UI capability:

`/opt/open-xchange/etc/plugins-unsubscribe.properties`

```
com.openexchange.plugins.unsubscribe.connector=plugins_unsubscribe_vadesecure
com.openexchange.plugins.unsubscribe.safemode=true
```

The request payload should be in one of the following JSON Formats, where "`mail`" is optional and the `unsubscribeUrl` may contain either a single unsubscribe url/ mailto link, or a JSONArray containing multiple unsubscribe locations:

16.1.5.1 Example 1

```
[{"mail":["First Last","first.last@example.com"],"unsubscribeUrl":"http://www.example.com
```

```
    "}]
```

16.1.5.2 Example 2

```
[{"mail":["First Last","first.last@example.com"],"unsubscribeUrl":["http://www.example.com
    ", "mailto:yourlist@example.com?subject=remove"]}]
```

This connector performs following steps:

- The user's aliases are looked up based on the uid/cid contained in the ServerSession.
- If the optional "mail" key is included in the JSON data, the email address is compared against existing user aliases, if found the unsubscribe request is sent to the VadeSecure unsubscribe API. If the mail is not found, a 400 error is returned to the client.
- If the optional "mail" key is not included in the JSON data, all aliases are sent to the VadeSecure unsubscribe API. If the VadeSecure API responds with a failure for any alias, a 400 response is returned to the client.
- When multiple unsubscribe urls and/or mailto links are included in the middleware request, it is possible for one or more unsubscribe scenarios succeed and one or more to fail. Additionally, mailto links are never processed immediately, and always return a PENDING result initially. In the case where a single unsubscribe scenario succeeds with either SUCCESS or PENDING, the request is considered successful.

/opt/open-xchange/etc/plugins-unsubscribe-vadesecure.properties

```
# The customer license provided by VadeSecure; required to access unsubscribe API
# Default: The OX customer license
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.unsubscribe.vadesecure.license.passcrypt=<Customer license
    provided by VadeSecure>

# Setting to change the VadeSecure unsubscribe API URL
# Default: https://ws.vaderetro-unsubscribe.com/
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.unsubscribe.vadesecure.unsubscribe_url=https://ws.vaderetro-
    safeunsubscribe.com/

# Setting to change the VadeSecure connector identifier referenced in plugins-unsubscribe.
    properties / com.openexchange.plugins.unsubscribe.connector
# Default: "plugins_unsubscribe_vadesecure"
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.unsubscribe.vadesecure.identifier=plugins_unsubscribe_vadesecure
```

17 Shipped Version

17.1 Package open-xchange-authentication-masterpassword

Authentication implementation that uses a global password for all users – DO NOT USE IN PRODUCTION This package provides an authentication implementation that verifies user passwords against a globally configured password. DO NOT USE THIS IN PRODUCTION ! This implementation is only meant for testing and migration scenarios.
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

Conflicts with:

```
open-xchange-authentication-database
open-xchange-authentication-ldap
```

17.1.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-authentication-masterpassword
```

17.1.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/masterpassword-authentication.properties (page 54)

17.2 Package open-xchange-ldap-client

This package provides an advanced LDAP client library that is used by other Open-Xchange bundles.
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

17.2.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-ldap-client
```

17.2.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/ldap-client.d/ldap-client-pools.yaml.example (page 57)

17.3 Package open-xchange-ldap-client-test

REST API to test open-xchange-ldap-client (NOT FOR PRODUCTION) Exposes a REST API to test whether the open-xchange-ldap-client feature functions properly. This package is only meant for testing - DO NOT INSTALL IN PRODUCTION.
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.5)
open-xchange-ldap-client
```

17.3.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-ldap-client-test
```

17.4 Package open-xchange-metrics-http

Metrics for HTTP requests This package provides a highly configurable set of metrics around HTTP requests.
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

17.4.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-metrics-http
```

17.4.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/metrics-http.properties (page 60)

17.5 Package open-xchange-metrics-imap

Metrics for IMAP operations This package provides a set of metrics around IMAP operations.
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

17.5.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-metrics-imap
```

17.5.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/metrics-imap.properties (page 61)

17.6 Package open-xchange-minimal-api

This package provides the base Minimal API
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
open-xchange-rest (<<7.10.7)
open-xchange-rest (>=7.10.6)
```

17.6.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-minimal-api
```

17.6.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/minimal-api.properties (page 62)

17.7 Package open-xchange-minimal-api-calendar

This package provides the calendar endpoints for the Minimal API
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
open-xchange-minimal-api (<<1.7.3)
open-xchange-minimal-api (>=1.7.2)
open-xchange-minimal-api-security
```

17.7.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-minimal-api-calendar
```

17.8 Package open-xchange-minimal-api-jwt

This package provides the security handling for the Minimal API
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
open-xchange-minimal-api (<<1.7.3)
open-xchange-minimal-api (>=1.7.2)
open-xchange-sessionstorage-hazelcast (<<7.10.7)
open-xchange-sessionstorage-hazelcast (>=7.10.6)
```

17.8.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-minimal-api-jwt
```

17.9 Package open-xchange-minimal-api-mail

This package provides the mail endpoints for the Minimal API
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

```
open-xchange-minimal-api (<<1.7.3)
open-xchange-minimal-api (>=1.7.2)
open-xchange-minimal-api-security
```

17.9.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-minimal-api-mail
```

17.10 Package open-xchange-plugins-antiphishing

Plugins abstraction layer for AntiPhishing API connectors
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

17.10.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-plugins-antiphishing
```

17.10.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/plugins-antiphishing.properties (page 63)

17.11 Package open-xchange-plugins-antiphishing-vadesecure

This package installs the OSGi bundles needed to access the VadeSecure antiphishing plugin
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
open-xchange-plugins-antiphishing (<<1.7.3)
open-xchange-plugins-antiphishing (>=1.7.2)
```

17.11.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-plugins-antiphishing-vadesecure
```

17.11.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/plugins-antiphishing-vadesecure.properties (page 64)

17.12 Package open-xchange-plugins-blackwhitelist

Plugins abstraction layer for blacklist/whitelist connectors
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

17.12.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-plugins-blackwhitelist
```

17.12.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/plugins-blackwhitelist.properties (page 64)

17.13 Package open-xchange-plugins-blackwhitelist-sieve

This package installs the OSGi bundles needed to access the blacklist for plugins within Sieve
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
open-xchange-mailfilter (<<7.10.7)
open-xchange-mailfilter (>=7.10.6)
open-xchange-plugins-blackwhitelist (<<1.7.3)
open-xchange-plugins-blackwhitelist (>=1.7.2)
```

17.13.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-plugins-blackwhitelist-sieve
```

17.13.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/plugins-blacklist-sieve.properties (page 64)

17.14 Package open-xchange-plugins-contact-storage-group

Plugins contact storage that creates group folders
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-admin (<<7.10.7)
open-xchange-admin (>=7.10.6)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

17.14.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-plugins-contact-storage-group
```

17.14.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/plugins-contact-storage-group.properties (page 65)

17.15 Package open-xchange-plugins-contact-whitelist-sync

Plugins abstraction layer for whitelist contact connectors
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
open-xchange-ldap-client (<<1.7.3)
open-xchange-ldap-client (>=1.7.2)
open-xchange-sql-client (<<1.7.3)
open-xchange-sql-client (>=1.7.2)
```

17.15.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-plugins-contact-whitelist-sync
```

