



**OX App Suite Engineering Services Plugins Technical
Documentation for
1.7.2-rev5**

2022-12-22

Copyright notice

©2022 by OX Software GmbH. All rights reserved. Open-Xchange and the Open-Xchange logo are trademarks or registered trademarks of OX Software GmbH. All other company and/or product names may be trademarks or registered trademarks of their owners. Information contained in this document is subject to change without notice.

Contents

1	General Information	2
1.1	Warnings	2
1.2	Delivery Comment	2
1.3	Install Package Repository	2
1.4	Build Dependencies	2
1.5	Notice	2
1.6	Release Announcements	3
2	Anti Phishing Framework	3
2.1	Framework Configuration	3
2.2	Anti Phishing API	4
2.2.1	Example Config Request	4
2.2.2	Example Get Request	4
2.2.3	Example Update Request	4
2.2.4	Example Linkcheck Request	4
2.3	The Anti Phishing Connectors	5
2.3.1	The VadeSecure Connector	5
2.3.1.1	Custom OXaaS LDAP Extension	6
3	Block Allow List Middleware	6
3.1	Connectors	6
3.1.1	Sieve Connector	6
3.1.2	LDAP Connector	7
3.2	Commandline Tools	7
3.3	Configuration	7
4	Contact Whitelist Sync	7
4.1	Configuration	8
5	Group Contact Storage	9
5.1	Installation and Configuration	10
5.2	Group Folders	10
5.3	Global Address Book	11
6	Imap Util Library	11
6.1	Usage	11
6.2	Configuration	12
7	Ldap Client Library	12
7.1	Configuration	13
7.2	Metrics	16
8	Main Authentication Password	18
8.1	Configuration	18
9	Minimal API for user data	19
9.1	Minimal API Framework	19
9.1.1	Concept	19
9.1.2	Security Layer	20
9.1.3	Authentication	20
9.1.4	User Consent	20
9.1.5	Database	20
9.1.6	JSON API for Client, Consent and Token Handling	21
9.1.6.1	Parameter	21
9.1.6.1.1	Example Default Consent Response	21
9.1.6.1.2	Example Client Response	21
9.1.6.2	JSON Error Codes	22
9.1.7	Accessing User Data via REST API	22

9.1.7.1	Parameter	22
9.1.7.2	General Handling	22
9.1.7.3	Claim readMail	22
9.1.7.3.1	Example Quota Info Response	22
9.1.7.3.2	Example Current Mail Info Response	23
9.1.7.3.3	Example Status of Mail Folder Response	23
9.1.7.3.4	Example E-Mail Details Response	23
9.1.7.4	Claim readCalendar	23
9.1.7.4.1	Example Calendar Events Response	23
9.1.7.5	HTTP Response Codes	24
9.1.8	Command Line Tool for Fetching the History	24
9.1.9	Configuration	25
10	MX SPF Record Checker	26
10.1	MX SPF Record Checker Framework	26
10.1.1	Configuration	27
11	Onboarding Wizard Change Login Details	27
11.0.1	Configuration (Middleware)	28
12	Send SMS via Twilio	29
12.1	Requirements	29
12.2	Configuration	29
13	Server Metrics	29
13.1	Metrics for HTTP	29
13.1.1	Configuration	30
13.2	Metrics for IMAP	34
13.2.1	Configuration	34
14	Sql Client Library	35
14.1	Configuration	35
14.2	Sample Configuration	35
15	Trusted Identity for external systems	37
15.1	Trusted Identity Provider	37
15.1.1	JSON Web Token	37
15.1.2	HTTP API	39
15.1.2.1	Example Sign UI Usage for Upsell	39
15.1.2.2	Example Encrypt UI Usage for Upsell	39
15.1.3	Configuration	39
15.1.3.1	Example Key Configuration	41
15.1.4	Caching	42
16	Unsubscribe Mailing Lists	43
16.1	Unsubscribe Mailing Lists Framework	43
16.1.1	Prerequisites	43
16.1.2	Safe mode	43
16.1.3	Unsafe mode	43
16.1.4	Configuration	43
16.1.5	Vade Connector	43
16.1.5.1	Example 1	44
16.1.5.2	Example 2	44
17	Shipped Version	45
17.1	Package open-xchange-authentication-masterpassword	45
17.1.1	Installation	45
17.1.2	Configuration	45
17.2	Package open-xchange-ldap-client	45
17.2.1	Installation	45

17.2.2	Configuration	45
17.3	Package open-xchange-ldap-client-test	45
17.3.1	Installation	46
17.4	Package open-xchange-metrics-http	46
17.4.1	Installation	46
17.4.2	Configuration	46
17.5	Package open-xchange-metrics-imap	46
17.5.1	Installation	46
17.5.2	Configuration	46
17.6	Package open-xchange-minimal-api	47
17.6.1	Installation	47
17.6.2	Configuration	47
17.7	Package open-xchange-minimal-api-calendar	47
17.7.1	Installation	47
17.8	Package open-xchange-minimal-api-jwt	47
17.8.1	Installation	47
17.9	Package open-xchange-minimal-api-mail	48
17.9.1	Installation	48
17.10	Package open-xchange-plugins-antiphishing	48
17.10.1	Installation	48
17.10.2	Configuration	48
17.11	Package open-xchange-plugins-antiphishing-vadesecure	48
17.11.1	Installation	48
17.11.2	Configuration	49
17.12	Package open-xchange-plugins-blackwhitelist	49
17.12.1	Installation	49
17.12.2	Configuration	49
17.13	Package open-xchange-plugins-blackwhitelist-sieve	49
17.13.1	Installation	49
17.13.2	Configuration	49
17.14	Package open-xchange-plugins-contact-storage-group	50
17.14.1	Installation	50
17.14.2	Configuration	50
17.15	Package open-xchange-plugins-contact-whitelist-sync	50
17.15.1	Installation	50
17.15.2	Configuration	50
17.16	Package open-xchange-plugins-mx-checker	50
17.16.1	Installation	51
17.16.2	Configuration	51
17.17	Package open-xchange-plugins-onboarding-maillogin	51
17.17.1	Installation	51
17.17.2	Configuration	51
17.18	Package open-xchange-plugins-trustedidentity	51
17.18.1	Installation	51
17.18.2	Configuration	51
17.19	Package open-xchange-plugins-unsubscribe	52
17.19.1	Installation	52
17.19.2	Configuration	52
17.20	Package open-xchange-plugins-unsubscribe-vadesecure	52
17.20.1	Installation	52
17.20.2	Configuration	52
17.21	Package open-xchange-sms-twilio	52
17.21.1	Installation	53
17.21.2	Configuration	53
17.22	Package open-xchange-sql-client	53
17.22.1	Installation	53
17.22.2	Configuration	53
17.23	Package open-xchange-util-imap	53



17.23.1 Installation	53
A Configuration Files	53

1 General Information

1.1 Warnings

Warning

This preview delivery is not for productive usage and not affected by service-level agreements.