17.15.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/plugins-contacts-whitelist.properties (page 65)
/opt/open-xchange/etc/plugins-contacts-whitelist-migration.properties (page 65)
/opt/open-xchange/etc/plugins-contacts-whitelist-rdb.properties (page 66)
/opt/open-xchange/etc/sql-client.d/sql-plugins-whitelist.yaml.example (page 67)

17.16 Package open-xchange-plugins-mx-checker

Plugins abstraction layer for MX Checker connectors
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

17.16.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-plugins-mx-checker
```

17.16.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/plugins-mx-checker.properties (page 67)

17.17 Package open-xchange-plugins-onboarding-maillogin

Plugin that enables the overriding of the login information that is shown to users during onboarding.
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-client-onboarding (<<7.10.7)
open-xchange-client-onboarding (>=7.10.6)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

17.17.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-plugins-onboarding-maillogin
```

17.17.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/client-onboarding-maillogin.properties (page 67)

17.18 Package open-xchange-plugins-trustedidentity

Enables Trusted Identity API Support.
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

17.18.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-plugins-trustedidentity
```

17.18.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/trustedidentity.properties (page 70)

17.19 Package open-xchange-plugins-unsubscribe

Plugins abstraction layer for unsubscribe API connectors
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

Conflicts with:

```
open-xchange-plugins-safeunsubscribe
```

17.19.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-plugins-unsubscribe
```

17.19.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/plugins-unsubscribe.properties (page 70)

17.20 Package open-xchange-plugins-unsubscribe-vadesecure

This package installs the OSGi bundles needed to access the VadeSecure unsubscribe plugin
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
open-xchange-plugins-unsubscribe (<<1.7.3)
open-xchange-plugins-unsubscribe (>=1.7.2)
```

Conflicts with:

```
open-xchange-plugins-safeunsubscribe-vadesecure
```

17.20.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-plugins-unsubscribe-vadesecure
```

17.20.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/plugins-unsubscribe-vadesecure.properties (page 70)

17.21 Package open-xchange-sms-twilio

This package installs the OSGi bundles needed to send SMS messages via twilio
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

17.21.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-sms-twilio
```

17.21.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/twilio.properties (page 71)

17.22 Package open-xchange-sql-client

This package provides an advanced SQL client library that is used by other Open-Xchange bundles.
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

17.22.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-sql-client
```

17.22.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/sql-client.properties (page 71)
/opt/open-xchange/etc/sql-client.d/sql-client-pools.yaml.example (page 73)

17.23 Package open-xchange-util-imap

This package is a library that provides various utilities for IMAP.
Version: 1.7.2-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

17.23.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-util-imap
```

A Configuration Files

File 1 /opt/open-xchange/etc/masterpassword-authentication.properties

```
# Configuration file for the master password authentication plugin
#
# DO NOT USE IN PRODUCTION !
#

# The clear text password to authenticate all users.
# Mandatory.
# Example:
```

```
# com.openexchange.authentication.masterpassword.password=supersecret
com.openexchange.authentication.masterpassword.password=

# The default value for the context when it is not specified.
# Optional and defaults to using the "defaultcontext" mapping.
#com.openexchange.authentication.masterpassword.default.context=

# Whether the username portion of the login should be lowercased
# before being looked up in the user database.
# Optional and defaults to false
#com.openexchange.authentication.masterpassword.lowercase=false

# Whether the context name portion of the login should be lowercased
# before being looked up in the context database.
# Optional and defaults to false
#com.openexchange.authentication.masterpassword.lowercase.context=false

# Whether to use the complete login string as the username,
# e.g. login "foo@bar.com" -> user name "foo@bar.com" and
# context name "bar.com"
# Optional and defaults to false
#com.openexchange.authentication.masterpassword.use.full.login.info=false

# Whether to use the complete login string for the context name,
# e.g. login "foo@bar.com" -> context name "foo@bar.com"
# Optional and defaults to false
#com.openexchange.authentication.masterpassword.use.full.login.info.for.context=false
```

File 2 /opt/open-xchange/etc/ldap-client.d/ldap-client-pools.yaml.example

```
# The top-level key is the identifier of the pool, which can be
# any string of text and is being used by the bundles and applications
# to access that pool configuration.
# Typically, those are fixed or need to be configured in the bundles
# that use this library.
#
# When Java Security Manager support is enabled, files that are referenced
# in these configuration files must be in a directory that is already
# whitelisted, or in a subdirectory thereof, such as
# /opt/open-xchange/etc/
#
# A good candidate would be something along the lines of
# /opt/open-xchange/etc/ldap-files/
#
# Otherwise, the filename or its directory must be put into a new .list
# file in the folder
# /opt/open-xchange/etc/security/
# with e.g. the following content:
#
# file:/etc/trust.jks
#
pool1:
  trust-store:
    # SSL: path to the JKS trust store file that contains the anchors
    file: /etc/trust.jks
    # SSL: indicates whether to reject certificates if the current time
    # is outside the validity window for the certificate
    validity: true
  key-store:
    # SSL: path to the JKS client key store file that contains the key
    file: /etc/private.jks
    # SSL: password to access the keystore and the key
    password: foobar
    # SSL: alias of the key to use
    alias: key
  # Configure a read/write pool with different settings for read operations
  # and for write operations (i.e. different pools of LDAP servers).
  # Here comes the part for the read operations:
```

```
  read:
    # Use a failover cluster of two nodes:
    failover:
      - ldap1.example.com
      - ldap2.example.com
    # Pool connection management
    # --------------------------
    # When creating a connection pool, you may specify an initial number of
    # connections (pool-min) and a maximum number of connections (pool-max).
    # The initial number of connections is the number of connections that should
    # be immediately established and available for use when the pool is created.
    # The maximum number of connections is the largest number of unused connections
    # that may be available in the pool at any time.
    # Whenever a connection is needed, whether by an attempt to check out a
    # connection or to use one of the pool's methods to process an operation,
    # the pool will first check to see if there is a connection that has already
    # been established but is not currently in use, and if so then that connection
    # will be used.
    # If there aren't any unused connections that are already established, then
    # the pool will determine if it has yet created the maximum number of
    # connections, and if not then it will immediately create a new connection
    # and use it.
    # If the pool has already created the maximum number of connections, then the
    # pool may wait for a period of time (as configured using 'maxWaitTimeMillis' below,
    # which has a default value of zero to indicate that it should not wait at all)
    # for an in-use connection to be released back to the pool.
    # If no connection is available after the specified wait time (or there should
    # not be any wait time), then the pool may automatically create a new connection
    # to use if 'createIfNecessary' is true (which is the default).
    # If it is able to successfully create a connection, then it will be used.
    # If it cannot create a connection, or if 'createIfNecessary' is set to false,
    # then an error will be thrown.
    # Note that the maximum number of connections specified when creating a pool
    # refers to the maximum number of connections that should be available for use
    # at any given time.
    # If 'createIfNecessary' is set to true, then there may temporarily be more
    # active connections than the configured maximum number of connections.
    # This can be useful during periods of heavy activity, because the pool will
    # keep those connections established until the number of unused connections
    # exceeds the configured maximum.
    # If you wish to enforce a hard limit on the maximum number of connections so
    # that there cannot be more than the configured maximum in use at any time,
    # then set 'createIfNecessary' to false to indicate that the pool should not
    # automatically create connections when one is needed but none are available,
    # and you may also want to set 'maxWaitTimeMillis' to a maximum wait time to allow
    # the pool to wait for a connection to become available rather than throwing
    # an exception if no connections are immediately available.
    pool-min: 10
    pool-max: 50
    maxConnectionAgeMillis: 30000
    maxWaitTimeMillis: 500
    createIfNecessary: true
    # Specifies whether certain operations that should be retried on a newly-created
    # connection if the initial attempt fails in a manner that indicates that the
    # connection used to process the request may no longer be valid.
    # Only a single retry will be attempted for any operation.
    retryFailedOperations: true
  # Here comes the part for the write operations:
  write:
    host: ldap0.example.com
    pool-min: 1
    pool-max: 10
    maxConnectionAgeMillis: 60000
    maxWaitTimeMillis: 1000
    createIfNecessary: false
    retryFailedOperations: false
  # Specifies whether the pool should attempt to abandon any request for which
  # no response is received in the maximum response timeout period:
  abandonOnTimeout: true
  # Specifies the maximum length of time in milliseconds that a connection attempt
  # should be allowed to continue before giving up.
  # A value of zero (default) indicates that there should be no connect timeout.
```

```
    connectionTimeoutMillis: 3000
    # Specifies the maximum length of time in milliseconds that an operation should
    # be allowed to block while waiting for a response from the server.
    # A value of zero indicates that there should be no timeout.
    responseTimeoutMillis: 5000
    # Specifies whether to use the SO_KEEPALIVE option for the underlying sockets
    # used by associated connections.
    keepAlive: true
    # Specifies whether to use the TCP_NODELAY option for the underlying sockets.
    tcpNoDelay: true
    # Specifies whether to operate in synchronous mode, in which at most one
    # operation may be in progress at any time on a given connection.
    # When using asynchronous mode, a background thread takes care of multiplexing
    # and dispatching all the operations on connections that are shared for
    # multiple operations.
    synchronousMode: true
    # Specifies the length of time in milliseconds between periodic background
    # health checks against the available connections in this pool.
    healthCheckIntervalMillis: 120000
    # Specifies whether associated connections should attempt to follow any
    # referrals that they encounter.
    followReferrals: true
    # Specifies the maximum number of hops that a connection should take when
    # trying to follow a referral, must be greater than zero when 'followReferrals'
    # is true.
    referralHopLimit: 1
    # Specifies the maximum size in bytes for an LDAP message that a connection
    # will attempt to read from the directory server.
    # If it encounters an LDAP message that is larger than this size, then the
    # connection will be terminated.
    # Disabled when not specified or set to 0.
    maxMessageSize: 1024

  pool2:
    # A failover pool that uses the same set of servers for read and for
    # write operations.
    failover:
      - ldap0.example.com
      - ldap1.example.com
    pool-min: 5
    pool-max: 20
    trust-store:
      file: /etc/trust.jks
    key-store:
      file: /etc/private.jks

  pool3:
    # A simple single-host setup
    host: ldap.example.com
    pool-min: 5
    pool-max: 20

  pool4:
    # A load-balancing setup that will use a round-robin algorithm to
    # select the server to which the connection should be established.
    # Any number of servers may be included, and each request will
    # attempt to retrieve a connection to the next server in the list,
    # circling back to the beginning of the list as necessary.
    # If a server is unavailable when an attempt is made to establish
    # a connection to it, then the connection will be established to
    # the next available server in the set.
    round-robin:
      - host: ldap1.example.com
        port: 10389
        responseTimeoutMillis: 5000
      - host: ldap2.example.com
        port: 10389
        responseTimeoutMillis: 12000
    pool-min: 10
    pool-max: 50

  pool5:
```

```
  # A DNS RR setup handles the case in which a given hostname may
  # resolve to multiplee IP addresses.
  # Note that while a setup like this is typically referred to as
  # "round-robin DNS", this option does not strictly require DNS (as names
  # may be resolved through alternate mechanisms like a hosts file or an
  # alternate name service), and it does not strictly require round-robin
  # use of those addresses (as alternate ordering mechanisms like
  # 'random' or 'failover' may be used).
  dns-round-robin:
    host: ldap.example.com
    # The selection mode that should be used if the hostname resolves
    # to multiple addresses.
    # Possible values:
    # - random: the order of addresses will be randomized for each attempt
    # - failover: addresses will be consistently attempted in the order
    #       they are retrieved from the name service.
    # - round-robin: connection attempts will be made in a round-robin order
    selection-mode: random
    # Only use DNS if set to 'true'.
    # If set to 'false' then the operating system's hostname resolution
    # service will be used, which may include a hosts file.
    only-dns: false
    # The maximum length of time in milliseconds to cache addresses resolved
    # from the provided hostname.
    # Caching resolved addresses can result in better performance and can
    # reduce the number of requests to the name service.
    # A value that is less than or equal to zero indicates that no caching
    # should be used.
    cache-timeout: 1440000
  pool-min: 5
  pool-max: 20

pool6:
  # A failover pool that uses the same set of servers for read and for
  # write operations, as well as StartTLS
  failover:
    - ldap0.example.com
    - ldap1.example.com
  pool-min: 5
  pool-max: 20
  starttls: true
  trust-store:
    file: /etc/trust.jks
  key-store:
    file: /etc/private.jks
```

File 3 /opt/open-xchange/etc/metrics-http.properties

```
#
# The following property defines the various elements to use to compose the names of
# the metrics, to determine how to group them and what to see.
#
# The elements are separated by dots (".") and parsed individually, then replaced by
# their respective value for each inbound HTTP request to determine the name of
# the metric to update.
#
# Note that not all elements necessarily always result in a value as some are only
# present for specific types of HTTP requests, and others are optional (for example
# all the user information related ones that are only available when the HTTP request
# is authenticated or used in the context of an established Open-Xchange session).
# Values that are not available are skipped in the resulting name of the metric.
#
# For each component, here are the possible values to specify in this property:
# status
# ======
# Will be replaced by "success" or "error" depending on the result, for example:
# /api/rest/x/y/z -> success
#
```

```
# path
# ====
# If the HTTP is an AJAX API call, it will be replaced by "//module/action", and if not
# (e.g. accessing a servlet instead), it will be replaced with the servlet path.
#
# Examples:
# /ajax/folders?action=get&id=1,2,4 -> //folders/get
# /rest/api/x/y/z --------------------> /rest/api/x/y/z
#
# info
# ====
# Will be replaced with the servlet path info, i.e. the part of the URL that is behind
# the servlet path.
#
# Examples:
# /rest/api/users/john.doe@example.com -> john.doe@example.com
#
# session
# =======
# The value "session", "session_id" or "sessionid" will be replaced by the Open-Xchange
    session
# identifier, if applicable.
# For HTTP operations that are not authenticated, it will be left out.
#
# context_id
# ==========
# The value "context_id" or "cid" will be replaced by the numeric context identifier of
    the
# user, if applicable.
# For HTTP operations that are not authenticated, it will be left out.
#
# user_id
# =======
# The value "user_id" or "cid" will be replaced by the numeric user identifier of the
# user within the context, if applicable.
# For HTTP operations that are not authenticated, it will be left out.
#
# login
# =====
# The value "login" will be replaced by the login the user entered to authenticate or the
# user identifier provided by an SSO mechanism, if applicable.
# For HTTP operations that are not authenticated, it will be left out.
#
# property(module)
# ================
# Will be replaced by the AJAX API module, if applicable.
#
# property(action)
# ================
# Will be replaced by the AJAX API module action, if applicable.
#
# header(...)
# ===========
# Will be replaced by the value of an HTTP request header, the name of the header
# being specified between the parentheses.
# Note that header names are case sensitive.
#
# Example:
# header(Host).path -> appsuite01.example.com.//folders/list
#
# parameter(...)
# ==============
# Will be replaced by the value of an HTTP request parameter, the name of the
# parameter being specified between the parentheses.
#
# Example:
# header(Host).parameter(app).path -> appsuite01.example.com.io.ox/mail.//folders/list
#
# cookie(...)
# ===========
# Will be replaced by the value of a cookie present in the HTTP request, the name of the
# cookie being specified between the parentheses.
```