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:

Build for patch 2022-12-30 (6203)

1.3 Install Package Repository

This delivery is part of a restricted preview software repository:

<https://software.open-xchange.com/components/plugins/preview/1.7.2/DebianBuster>
<https://software.open-xchange.com/components/plugins/preview/1.7.2/DebianBullseye>
<https://software.open-xchange.com/components/plugins/preview/1.7.2/RHEL7>

1.4 Build Dependencies

This delivery was build with following dependencies:

AppSuite:node-10,frontend-7.10.6-rev21,backend-7.10.6-rev32

1.5 Notice

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`.

```

1  # Setting to control the used connector for a specific user
2  # This setting is config-cascade aware to support different implementations for each user.
3  # Default is <none> which means that the feature is disabled for a user
4  # To enable vade secure com.openexchange.plugins.antiphishing.connector=
5    plugins_antiphishing_vadesecure
6  com.openexchange.plugins.antiphishing.connector=
7
8  # Setting to enable/disable the antiphishing capability
9  # This setting is config-cascade aware to support different implementations for each user.
10 # Default is false which means that the feature is disabled for a user
11 com.openexchange.plugins.antiphishing.enabled=false
12
13 # Setting to enable/disable the antiphishing mta_capability
14 # If true, the user has the ability to choose antiphishing at the MTA level
15 # This setting is config-cascade aware to support different implementations for each user.
16 # Default is false which means that the feature is disabled for a user
17 com.openexchange.plugins.antiphishing.mta_capability=false
18
19 # Setting to enable/disable the antiphishing at the mta level
20 # If true, an antiphishing check will take place at the MTA level
21 # This setting is config-cascade aware to support different implementations for each user.
22 # Additionally, this property can be set by the user in the UI
23 # Default is false which means that the feature is disabled for a user
24 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 SUCCESS or ERROR with an appropriate message.

2.2.1 Example Config Request

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

Example Config Response

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

2.2.2 Example Get Request

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

Example Get Response

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

2.2.3 Example Update Request

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

Example Update Response

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

2.2.4 Example Linkcheck Request

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

Example Linkcheck Response

```
1 {"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`

```

1  # The customer name as provided by VadeSecure; required to access Phishing API
2  # Default: NONE
3  # Config-cascade aware: true
4  # Lean: false
5  com.openexchange.plugins.antiphishing.vadesecure.name.passcrypt=<Customer name provided by
6      VadeSecure>
7
7  # The customer license provided by VadeSecure; required to access Phishing API
8  # Default: NONE
9  # Config-cascade aware: true
10 # Lean: false
11 com.openexchange.plugins.antiphishing.vadesecure.license.passcrypt=<Customer license
12     provided by VadeSecure>
13
13 # Setting to change the VadeSecure IsItPhishing API URL
14 # Default: https://iip.eu.vadesecure.com/api/v2/url
15 # Config-cascade aware: true
16 # Lean: true
17 com.openexchange.plugins.antiphishing.vadesecure.phishing_url=https://iip.eu.vadesecure.
18     com/api/v2/url
19
19 # Setting to change the VadeSecure GRAPH Authentication API URL
20 # Default: https://api.vadesecure.com/oauth2/v2/token
21 # Config-cascade aware: false
22 com.openexchange.plugins.antiphishing.vadesecure.graph_url=https://api.vadesecure.com/
23     oauth2/v2/token
24
24 # Setting to change the VadeSecure connector identifier referenced in plugins-antiphishing
25     .properties / com.openexchange.plugins.antiphishing.connector
26 # Default: "plugins_antiphishing_vadesecure"
27 # Config-cascade aware: true
28 # Lean: true
29 com.openexchange.plugins.antiphishing.vadesecure.identifier=
30     plugins_antiphishing_vadesecure
31
31 # If set to true, the URL will always be crawled and analyzed, even if it can trigger
32     collateral damages (such as unsubscribing a user, canceling an order, etc.).
33 # If set to false, the service checks whether the URL may cause collateral damage to the
34     end user (unsubscribe, order confirmation, etc.). If so, the URL is not crawled and
35     NOT_EXPLORED is returned in the response.
36 # Default: false
37 # Config-cascade aware: true
38 # Lean: true
39 com.openexchange.plugins.antiphishing.vadesecure.force=false
40
40 # Vade Secure IsItPhishing Smart mode enables URL anonymization. Typically, this is meant
41     to
42 # replace any unique-ID like tokens in a URL by random characters, to prevent side effects
43     when crawling certain URLs, which if visited, could trigger unwanted actions:
44     unsubscription, cancelation, etc.
45 # Set to true to enable the smart mode. If set to false, URLs will be crawled in the way
46     they were originally provided. If argument randomization fails, the URL is not crawled
47     and NOT_EXPLORED is returned. # Default: "plugins_antiphishing_vadesecure"
48 # NOTE: Vade Secure strongly recommends enabling the smart parameter to true, so that the
49     API can trigger token anonymization, to try and prevent any collateral damages.
50 # Default: false
51 # Config-cascade aware: true
52 # Lean: true
53 com.openexchange.plugins.antiphishing.vadesecure.smart=true
54

```

```

46 # Timeout in milliseconds, with a minimum value of 1000. Once timeout is reached, TIMEOUT
    response is returned.
47 # Default: 3000
48 # Config-cascade aware: true
49 # Lean: true
50 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	<code>com.openexchange.plugins.blackwhitelist,</code> <code>com.openexchange.plugins.blackwhitelist.connector.sieve,</code> <code>com.openexchange.plugins.blackwhitelist.clt,</code> <code>com.openexchange.plugins.blackwhitelist.json</code>
Package(s)	<code>open-xchange-plugins-blackwhitelist,</code> <code>open-xchange-plugins-blackwhitelist-sieve</code>
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` provides 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.