```
#
# session(...)
# ============
# Will be replaced by the value of a parameter present in the user's Open-Xchange session,
# the name of the session parameter being specified between the parentheses.
#
# text(...)
# =========
# Specifies text that will be used as-is.
#
com.openexchange.metrics.http.elements=path.status

# When aggregation is enabled (by setting this value to true), each element as configured
# by the property com.openexchange.metrics.http.elements will be a metric in its own right
    ,
# and aggregated accordingly to its path.
# Without aggregation, each metric is "flat".
#
# For example, with the following configuration
#   com.openexchange.metrics.http.elements=header(Host).path.status
#   com.openexchange.metrics.http.aggregation=true
# each element will be a metric, namely:
# 1. header(Host)
# 2. header(Host).path
# 3. header(Host).path.status
#
# Specifically, results will look along the lines of the following, each being a metric:
# - appsuite01.example.com
# - appsuite01.example.com.//folders/list
# - appsuite01.example.com.//folders/list.success
#
# Each of those metrics except for the last one will be aggregating the measurements
# of their parent metrics.
#
com.openexchange.metrics.http.aggregation=false

# List of logins for which to create specific metrics.
# In order to be able to track and aggregate the metrics of specific users, the
# following property can be set to a (full) login name as entered by the user when
# authenticating or as provided by an SSO system if applicable.
#
# For each of the logins specified through this property, an additional set
# of metrics will be created, prefixing the elements that are defined in
# com.openexchange.metrics.http.elements
# with the login value.
#
# For example, the following configuration
#   com.openexchange.metrics.http.elements=header(host).path.status
#   com.openexchange.metrics.http.aggregation=true
#   com.openexchange.metrics.http.logins=jdoe@example.com
# will result in the following list of metrics:
# 1. header(Host)
# 2. header(Host).path
# 3. header(Host).path.status
# 4. login
# 5. login.header(Host)
# 6. login.header(Host).path
# 7. login.header(Host).path.status
#
# Specifically, results will look along the lines of the following, each being a metric:
# - appsuite01.example.com
# - appsuite01.example.com.//folders/list
# - appsuite01.example.com.//folders/list.success
# - jdoe@example.com
# - jdoe@example.com.appsuite01.example.com
# - jdoe@example.com.appsuite01.example.com.//folders/list
# - jdoe@example.com.appsuite01.example.com.//folders/list.success
#
# Without aggregation, the following configuration
#   com.openexchange.metrics.http.elements=header(host).path.status
#   com.openexchange.metrics.http.aggregation=false
#   com.openexchange.metrics.http.logins=jdoe@example.com
# will result in this list of metrics instead:
# 1. header(Host).path.status
# 2. login.header(Host).path.status
#
# Note that if this property is commented out (not set) or left empty,
# no such additional per-login metrics will be created, which is the default
# behavior.
#
# Multiple logins may be specified, either by separating them with whitespaces
# and/or commas, e.g.:
# com.openexchange.metrics.http.logins=john.doe@example.com, jane.doe@example.com
# or by specifying multiple properties as follows:
# com.openexchange.metrics.http.logins.1=john.doe@example.com
# com.openexchange.metrics.http.logins.2=jane.doe@example.com
# (both may also be combined).
#
# Furthermore, it is possible to use regular expressions and wildcards:
# - if a login contains * or ?, it is understood to be a wildcard
# - if a login is enclosed in /.../ or /.../i (case insensitive), it is understood
#   to be a regular expression
# Examples:
# com.openexchange.metrics.http.logins=*@example.com, /^j(ohn|ane)\.doe@example\.cm$/
#
# Being a wildcard, the following value would match all logins:
# com.openexchange.metrics.http.logins=*
#
com.openexchange.metrics.http.logins=

# List of paths and path patterns for which to maintain metrics.
#
# The following property specifies discrete paths, path wildcard patterns, or
# regular expressions that will be matched against the HTTP request paths, and
# only those that match will have metrics.
#
# If the property value contains * or ?, it will be understood as a wildcard pattern.
# If it starts with / and ends with / or /i (cae insensitive), it will be understood
# as a regular expression.
# If it is neither of those, it will be interpreted as an exact (string comparison) value.
#
# To enable metric collection for all URLs, use the following value:
# com.openexchange.metrics.http.path=*
#
# If the value is not defined or empty, no metrics will be collected:
# com.openexchange.metrics.http.path=
#
# Example:
# com.openexchange.metrics.http.path.1=/^/appsuite/.+/(boot|precore)\.js$/
# com.openexchange.metrics.http.path.2=/appsuite/api/apps/manifests
# com.openexchange.metrics.http.path.3=/appsuite/api/mail
#
com.openexchange.metrics.http.path=

# The behavior of the path matching above can be configured with the following property.
# Possible values:
# - whitelist: any URL path that matches one of the URL patterns configured
#   using com.openexchange.metrics.http.path will be measured with metrics;
#   any URL path that does not, will not be measured with metrics
# - blacklist: any URL path that does not matches one of the URL patterns configured
#   using com.openexchange.metrics.http.path will be measured with metrics
#
# When omitted, left empty or invalid, the default mode is whitelist
#
# Example:
# com.openexchange.metrics.http.path.mode=blacklist
#
com.openexchange.metrics.http.path.mode=whitelist
```

File 4 /opt/open-xchange/etc/metrics-imap.properties

```
# Configure whether to enable metrics for IMAP operations.
# When this property is omitted (commented out) or set to false, or empty,
# IMAP metrics will not be collected.
com.openexchange.metrics.imap.enable=false

# The number of threads to use to process IMAP operation results,
# updating metrics.
com.openexchange.metrics.imap.threads=2
```

File 5 /opt/open-xchange/etc/minimal-api.properties

```
# The capability to control whether or not the user is allowed to access the API
# at all
#
# Optional, default value: false
#
# Example:
# com.openexchange.capability.minimalapi=true
com.openexchange.capability.minimalapi=false

# The clients names enabled for a user
# Must be provided as a comma separated list
#
# Optional, default value: ""
#
# Must be provided as a comma separated list
#
# Example:
# com.openexchange.plugins.minimal.api.clients=exampleClient,exampleClient2
com.openexchange.plugins.minimal.api.clients=

# The user-friendly name of a client
#
# Optional, default value: ""
#
# If not set, the client identifier is returned.
#
# Example:
# com.openexchange.plugins.minimal.api.exampleClient.name=Example Preview
com.openexchange.plugins.minimal.api.[client].name=

# The claims assigned to a client
#
# Optional, default value: ""
#
# Must be provided as a comma separated list
#
# Example:
# com.openexchange.plugins.minimal.api.exampleClient.claims=readMail
com.openexchange.plugins.minimal.api.[client].claims=

# Default consent if user has not yet decided on first access
# WARNING: It might be required by law to enforce user consent
#
# Optional, default value: false
#
# Example:
# com.openexchange.plugins.minimal.api.exampleClient.defaultconsent=true
com.openexchange.plugins.minimal.api.[client].defaultconsent=false