```

1 # Identifier of this blackwhitelist connector: plugins_blackwhitelist_sieve
2 # Setting to control the rulename to be set and checked as a antispam value inside the
    sieve rules
3 # Default: Blacklist
4 # Config-cascade aware: true
5 # Lean: true
6 com.openexchange.plugins.blackwhitelist.connector.sieve.rulename=Blacklist
7
8 # Setting to control wether the blacklisted mails should be moved to SPAM or deleted
    directly
9 # If set to true, mails are moved to SPAM
10 # If set to false, mails are deleted
11 # Default: true
12 # Config-cascade aware: true
13 # Lean: true
14 com.openexchange.plugins.blackwhitelist.connector.sieve.moveToSpam=true
15
16 # Setting to check if memory backed test System should be started
17 # This connector is identified by plugins_blwl_test
18 # Default: false
19 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
<code>listblackwhitelist</code>	List black and whitelist entries of a user
<code>searchblackwhitelist</code>	Search for black and whitelist entries of a user
<code>deleteblackwhitelist</code>	Delete entries for black and whitelist of user
<code>addblackwhitelist</code>	Add entries for black and whitelist of user

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

```

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

```

3.3 Configuration

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

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

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

```

1  # Setting to control the used connector for a specific user
2  # This setting is config-cascade aware to support different implementations for each user.
3  # Default is <none> which means that the feature is disabled for a user
4  com.openexchange.plugins.blackwhitelist.connector=
5
6  # Setting to check if memory backed test System should be started
7  # This connector is identified by plugins_blwl_test
8  # Default: false
9  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:

```
1 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:

```
1 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:

```
1 # This setting enables or disables special handling for the ContactCollectionFolder
2 # If set to true, the contactCollectFolder is ignored and contacts in this folder
3 # are not added to the whitelist. Contacts moved to this folder are also removed from the
  whitelist
4 # If set to false, the contactCollectFolder is handled like any other folder.
5 # config-cascade aware
6 # Default: true
7 com.openexchange.plugins.contacts.whitelist.ignoreContactCollectFolder=true
8
9 # This setting is used to set the connector for the contact sync.
10 # Currently available options are:
11 # <not-set> (this will disable the sync for the user)
12 # rdb
13 # Default: <not-set>
14 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 being **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.

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

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`:

```

1  # Pool to be used
2  com.openexchange.plugins.contacts.whitelist.rdb.pool=contact-whitelist-pool
3
4  # normal or tombstone
5  com.openexchange.plugins.contacts.whitelist.rdb.strategy=normal
6
7  # table name
8  com.openexchange.plugins.contacts.whitelist.rdb.tableName=senderwl
9
10 # Name of the column used for the primary mail
11 com.openexchange.plugins.contacts.whitelist.rdb.primaryAddressColumnName=rcpt
12
13 # Name of the column used for the contact mails
14 com.openexchange.plugins.contacts.whitelist.rdb.contactMailColumnName=sender
15
16 # Name of the column used for the individual contactIds
17 com.openexchange.plugins.contacts.whitelist.rdb.contactIdColumnName=contactid
18
19 # Name of the deleted_at column if tombstone is enabled
20 com.openexchange.plugins.contacts.whitelist.rdb.tombstone.deletedAtColumnName=deleted_at
21
22 # Name of the updated_at column if tombstone is enabled
23 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 Login-Handler via `/opt/open-xchange/etc/plugins-contacts-whitelist-migration.properties`:

```

1  # Defines the strategy of the automatic migration
2  # Can be either
3  #   <not-set> which disables the automatic migration
4  #   once
5  #   time:<timeinmillis>
6  # Default: <not-set>
7  #
8  # Examples
9  # If sync should happen once a day:
10 # com.openexchange.plugins.contacts.whitelist.migration.strategy=time:86400000
11 # If sync should happen once a week
12 # com.openexchange.plugins.contacts.whitelist.migration.strategy=time:604800000
13 com.openexchange.plugins.contacts.whitelist.migration.strategy=
14
15 # Setting, if a warning should appear in the logs, if a user has more than configured
16 #   contacts in one folder.
17 # Default: 10000
18 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:

```
1 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:

```
1 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:

```
1 # Configures whether the group contact storage is enabled for a context or not.
2 # Default: false
3 com.openexchange.plugins.contact.storage.group.enabled=false
4
5 # Defines an optional list of those groups for which no group contact folder should
6 # be used, as a comma-separated string of the identifiers of those groups that should
7 # be excluded. The groups "All Users", "All Guests" and the "Standard Group" are
8 # always excluded.
9 # Default: <empty>
10 com.openexchange.plugins.contact.storage.group.excludedGroups=
11
12 # Defines if the display name of the groups should be used to create the folder
13 # names in the folder tree.
14 # If set to <true>, the displayname is used
15 # If set to <false>, the group name is used
16 # The Group Names are limited by the property CHECK_GROUP_UID_REGEX
17 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.



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, especially 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 auto-complete 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	<code>com.openexchange.util.imap</code>
Package(s)	<code>open-xchange-util-imap</code>
Required capabilities	<code>none</code>
Available since	<code>1.3.0-rev3</code>

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

```

1
2 public final class ProjectActivator extends ActivatorTemplate {
3     @Override
4     protected void registerServices(final Registrar registrar) {
5         registrar
6             .service(JavaMailImapAuthenticator.class, properties(
7                 ImapAuthenticator.CONFIG_PREFIX_PROPERTY, "com.openexchange.my.project.name.
8                     imap"
9             ))
10            // ...
11    }
12 }

```

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

7 Ldap Client Library

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

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 `".yaml"` or `".yml"` 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`