# Maximum amount of requests per second per source IP address if the token could not be
    validated from cache
# May be a decimal number.
#
# Optional, default value: 1.0
# Optional, default for client: 5.0
#
```

```
# Example:
# com.openexchange.plugins.minimal.api.ratelimit.requestsPerSecond=10.0
# com.openexchange.plugins.minimal.api.ratelimit.exampleClient.maxRequestsPerSecond=10.0
com.openexchange.plugins.minimal.api.ratelimit.requestsPerSecond=1.0

# Maximal time window, in milliseconds: after a given source IP address has not accessed
# the minimal API, its number of requests per second rate is reset.
#
# Optional, default value: 300000
# Optional, default for client: 300000
#
# Example:
# com.openexchange.plugins.minimal.api.ratelimit.maxRateTimeWindow=60000
# com.openexchange.plugins.minimal.api.ratelimit.exampleClient.maxRateTimeWindow=60000
com.openexchange.plugins.minimal.api.ratelimit.maxRateTimeWindow=300000

# Strategy to use for reacting to the inability to access the API for a given source
# IP address due to surpassing the maxRequestsPerSecond rate.
#
# Format: it must be one of:
# * fail-fast
# * block
# * timeout:...
#
# fail-fast
#   if the rate limit is exceeded, the API will respond with a 401 Unauthorized
# block
#   if the rate limit is exceeded, the API will block infinitely until the rate limit
#   allows for another request to be performed
# timeout:...
#   block until the specified timeout is reached, after which the API responds with a
#   401 Unauthorized
#   if the timeout does not allow to get a new token in time, a 401 Unauthorized is
#   returned
#   The value after "timeout:" consists of a number followed by a time unit, examples:
#   - timeout:400s ---> 400 seconds
#   - timeout:1m ------>   1 minute
#   - timeout:2000ms -> 2000 milliseconds
#
# If the token could be validated and is correct, the API will not return a
# 401 Unauthorized but a 429 Too Many Requests instead.
#
# Optional, default value: timeout:250ms
# Optional, default for client: timeout:500ms
#
# Example:
# com.openexchange.plugins.minimal.api.ratelimit.strategy=timeout:1s
# com.openexchange.plugins.minimal.api.ratelimit.exampleClient.strategy=timeout:5s
com.openexchange.plugins.minimal.api.ratelimit.strategy=timeout:250ms
```

File 6 /opt/open-xchange/etc/plugins-antiphishing.properties

```
# Setting to control the used connector for a specific user
# This setting is config-cascade aware to support different implementations for each user.
# Default is <none> which means that the feature is disabled for a user
# To enable vade secure com.openexchange.plugins.antiphishing.connector=
    plugins_antiphishing_vadesecure
com.openexchange.plugins.antiphishing.connector=

# Setting to enable/disable the antiphishing capability
# This setting is config-cascade aware to support different implementations for each user.
# Default is false which means that the feature is disabled for a user
com.openexchange.plugins.antiphishing.enabled=false

# Setting to enable/disable the antiphishing mta_capability
# If true, the user has the ability to choose antiphishing at the MTA level
# This setting is config-cascade aware to support different implementations for each user.
# Default is false which means that the feature is disabled for a user
```

```
com.openexchange.plugins.antiphishing.mta_capability=false

# Setting to enable/disable the antiphishing at the mta level
# If true, an antiphishing check will take place at the MTA level
# This setting is config-cascade aware to support different implementations for each user.
# Additionally, this property can be set by the user in the UI
# Default is false which means that the feature is disabled for a user
com.openexchange.plugins.antiphishing.mta_antiphishing=false
```

File 7 /opt/open-xchange/etc/plugins-antiphishing-vadesecure.properties

```
# The customer name as provided by VadeSecure; required to access Phishing API
# Default: NONE
# Config-cascade aware: true
# Lean: false
com.openexchange.plugins.antiphishing.vadesecure.name.passcrypt=<Customer name provided by
     VadeSecure>

# The customer license provided by VadeSecure; required to access Phishing API
# Default: NONE
# Config-cascade aware: true
# Lean: false
com.openexchange.plugins.antiphishing.vadesecure.license.passcrypt=<Customer license
    provided by VadeSecure>

# Setting to change the VadeSecure IsItPhishing API URL
# Default: https://iip.eu.vadesecure.com/api/v2/url
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.antiphishing.vadesecure.phishing_url=https://iip.eu.vadesecure.
    com/api/v2/url

# Setting to change the VadeSecure connector identifier referenced in plugins-antiphishing
    .properties / com.openexchange.plugins.antiphishing.connector
# Default: "plugins_antiphishing_vadesecure"
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.antiphishing.vadesecure.identifier=
    plugins_antiphishing_vadesecure

# If set to true, the URL will always be crawled and analyzed, even if it can trigger
    collateral damages (such as unsubscribing a user, canceling an order, etc.).
# If set to false, the service checks whether the URL may cause collateral damage to the
    end user (unsubscribe, order confirmation, etc.). If so, the URL is not crawled and
    NOT_EXPLORED is returned in the response.
# Default: false
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.antiphishing.vadesecure.force=false

# Vade Secure IsItPhishing Smart mode enables URL anonymization. Typically, this is meant
    to
# replace any unique-ID like tokens in a URL by random characters, to prevent side effects
     when crawling certain URLs, which if visited, could trigger unwanted actions:
    unsubscription, cancelation, etc.
# Set to true to enable the smart mode. If set to false, URLs will be crawled in the way
    they were originally provided. If argument randomization fails, the URL is not crawled
     and NOT_EXPLORED is returned.# Default: "plugins_antiphishing_vadesecure"
# NOTE: Vade Secure strongly recommends enabling the smart parameter to true, so that the
    API can trigger token anonymization, to try and prevent any collateral damages.
# Default: false
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.antiphishing.vadesecure.smart=true

# Timeout in milliseconds, with a minimum value of 1000. Once timeout is reached, TIMEOUT
    response is returned.
# Default: 3000
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.antiphishing.vadesecure.timeout=3000

# The Vade GRAPH API to retrieve authorization tokens
# Default: https://api.vadesecure.com/oauth2/v2/token
# Config-cascade aware: false
com.openexchange.plugins.antiphishing.vadesecure.graph_url=https://api.vadesecure.com/oauth2/v2/token
```

File 8 /opt/open-xchange/etc/plugins-blackwhitelist.properties

```
# Setting to control the used connector for a specific user
# This setting is config-cascade aware to support different implementations for each user.
# Default is <none> which means that the feature is disabled for a user
com.openexchange.plugins.blackwhitelist.connector=

# Setting to check if memory backed test System should be started
# This connector is identified by plugins_blwl_test
# Default: false
com.openexchange.plugins.blackwhitelist.test=false
```

File 9 /opt/open-xchange/etc/plugins-blacklist-sieve.properties

```
# Identifier of this blackwhitelist connector: plugins_blackwhitelist_sieve
# Setting to control the rulename to be set and checked as a antispam value inside the sieve rules
# Default: Blacklist
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.blackwhitelist.connector.sieve.rulename=Blacklist

# Setting to control wether the blacklisted mails should be moved to SPAM or deleted directly
# If set to true, mails are moved to SPAM
# If set to false, mails are deleted
# Default: true
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.blackwhitelist.connector.sieve.moveToSpam=true