```

1  # The top-level key is the identifier of the pool, which can be
2  # any string of text and is being used by the bundles and applications
3  # to access that pool configuration.
4  # Typically, those are fixed or need to be configured in the bundles
5  # that use this library.
6  #
7  # When Java Security Manager support is enabled, files that are referenced
8  # in these configuration files must be in a directory that is already
9  # whitelisted, or in a subdirectory thereof, such as
10 # /opt/open-xchange/etc/
11 #
12 # A good candidate would be something along the lines of
13 # /opt/open-xchange/etc/ldap-files/
14 #
15 # Otherwise, the filename or its directory must be put into a new .list
16 # file in the folder
17 # /opt/open-xchange/etc/security/
18 # with e.g. the following content:

```

```

19 #
20 # file:/etc/trust.jks
21 #
22 pool1:
23   trust-store:
24     # SSL: path to the JKS trust store file that contains the anchors
25     file: /etc/trust.jks
26     # SSL: indicates whether to reject certificates if the current time
27     # is outside the validity window for the certificate
28     validity: true
29   key-store:
30     # SSL: path to the JKS client key store file that contains the key
31     file: /etc/private.jks
32     # SSL: password to access the keystore and the key
33     password: foobar
34     # SSL: alias of the key to use
35     alias: key
36 # Configure a read/write pool with different settings for read operations
37 # and for write operations (i.e. different pools of LDAP servers).
38 # Here comes the part for the read operations:
39 read:
40   # Use a failover cluster of two nodes:
41   failover:
42     - ldap1.example.com
43     - ldap2.example.com
44   # Pool connection management
45   # -----
46   # When creating a connection pool, you may specify an initial number of
47   # connections (pool-min) and a maximum number of connections (pool-max).
48   # The initial number of connections is the number of connections that should
49   # be immediately established and available for use when the pool is created.
50   # The maximum number of connections is the largest number of unused connections
51   # that may be available in the pool at any time.
52   # Whenever a connection is needed, whether by an attempt to check out a
53   # connection or to use one of the pool's methods to process an operation,
54   # the pool will first check to see if there is a connection that has already
55   # been established but is not currently in use, and if so then that connection
56   # will be used.
57   # If there aren't any unused connections that are already established, then
58   # the pool will determine if it has yet created the maximum number of
59   # connections, and if not then it will immediately create a new connection
60   # and use it.
61   # If the pool has already created the maximum number of connections, then the
62   # pool may wait for a period of time (as configured using 'maxWaitTimeMillis' below,
63   # which has a default value of zero to indicate that it should not wait at all)
64   # for an in-use connection to be released back to the pool.
65   # If no connection is available after the specified wait time (or there should
66   # not be any wait time), then the pool may automatically create a new connection
67   # to use if 'createIfNecessary' is true (which is the default).
68   # If it is able to successfully create a connection, then it will be used.
69   # If it cannot create a connection, or if 'createIfNecessary' is set to false,
70   # then an error will be thrown.
71   # Note that the maximum number of connections specified when creating a pool
72   # refers to the maximum number of connections that should be available for use
73   # at any given time.
74   # If 'createIfNecessary' is set to true, then there may temporarily be more
75   # active connections than the configured maximum number of connections.
76   # This can be useful during periods of heavy activity, because the pool will
77   # keep those connections established until the number of unused connections
78   # exceeds the configured maximum.
79   # If you wish to enforce a hard limit on the maximum number of connections so
80   # that there cannot be more than the configured maximum in use at any time,
81   # then set 'createIfNecessary' to false to indicate that the pool should not
82   # automatically create connections when one is needed but none are available,
83   # and you may also want to set 'maxWaitTimeMillis' to a maximum wait time to allow
84   # the pool to wait for a connection to become available rather than throwing
85   # an exception if no connections are immediately available.
86   pool-min: 10
87   pool-max: 50
88   maxConnectionAgeMillis: 30000
89   maxWaitTimeMillis: 500
90   createIfNecessary: true

```

```

91      # Specifies whether certain operations that should be retried on a newly-created
92      # connection if the initial attempt fails in a manner that indicates that the
93      # connection used to process the request may no longer be valid.
94      # Only a single retry will be attempted for any operation.
95      retryFailedOperations: true
96  # Here comes the part for the write operations:
97  write:
98      host: ldap0.example.com
99      pool-min: 1
100     pool-max: 10
101     maxConnectionAgeMillis: 60000
102     maxWaitTimeMillis: 1000
103     createIfNecessary: false
104     retryFailedOperations: false
105     # Specifies whether the pool should attempt to abandon any request for which
106     # no response is received in the maximum response timeout period:
107     abandonOnTimeout: true
108     # Specifies the maximum length of time in milliseconds that a connection attempt
109     # should be allowed to continue before giving up.
110     # A value of zero (default) indicates that there should be no connect timeout.
111     connectionTimeoutMillis: 3000
112     # Specifies the maximum length of time in milliseconds that an operation should
113     # be allowed to block while waiting for a response from the server.
114     # A value of zero indicates that there should be no timeout.
115     responseTimeoutMillis: 5000
116     # Specifies whether to use the SO_KEEPALIVE option for the underlying sockets
117     # used by associated connections.
118     keepAlive: true
119     # Specifies whether to use the TCP_NODELAY option for the underlying sockets.
120     tcpNoDelay: true
121     # Specifies whether to operate in synchronous mode, in which at most one
122     # operation may be in progress at any time on a given connection.
123     # When using asynchronous mode, a background thread takes care of multiplexing
124     # and dispatching all the operations on connections that are shared for
125     # multiple operations.
126     synchronousMode: true
127     # Specifies the length of time in milliseconds between periodic background
128     # health checks against the available connections in this pool.
129     healthCheckIntervalMillis: 120000
130     # Specifies whether associated connections should attempt to follow any
131     # referrals that they encounter.
132     followReferrals: true
133     # Specifies the maximum number of hops that a connection should take when
134     # trying to follow a referral, must be greater than zero when 'followReferrals'
135     # is true.
136     referralHopLimit: 1
137     # Specifies the maximum size in bytes for an LDAP message that a connection
138     # will attempt to read from the directory server.
139     # If it encounters an LDAP message that is larger than this size, then the
140     # connection will be terminated.
141     # Disabled when not specified or set to 0.
142     maxMessageSize: 1024
143
144  pool2:
145     # A failover pool that uses the same set of servers for read and for
146     # write operations.
147     failover:
148         - ldap0.example.com
149         - ldap1.example.com
150     pool-min: 5
151     pool-max: 20
152     trust-store:
153         file: /etc/trust.jks
154     key-store:
155         file: /etc/private.jks
156
157  pool3:
158     # A simple single-host setup
159     host: ldap.example.com
160     pool-min: 5
161     pool-max: 20
162

```