# Setting to check if memory backed test System should be started
# This connector is identified by plugins_blwl_test
# Default: false
com.openexchange.plugins.blackwhitelist.connector.sieve.test=false
```

File 10 /opt/open-xchange/etc/plugins-contact-storage-group.properties

```
# Configures whether the group contact storage is enabled for a context or not.
# Default: false
com.openexchange.plugins.contact.storage.group.enabled=false

# Defines an optional list of those groups for which no group contact folder should
# be used, as a comma-separated string of the identifiers of those groups that should
# be excluded. The groups "All Users", "All Guests" and the "Standard Group" are
# always excluded.
# Default: <empty>
com.openexchange.plugins.contact.storage.group.excludedGroups=

# Defines if the display name of the groups should be used to create the folder
# names in the folder tree.
# If set to <true>, the displayname is used
# If set to <false>, the group name is used
# The Group Names are limited by the property CHECK_GROUP_UID_REGEXP
com.openexchange.plugins.contact.storage.group.useDisplayName=true
```

File 11 /opt/open-xchange/etc/plugins-contacts-whitelist.properties

```
# This setting enables or disables special handling for the ContactCollectionFolder
# If set to true, the contactCollectFolder is ignored and contacts in this folder
# are not added to the whitelist. Contacts moved to this folder are also removed from the
    whitelist
# If set to false, the contactCollectFolder is handled like any other folder.
# config-cascade aware
# Default: true
com.openexchange.plugins.contacts.whitelist.ignoreContactCollectFolder=true

# This setting is used to set the connector for the contact sync.
# Currently available options are:
#  <not-set> (this will disable the sync for the user)
#  rdb
# Default: <not-set>
com.openexchange.plugins.contacts.whitelist.connector=
```

File 12 /opt/open-xchange/etc/plugins-contacts-whitelist-migration.properties

```
# Defines the strategy of the automatic migration
# Can be either
#    <not-set> which disables the automatic migration
#    once
#    time:<timeinmillis>
# Default: <not-set>
#
# Examples
# If sync should happen once a day:
# com.openexchange.plugins.contacts.whitelist.migration.strategy=time:86400000
# If sync should happen once a week
# com.openexchange.plugins.contacts.whitelist.migration.strategy=time:604800000
com.openexchange.plugins.contacts.whitelist.migration.strategy=

# Setting, if a warning should appear in the logs, if a user has more than configured
    contacts in one folder.
# Default: 10000
com.openexchange.plugins.contacts.whitelist.migration.warningSize=10000
```

File 13 /opt/open-xchange/etc/plugins-contacts-whitelist-rdb.properties

```
# Pool to be used
com.openexchange.plugins.contacts.whitelist.rdb.pool=contact-whitelist-pool

# normal or tombstone
com.openexchange.plugins.contacts.whitelist.rdb.strategy=normal

# table name
com.openexchange.plugins.contacts.whitelist.rdb.tableName=senderwl

# Name of the column used for the primary mail
com.openexchange.plugins.contacts.whitelist.rdb.primaryAddressColumnName=rcpt
```

```

# Name of the column used for the contact mails
com.openexchange.plugins.contacts.whitelist.rdb.contactMailColumnName=sender

# Name of the column used for the individual contactIds
com.openexchange.plugins.contacts.whitelist.rdb.contactIdColumnName=contactid

# Name of the deleted_at column if tombstone is enabled
com.openexchange.plugins.contacts.whitelist.rdb.tombstone.deletedAtColumnName=deleted_at

# Name of the updated_at column if tombstone is enabled
com.openexchange.plugins.contacts.whitelist.rdb.tombstone.updatedAtColumnName=updated_at
```

File 14 /opt/open-xchange/etc/sql-client.d/sql-plugins-whitelist.yaml.example

```
# The top-level key is the identifier of the pool, which can be
# any string of text and is being used by the bundles and applications
# to access that pool configuration.
# Typically, those are fixed or need to be configured in the bundles
# that use this library.
#
# When Java Security Manager support is enabled, files that are referenced
# in these configuration files must be in a directory that is already
# whitelisted, or in a subdirectory thereof, such as
# /opt/open-xchange/etc/
#
# A good candidate would be something along the lines of
# /opt/open-xchange/etc/sql-files/
#
# Otherwise, the filename or its directory must be put into a new .list
# file in the folder
# /opt/open-xchange/etc/security/
# with e.g. the following content:
#
# file:/etc/trust.jks
#
contact-whitelist-pool:
  # This is the name of the DataSource class provided by the JDBC driver.
  # Consult the documentation for your specific JDBC driver to get this class name, or see
       the table below.
  # Note XA data sources are not supported. XA requires a real transaction manager like
      bitronix.
  # Note that you do not need this property if you are using jdbcUrl for "old-school"
      DriverManager-based JDBC driver configuration.
  # Default: none
  dataSourceClassName: com.mysql.jdbc.jdbc2.optional.MysqlDataSource
  # This property directs HikariCP to use "DriverManager-based" configuration.
  # We feel that DataSource-based configuration (above) is superior for a variety of
      reasons (see below), but for many deployments there is little significant difference
      .
  # When using this property with "old" drivers, you may also need to set the
      driverClassName property, but try it first without.
  # Note that if this property is used, you may still use DataSource properties to
      configure your driver and is in fact recommended over driver parameters specified in
       the URL itself.
  # Default: none
  jdbcUrl: jdbc:mysql://mysql.example.com
  # This property sets the default authentication username used when obtaining Connections
       from the underlying driver.
  # Note that for DataSources this works in a very deterministic fashion by calling
      DataSource.getConnection(*username*, password) on the underlying DataSource.
  # However, for Driver-based configurations, every driver is different.
  # In the case of Driver-based, HikariCP will use this username property to set a user
      property in the Properties passed to the driver's DriverManager.getConnection(
      jdbcUrl, props) call.
  # If this is not what you need, skip this method entirely and call addDataSourceProperty
      ("username", ...), for example.
  # Default: none
  username: user
  # sets the password of the connection
  password: secret
```

File 15 /opt/open-xchange/etc/plugins-mx-checker.properties

```
# Determines which connector will be used for a user
# This setting is config-cascade aware to support different implementations for each user.
# Default is <none> which means that the feature is disabled for a user
com.openexchange.plugins.mx.checker.connector=
```

File 16 /opt/open-xchange/etc/client-onboarding-maillogin.properties

```
# Default value for overriding the login information displayed
# in the client onboarding.
#
# Possible values:
# email
#   uses the user's defaultSenderAddress
# attr:<name>
#   uses the user's attribute <name>
# login
#   uses the user's login, which is the same as if the
#   onboarding login was not overriden by this plugin
# login_name
#   uses the loginName attribute when possible, which is only the case
#   for session based logins (IMAP, SMTP) and for protocols that do not
#   create a session (CalDAV, CardDAV, EAS), it falls back on the login
#   instead
#
# This property is config cascade aware and must be set globally
# (in this file), and can then be overriden by context and/or by
# user.
#
# Note that for this feature to be enabled, one is also required
# to set one or more the following properties, depending on the
# client onboaridng dialogs that need the login information to
# be overriden by this plugin:
# com.openexchange.client.onboarding.caldav.login.customsource=true
# com.openexchange.client.onboarding.carddav.login.customsource=true
# com.openexchange.client.onboarding.mail.imap.login.customsource=true
# com.openexchange.client.onboarding.mail.smtp.login.customsource=true
#
com.openexchange.plugins.onboarding.login=login
```

File 17 /opt/open-xchange/etc/trustedidentity.properties

```
# URI to the private and public key resource to use to sign JWTs.
#
# The format of the URi epends on the scheme and driver.
# The "file" scheme is always supported.
#
# Format: file:<algorithm>:<path>[#<keyid>]
#
# Algorithm may either be "auto" in which case the signing algorithm will be inferred
# from the EC curve OID within the encoded private key part in the file, or be explicitly
# one of the supported values:
# - ES256: ECDSA using P-256 curve and SHA-256 hash algorithm
# - ES384: ECDSA using P-384 curve and SHA-384 hash algorithm
# - ES512: ECDSA using P-521 curve and SHA-512 hash algorithm
#
# Note that for the time being, only ECDSA keys are supported.
#
# The key id may be set as the fragment part of the URI: if set, will be stored as a kid (key id)
# claim in the JWT header, which identifies the key in some form that is understandable for consumers
# of the JWT token.
# Optional, does not set the kid claim when absent.
#
# The path is a fully qualified filesystem path to the private key PEM file to use for signing.
# It should also contain the certificate (public key part) in order to include the
# x5t#S256 (X.509 certificate SHA-256 thumbprint) in the signed token.
#
# Content of the file:
# -----BEGIN EC PRIVATE KEY-----
# MIGHAgEAMBMGByqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQgyGdEuJcaHla0CDtX
# ...
# Jvb9wIBomkOsFr++dEnvM97Sm3G+c8wkqL0+WFBRwTw79sQioT3V0MVV
# -----END EC PRIVATE KEY-----
# -----BEGIN EC PUBLIC KEY-----
# MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEqxHR/v8D3NktT/EfE5Mq2dvlIZ6H
# QCb2/cCAaJpDrBa/vnRJ7zPe0ptxvnPMJKi9PlhQUcE80/bEIqE91TjFVQ==
# -----END EC PUBLIC KEY-----
#
# The type (specified after BEGIN and END) in the PEM headers must be one of:
# - for the mandatory private key: PRIVATE KEY, EC PRIVATE KEY
# - for the optional public key: CERTIFICATE, PUBLIC KEY, EC PUBLIC KEY
#
# Examples:
# com.openexchange.plugins.trustedidentity.key=file:auto:/opt/open-xchange/etc/trustedidentity.pem#ox-trust-key-2021-1
# com.openexchange.plugins.trustedidentity.key=file:ES256:/opt/open-xchange/etc/trustedidentity.pem
#
# Mandatory, there is no default value.
com.openexchange.plugins.trustedidentity.key=

# The issuer (iss) string to include in the signed JWT.
# Describes this App Suite instance in its role as an authority.
#
# Mandatory, has no default value.
#
# Example:
# com.openexchange.plugins.trustedidentity.issuer=Open-Xchange
com.openexchange.plugins.trustedidentity.issuer=

# Expiration duration: the signed JWT contains a standard claim field
# "exp" that defines when the validity of the JWT should expire.
# The following configuration property configures how long that expiration
# time frame should be, always in addition to the current timestamp as
# of the system clock.
# e.g. "5m" will produce an expiration timestamp that is 5m in the future
#
# Format: <duration>[h|m|s|ms]
#
# Example:
# com.openexchange.plugins.trustedidentity.expiration=30m
#
# Optional, the default value is "5m" (5 minutes)
#
com.openexchange.plugins.trustedidentity.expiration=5m

# Public key file (PEM) location on disk.
#
# This is the public key to use for encrypting JWTs. That public key must be
# provided to us by the peer or customer that will receive the encrypted
# JWT, as they will be able to decrypt it using their private key part.
#
# Note tha this property is config-cascade aware.
```

```
#
# Example:
# com.openexchange.plugins.trustedidentity.peer.publicKeyFile=/opt/open-xchange/keys/customer1-pubkey1.pem
#
# This configuration setting is mandatory and has no default value.
# When left empty, it disables encryption.
com.openexchange.plugins.trustedidentity.peer.publicKeyFile=

# Algorithm to use to encrypt the JWT.
#
# The supported algorithms depend on the type of the public key.
#
# For an EC key:
#
# - ECDH-ES: Elliptic Curve Diffie-Hellman Ephemeral Static (RFC 6090) key agreement using the
#            Concat KDF, as defined in section 5.8.1 of NIST.800-56A, with the agreed-upon key
#            being used directly as the Content Encryption Key (CEK) (rather than being used to
#            wrap the CEK).
#
# - ECDH-ES+A128KW: Elliptic Curve Diffie-Hellman Ephemeral Static key agreement per "ECDH-ES",
#            but where the agreed-upon key is used to wrap the Content Encryption Key (CEK) with
#            the "A128KW" function (rather than being used directly as the CEK).
#
# - ECDH-ES+A192KW: Elliptic Curve Diffie-Hellman Ephemeral Static key agreement per "ECDH-ES",
#            but where the agreed-upon key is used to wrap the Content Encryption Key (CEK) with
#            the "A192KW" function (rather than being used directly as the CEK).
#
# - ECDH-ES+A256KW: Elliptic Curve Diffie-Hellman Ephemeral Static key agreement per "ECDH-ES",
#            but where the agreed-upon key is used to wrap the Content Encryption Key (CEK) with
#            the "A256KW" function (rather than being used directly as the CEK).
#
# For an RSA key:
#
# - RSA-OAEP-256: RSAES using Optimal Asymmetric Encryption Padding (OAEP) (RFC 3447), with the
#            SHA-256 hash function and the MGF1 with SHA-256 mask generation function.
#
# Note tha this property is config-cascade aware.
#
# Example:
# com.openexchange.plugins.trustedidentity.peer.algorithm=ECDH-ES+A256KW
#
# The property is optional and defaults to either ECDH-ES for EC keys, or
# to RSA-OAEP-256 for RSA keys.
com.openexchange.plugins.trustedidentity.peer.algorithm=

# Encryption Method to use to encrypt the JWT.
#
# The supported methods are as follows:
#
# - A128GCM: AES in Galois/Counter Mode (GCM) (NIST.800-38D) using a 128 bit key
# - A192GCM: AES in Galois/Counter Mode (GCM) (NIST.800-38D) using a 192 bit key
# - A256GCM: AES in Galois/Counter Mode (GCM) (NIST.800-38D) using a 256 bit key
#
# Note tha this property is config-cascade aware.
#
# Example:
# com.openexchange.plugins.trustedidentity.peer.encryptionMethod=A256GCM
#
# The property is optional and defaults to A256GCM
com.openexchange.plugins.trustedidentity.peer.encryptionMethod=

# Peer public key time-to-live in cache.
#
# Public keys are loaded from PEM files on-demand and are then cached for a configurable
# amount of time before being loaded again.
#
# Format: <duration>[w|d|h|m|s|ms]
#
# Example:
# com.openexchange.plugins.trustedidentity.peer.publicKeyCacheTtl=5d
#
# The property is optional and defaults to 1d (1 day)
com.openexchange.plugins.trustedidentity.peer.publicKeyCacheTtl=
```

File 18 /opt/open-xchange/etc/plugins-unsubscribe.properties

```
# Setting to control the used connector for a specific user
# This setting is config-cascade aware to support different implementations for each user.
# Default is <none> which means that the feature is disabled for a user
# To enable vade secure com.openexchange.plugins.unsubscribe.connector=
    plugins_unsubscribe_vadesecure
com.openexchange.plugins.unsubscribe.connector=