```

163 pool4:
164 # A load-balancing setup that will use a round-robin algorithm to
165 # select the server to which the connection should be established.
166 # Any number of servers may be included, and each request will
167 # attempt to retrieve a connection to the next server in the list,
168 # circling back to the beginning of the list as necessary.
169 # If a server is unavailable when an attempt is made to establish
170 # a connection to it, then the connection will be established to
171 # the next available server in the set.
172 round-robin:
173   - host: ldap1.example.com
174     port: 10389
175     responseTimeoutMillis: 5000
176   - host: ldap2.example.com
177     port: 10389
178     responseTimeoutMillis: 12000
179 pool-min: 10
180 pool-max: 50
181
182 pool5:
183 # A DNS RR setup handles the case in which a given hostname may
184 # resolve to multiple IP addresses.
185 # Note that while a setup like this is typically referred to as
186 # "round-robin DNS", this option does not strictly require DNS (as names
187 # may be resolved through alternate mechanisms like a hosts file or an
188 # alternate name service), and it does not strictly require round-robin
189 # use of those addresses (as alternate ordering mechanisms like
190 # 'random' or 'failover' may be used).
191 dns-round-robin:
192   host: ldap.example.com
193   # The selection mode that should be used if the hostname resolves
194   # to multiple addresses.
195   # Possible values:
196   # - random: the order of addresses will be randomized for each attempt
197   # - failover: addresses will be consistently attempted in the order
198   #             they are retrieved from the name service.
199   # - round-robin: connection attempts will be made in a round-robin order
200   selection-mode: random
201   # Only use DNS if set to 'true'.
202   # If set to 'false' then the operating system's hostname resolution
203   # service will be used, which may include a hosts file.
204   only-dns: false
205   # The maximum length of time in milliseconds to cache addresses resolved
206   # from the provided hostname.
207   # Caching resolved addresses can result in better performance and can
208   # reduce the number of requests to the name service.
209   # A value that is less than or equal to zero indicates that no caching
210   # should be used.
211   cache-timeout: 1440000
212   pool-min: 5
213   pool-max: 20
214
215 pool6:
216 # A failover pool that uses the same set of servers for read and for
217 # write operations, as well as StartTLS
218 failover:
219   - ldap0.example.com
220   - ldap1.example.com
221 pool-min: 5
222 pool-max: 20
223 starttls: true
224 trust-store:
225   file: /etc/trust.jks
226 key-store:
227   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_max_connections	The maximum number of connections that may be available in the pool at any time
appsuite_ldap_pool_available_connections	The number of connections currently available for use in the pool
appsuite_ldap_pool_defunct_connections	The number of connections that have been closed as defunct (i.e., they are no longer believed to be valid)
appsuite_ldap_pool_expired_connections	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)
appsuite_ldap_pool_unneeded_connections	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_failed_attempts_to_checkout	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_failed_attempts_to_create	The number of failed attempts to create a connection for use in the connection pool
appsuite_ldap_pool_released_valid_connections	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_successful_attempts_to_checkout	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_successful_attempts_to_checkout_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_successful_attempts_to_create_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_successful_attempts_to_checkout_existing	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_successful_creations	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:

```

1 mypool:
2   read:
3     host: localhost
4     port: 10389
5   write:
6     host: localhost
7     port: 11389

```

metrics will be exported for the following tags:

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

Example in Grafana:

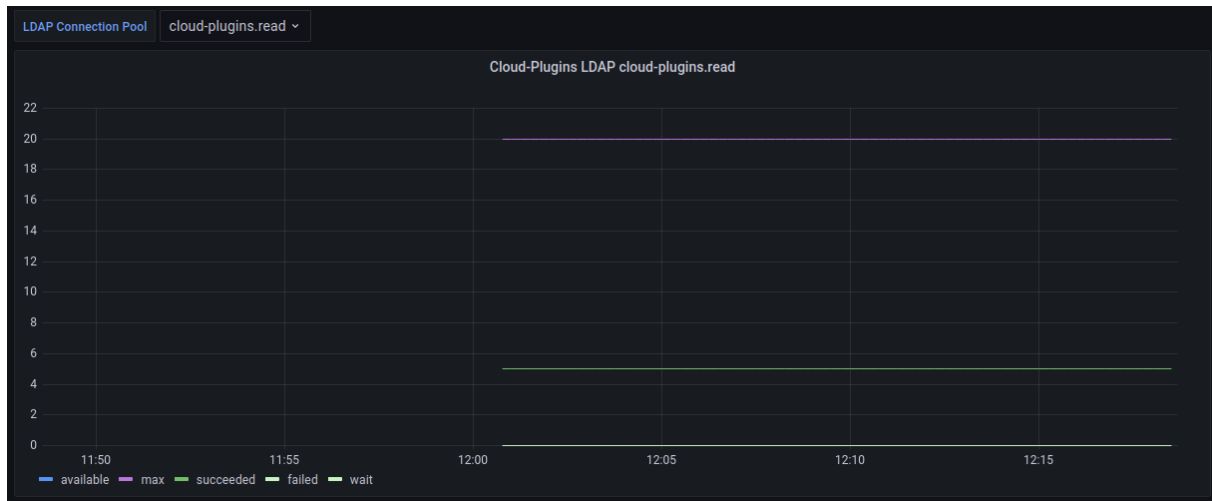


Figure 1: LDAP Client Library Metrics in Grafana

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

```

1  # Configuration file for the master password authentication plugin
2  #
3  # DO NOT USE IN PRODUCTION !
4  #
5
6  # The clear text password to authenticate all users.
7  # Mandatory.
8  # Example:
9  # com.openexchange.authentication.masterpassword.password=supersecret
10 com.openexchange.authentication.masterpassword.password=
11
12 # The default value for the context when it is not specified.
13 # Optional and defaults to using the "defaultcontext" mapping.
14 #com.openexchange.authentication.masterpassword.default.context=
15
16 # Whether the username portion of the login should be lowercased
17 # before being looked up in the user database.
18 # Optional and defaults to false
19 #com.openexchange.authentication.masterpassword.lowercase=false
20
21 # Whether the context name portion of the login should be lowercased
22 # before being looked up in the context database.
23 # Optional and defaults to false
24 #com.openexchange.authentication.masterpassword.lowercase.context=false
25