# Setting to enable safe_mode capability via config-cascade
# This setting is config-cascade aware to support different implementations for each user.
# Default is false which means that the feature is disabled for a user
com.openexchange.plugins.unsubscribe.safemode=false
```

File 19 /opt/open-xchange/etc/plugins-unsubscribe-vadesecure.properties

```
# The customer license provided by VadeSecure; required to access unsubscribe API
# Default: The OX customer license
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.unsubscribe.vadesecure.license.passcrypt=<Customer license
    provided by VadeSecure>

# Setting to change the VadeSecure unsubscribe API URL
# Default: https://ws.vaderetro-unsubscribe.com/
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.unsubscribe.vadesecure.unsubscribe_url=https://ws.vaderetro-
    safeunsubscribe.com/

# Setting to change the VadeSecure connector identifier referenced in plugins-unsubscribe.
    properties / com.openexchange.plugins.unsubscribe.connector
# Default: "plugins_unsubscribe_vadesecure"
# Config-cascade aware: true
# Lean: true
com.openexchange.plugins.unsubscribe.vadesecure.identifier=plugins_unsubscribe_vadesecure
```

File 20 /opt/open-xchange/etc/twilio.properties

```
# Twilio accountSID
com.openexchange.plugins.sms.twilio.accountSID.secret=ACCOUNT_SID

# Twilio auth token
com.openexchange.plugins.sms.twilio.authtoken.secret=AUTH_TOKEN

# Twilio Message Service SID
```

```
com.openexchange.plugins.sms.twilio.messageservicesid.secret=SERVICE_SID

# Max message length. 1600 characters is Twilio's maximum
com.openexchange.plugins.sms.twilio.maxlength=1600
```

File 21 /opt/open-xchange/etc/sql-client.properties

```
# Comma seperated list of drivers to read into the system
# As the sql-client is very early, it may happen that the excpected driver is not yet registered.
# To work around this issue, the following list of drivers will be read before any connection is
# created.
#
# Default: com.mysql.jdbc.Driver
com.openexchange.sql.client.drivers=com.mysql.jdbc.Driver
```

File 22 /opt/open-xchange/etc/sql-client.d/sql-client-pools.yaml.example

```
# The top-level key is the identifier of the pool, which can be
# any string of text and is being used by the bundles and applications
# to access that pool configuration.
# Typically, those are fixed or need to be configured in the bundles
# that use this library.
#
# When Java Security Manager support is enabled, files that are referenced
# in these configuration files must be in a directory that is already
# whitelisted, or in a subdirectory thereof, such as
# /opt/open-xchange/etc/
#
# A good candidate would be something along the lines of
# /opt/open-xchange/etc/sql-files/
#
# Otherwise, the filename or its directory must be put into a new .list
# file in the folder
# /opt/open-xchange/etc/security/
# with e.g. the following content:
#
# file:/etc/trust.jks
#
# For a complete list of property values, read https://github.com/brettwooldridge/HikariCP
pool1:
  # This is the name of the DataSource class provided by the JDBC driver.
  # Consult the documentation for your specific JDBC driver to get this class name, or see the table below.
  # Note XA data sources are not supported. XA requires a real transaction manager like bitronix.
  # Note that you do not need this property if you are using jdbcUrl for "old-school" DriverManager-based JDBC driver configuration.
  # Default: none
  dataSourceClassName: com.mysql.jdbc.jdbc2.optional.MysqlDataSource
  # This property directs HikariCP to use "DriverManager-based" configuration.
  # We feel that DataSource-based configuration (above) is superior for a variety of reasons (see below), but for many deployments there is little significant difference.
  # When using this property with "old" drivers, you may also need to set the driverClassName property, but try it first without.
  # Note that if this property is used, you may still use DataSource properties to configure your driver and is in fact recommended over driver parameters specified in the URL itself.
  # Default: none
  jdbcUrl: jdbc:mysql://mysql.example.com
  # This property sets the default authentication username used when obtaining Connections
```

```
         from the underlying driver.
    # Note that for DataSources this works in a very deterministic fashion by calling
        DataSource.getConnection(*username*, password) on the underlying DataSource.
    # However, for Driver-based configurations, every driver is different.
    # In the case of Driver-based, HikariCP will use this username property to set a user
        property in the Properties passed to the driver's DriverManager.getConnection(
        jdbcUrl, props) call.
    # If this is not what you need, skip this method entirely and call addDataSourceProperty
        ("username", ...), for example.
    # Default: none
    username: user
    # sets the password of the connection
    password: secret

  pool2:
    jdbcUrl: jdbc:mysql://mysql.example.com
    # This property controls the maximum number of milliseconds that a client (that's you)
        will wait for a connection from the pool.
    # If this time is exceeded without a connection becoming available, a SQLException will
        be thrown.
    # Lowest acceptable connection timeout is 250 ms.
    # Default: 30000 (30 seconds)
    connectionTimeout: 30000
    # This property controls the maximum amount of time that a connection is allowed to sit
        idle in the pool.
    # This setting only applies when minimumIdle is defined to be less than maximumPoolSize.
         Idle connections will not be retired once the pool reaches minimumIdle connections.
    # Whether a connection is retired as idle or not is subject to a maximum variation of
        +30 seconds, and average variation of +15 seconds.
    # A connection will never be retired as idle before this timeout.
    # A value of 0 means that idle connections are never removed from the pool.
    # The minimum allowed value is 10000ms (10 seconds).
    # Default: 600000 (10 minutes)
    idleTimeout: 600000
    # This property controls the maximum lifetime of a connection in the pool. An in-use
        connection will never be retired, only when it is closed will it then be removed.
    # On a connection-by-connection basis, minor negative attenuation is applied to avoid
        mass-extinction in the pool.
    # We strongly recommend setting this value, and it should be several seconds shorter
        than any database or infrastructure imposed connection time limit.
    # A value of 0 indicates no maximum lifetime (infinite lifetime), subject of course to
        the idleTimeout setting.
    # Default: 1800000 (30 minutes)
    maxLifetime: 1800000
    # This property controls the minimum number of idle connections that HikariCP tries to
        maintain in the pool.
    # If the idle connections dip below this value and total connections in the pool are
        less than maximumPoolSize, HikariCP will make a best effort to add additional
        connections quickly and efficiently.
    # However, for maximum performance and responsiveness to spike demands, we recommend not
         setting this value and instead allowing HikariCP to act as a fixed size connection
        pool.
    # Default: same as maximumPoolSize
    minimumIdle: 0
    # This property controls the maximum size that the pool is allowed to reach, including
        both idle and in-use connections.
    # Basically this value will determine the maximum number of actual connections to the
        database backend. A reasonable value for this is best determined by your execution
        environment.
    # When the pool reaches this size, and no idle connections are available, calls to
        getConnection() will block for up to connectionTimeout milliseconds before timing
        out.
    # Default: 10
    maximumPoolSize: 10

  # The following example shows how to provide additional dataSource properties to the pool
      by using the dataSourceProperties key.
  # The DataSource will be started with all key-value pairs added.
  pool3:
    jdbcUrl: jdbc:mysql://mysql.example.com
    username: user
    password: secret
    dataSourceProperties:
      useUnicode: true
      characterEncoding: UTF-8
      autoReconnect: false
      useServerPrepStmts: false
      useTimezone: true
      serverTimezone: UTC
      connectTimeout: 15000
      socketTimeout: 15000
      useSSL: false
      requireSSL: false
      verifyServerCertificate: false
      enabledTLSProtocols: TLSv1,TLSv1.1,TLSv1.2
```