```



```

26 # Whether to use the complete login string as the username,
27 # e.g. login "foo@bar.com" -> user name "foo@bar.com" and
28 # context name "bar.com"
29 # Optional and defaults to false
30 #com.openexchange.authentication.masterpassword.use.full.login.info=false
31
32 # Whether to use the complete login string for the context name,
33 # e.g. login "foo@bar.com" -> context name "foo@bar.com"
34 # Optional and defaults to false
35 #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 too 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 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:

```
1 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:

```
1 {"data":{"token":"eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIwM2E2Y...LCJyZWFrTWFPbCI6dHJ1ZX0.9xQ6_YXDxZlZsZWAZzLFRqgTptYERTl_-Qma80UsWY"}}
```

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

```
1  HEADER:ALGORITHM & TOKEN TYPE
2  {
3    "alg": "HS256"
4  }
5
6  PAYLOAD:DATA
7  {
8    "sub": "03a6bd595320422288e1767222603bce",
9    "client": "example",
10   "readMail": true
11  }
12
13  VERIFY SIGNATURE
14  HMACSHA256(
15    base64UrlEncode(header) + "." +
16    base64UrlEncode(payload),
17
18  ) 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:

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

Action URL Suffix	Description
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

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

9.1.6.1.1 Example Default Consent Response

```
1 {
2   "data": {
3     "client": "example",
4     "defaultConsent": true // or false
5   }
6 }
```

9.1.6.1.2 Example Client Response

```
1 {
2   "data": [
3     {
4       "client": "example",
5       "name": "Special Example Client",
6       "configuredClaims": [
7         "readMail"
8       ],
9       "configuredClaimsButNotInStorage": [
10      ],
11      "configuredAndActiveClaims": [
12        "readMail"
13      ],
14      "configuredAndDisabledClaims": [
15      ]
16    }
17  ]
18 }
```

```

16     }
17   ]
18 }

```

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:

```

1  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
<code>/mail/v1/quota</code>	Info about the current mail quota.
<code>/mail/v1/info</code>	Info about the current mail of the user which internally is handled.
<code>/mail/v1/stats</code>	Status of total and unread count in inbox.
<code>/mail/v1/stats/{folder}</code>	Status of total and unread count in folder.
<code>/mail/v1/inbox</code>	Details about the last 5 by date emails in INBOX.
<code>/mail/v1/inbox/{amount}</code>	Details about the amount of mails by date emails in INBOX.
<code>/mail/v1/emails/{folder}</code>	Details about the last 5 by date emails in folder.
<code>/mail/v1/emails/{folder}/{amount}</code>	Details about the last amount of emails in the folder by date.

9.1.7.3.1 Example Quota Info Response

```

1  {

```

```

2  "storageQuota": {
3      "used": -1,
4      "limit": -1
5  },
6  "messageQuota": {
7      "used": -1,
8      "limit": -1
9  }
10 }
```

9.1.7.3.2 Example Current Mail Info Response

```

1  {
2      "mail" : "string",
3      "displayName" : "string"
4  }
```

9.1.7.3.3 Example Status of Mail Folder Response

```

1  {
2      "total" : 10,
3      "unread" : 5
4  }
```

9.1.7.3.4 Example E-Mail Details Response

```

1  {
2      "amount" : 5 // the number of returned mails
3      "mails" :[
4          {
5              "id": "string",
6              "subject": "string",
7              "from": "string",
8              "seen": true,
9              "arrival": 0
10         },...
11     ]
12 }
```

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

```

1  [
2      {
3          "location": "The Doctor Office",
4          "created": 1581457892837,
5          "description": "Need to get my ears checked.",
6          "attendees": [
7              {
8                  "uri": "mailto:john.doe@ox.com",
9                  "cn": "Doe, John",
10                 "email": "john.doe@ox.com",
11                 "entity": 3,
12                 "cuType": "INDIVIDUAL",
13                 "partStat": "ACCEPTED",
14                 "folder": "31"

```

```

15     }
16   ],
17   "modifiedBy": {
18     "uri": "mailto:john.doe@ox.com",
19     "cn": "Doe, John",
20     "email": "john.doe@ox.com",
21     "entity": 3
22   },
23   "attendeePrivileges": "DEFAULT",
24   "transp": "OPAQUE",
25   "createdBy": {
26     "uri": "mailto:john.doe@ox.com",
27     "cn": "Doe, John",
28     "email": "john.doe@ox.com",
29     "entity": 3
30   },
31   "folder": "cal://0/31",
32   "summary": "Doctor",
33   "startDate": {
34     "tzid": "Europe/Berlin",
35     "value": "20200213T210000"
36   },
37   "organizer": {
38     "uri": "mailto:john.doe@ox.com",
39     "cn": "Doe, John",
40     "email": "john.doe@ox.com",
41     "entity": 3
42   },
43   "class": "PUBLIC",
44   "lastModified": 1581542567786,
45   "uid": "88f822e7-2f98-4c8c-8ff0-96c2f6aaf265",
46   "timestamp": 1581542567786,
47   "endDate": {
48     "tzid": "Europe/Berlin",
49     "value": "20200213T213000"
50   },
51   "calendarUser": {
52     "uri": "mailto:john.doe@ox.com",
53     "cn": "Doe, John",
54     "email": "john.doe@ox.com",
55     "entity": 3
56   },
57   "id": "3",
58   "sequence": 1,
59   "flags": [
60     "organizer",
61     "accepted"
62   ]
63 }
64 ]

```

Response	Description
200	The request was successful.
400	The amount is not in range from 1 to 100.
403	The token is not allowed 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:

```

1  usage: listminimalconsenthistory
2  -A,--adminuser <arg>          Admin username

```

```

3  -c,--context <arg>          A valid context identifier
4  -e,--end-time <arg>        End time in seconds since 1970-01-01
5                               00:00:00 UTC. Only consent given before this
6                               time is shown. If not set, all feedback
7                               since -s is shown.
8  -h,--help                  Prints a help text
9  -i,--userid <arg>          A valid user identifier
10 -n,--client-name <arg>      A valid (and used) client identifier. If not
11                             set, all clients are returned
12 -p,--port <arg>             The optional RMI port (default:1099)
13 -P,--adminpass <arg>        Admin password
14     --responsetimeout <arg> The optional response timeout in seconds
15                             when reading data from server (default: 0s;
16                             infinite)
17 -s,--server <arg>           The optional RMI server (default: localhost)
18 The command-line tool to list given consent of a user for the minimal api

```

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

```

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

```

9.1.9 Configuration

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

```

1  # The capability to control whether or not the user is allowed to access the API
2  # at all
3  #
4  # Optional, default value: false
5  #
6  # Example:
7  # com.openexchange.capability.minimalapi=true
8  com.openexchange.capability.minimalapi=false
9
10 # The clients names enabled for a user
11 # Must be provided as a comma separated list
12 #
13 # Optional, default value: ""
14 #
15 # Must be provided as a comma separated list
16 #
17 # Example:
18 # com.openexchange.plugins.minimal.api.clients=exampleClient,exampleClient2
19 com.openexchange.plugins.minimal.api.clients=
20
21 # The user-friendly name of a client
22 #
23 # Optional, default value: ""
24 #
25 # If not set, the client identifier is returned.
26 #
27 # Example:
28 # com.openexchange.plugins.minimal.api.exampleClient.name=Example Preview
29 com.openexchange.plugins.minimal.api.[client].name=
30
31 # The claims assigned to a client
32 #
33 # Optional, default value: ""
34 #
35 # Must be provided as a comma separated list
36 #
37 # Example:
38 # com.openexchange.plugins.minimal.api.exampleClient.claims=readMail
39 com.openexchange.plugins.minimal.api.[client].claims=
40

```

```

41 # Default consent if user has not yet decided on first access
42 # WARNING: It might be required by law to enforce user consent
43 #
44 # Optional, default value: false
45 #
46 # Example:
47 # com.openexchange.plugins.minimal.api.exampleClient.defaultconsent=true
48 com.openexchange.plugins.minimal.api.[client].defaultconsent=false
49
50 # Maximum amount of requests per second per source IP address if the token could not be
51 # validated from cache
52 #
53 # May be a decimal number.
54 #
55 # Optional, default value: 1.0
56 # Optional, default for client: 5.0
57 #
58 # Example:
59 # com.openexchange.plugins.minimal.api.ratelimit.requestsPerSecond=10.0
60 # com.openexchange.plugins.minimal.api.ratelimit.exampleClient.maxRequestsPerSecond=10.0
61 com.openexchange.plugins.minimal.api.ratelimit.requestsPerSecond=1.0
62
63 # Maximal time window, in milliseconds: after a given source IP address has not accessed
64 # the minimal API, its number of requests per second rate is reset.
65 #
66 # Optional, default value: 300000
67 # Optional, default for client: 300000
68 #
69 # Example:
70 # com.openexchange.plugins.minimal.api.ratelimit.maxRateTimeWindow=60000
71 # com.openexchange.plugins.minimal.api.ratelimit.exampleClient.maxRateTimeWindow=60000
72 com.openexchange.plugins.minimal.api.ratelimit.maxRateTimeWindow=300000
73
74 # Strategy to use for reacting to the inability to access the API for a given source
75 # IP address due to surpassing the maxRequestsPerSecond rate.
76 #
77 # Format: it must be one of:
78 # * fail-fast
79 # * block
80 # * timeout:...
81 #
82 # fail-fast
83 # if the rate limit is exceeded, the API will respond with a 401 Unauthorized
84 # block
85 # if the rate limit is exceeded, the API will block infinitely until the rate limit
86 # allows for another request to be performed
87 # timeout:...
88 # block until the specified timeout is reached, after which the API responds with a
89 # 401 Unauthorized
90 # if the timeout does not allow to get a new token in time, a 401 Unauthorized is
91 # returned
92 # The value after "timeout:" consists of a number followed by a time unit, examples:
93 # - timeout:400s ---> 400 seconds
94 # - timeout:1m -----> 1 minute
95 # - timeout:2000ms -> 2000 milliseconds
96 #
97 # If the token could be validated and is correct, the API will not return a
98 # 401 Unauthorized but a 429 Too Many Requests instead.
99 #
100 # Optional, default value: timeout:250ms
101 # Optional, default for client: timeout:500ms
102 #
103 # Example:
104 # com.openexchange.plugins.minimal.api.ratelimit.strategy=timeout:1s
105 # com.openexchange.plugins.minimal.api.ratelimit.exampleClient.strategy=timeout:5s
106 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`

```

1 # Determines which connector will be used for a user
2 # This setting is config-cascade aware to support different implementations for each user.
3 # Default is <none> which means that the feature is disabled for a user
4 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.

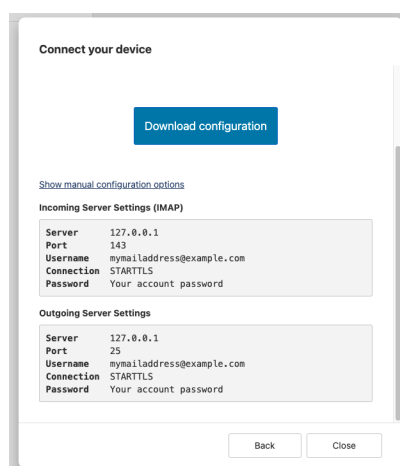


Figure 2: example wizard

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`

```

1  # Default value for overriding the login information displayed
2  # in the client onboarding.
3  #
4  # Possible values:
5  # email
6  #   uses the user's defaultSenderAddress
7  # attr:<name>
8  #   uses the user's attribute <name>
9  # login
10 #   uses the user's login, which is the same as if the
11 #   onboarding login was not overridden by this plugin
12 # login_name
13 #   uses the loginName attribute when possible, which is only the case
14 #   for session based logins (IMAP, SMTP) and for protocols that do not
15 #   create a session (CalDAV, CardDAV, EAS), it falls back on the login
16 #   instead
17 #
18 # This property is config cascade aware and must be set globally
19 # (in this file), and can then be overridden by context and/or by
20 # user.
21 #
22 # Note that for this feature to be enabled, one is also required
23 # to set one or more the following properties, depending on the
24 # client onboarding dialogs that need the login information to
25 # be overridden by this plugin:
26 # com.openexchange.client.onboarding.caldav.login.customsource=true
27 # com.openexchange.client.onboarding.carddav.login.customsource=true
28 # com.openexchange.client.onboarding.mail.imap.login.customsource=true
29 # com.openexchange.client.onboarding.mail.smtp.login.customsource=true
30 #

```

```
31 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

```

1  # Twilio accountSID
2  com.openexchange.plugins.sms.twilio.accountSID.secret=ACCOUNT_SID
3
4  # Twilio auth token
5  com.openexchange.plugins.sms.twilio.authtoken.secret=AUTH_TOKEN
6
7  # Twilio Message Service SID
8  com.openexchange.plugins.sms.twilio.messageservicesid.secret=SERVICE_SID
9
10 # Max message length. 1600 characters is Twilio's maximum
11 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:

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

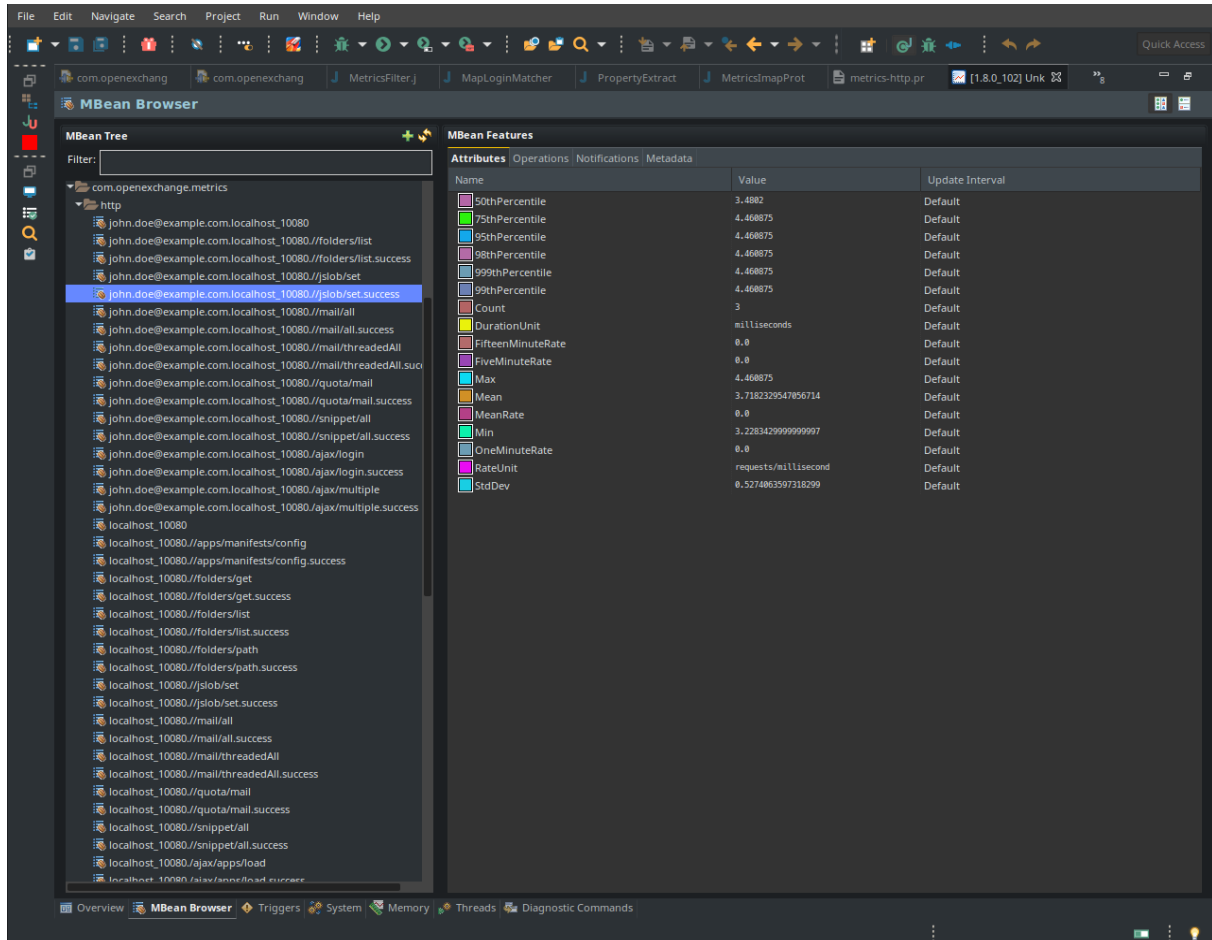


Figure 3: metrics for http

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

```
1 #
2 # The following property defines the various elements to use to compose the names of
3 # the metrics, to determine how to group them and what to see.
4 #
5 # The elements are separated by dots (".") and parsed individually, then replaced by
6 # their respective value for each inbound HTTP request to determine the name of
7 # the metric to update.
8 #
9 # Note that not all elements necessarily always result in a value as some are only
10 # present for specific types of HTTP requests, and others are optional (for example
11 # all the user information related ones that are only available when the HTTP request
12 # is authenticated or used in the context of an established Open-Xchange session).
13 # Values that are not available are skipped in the resulting name of the metric.
```

```

14 #
15 # For each component, here are the possible values to specify in this property:
16 # status
17 # =====
18 # Will be replaced by "success" or "error" depending on the result, for example:
19 # /api/rest/x/y/z -> success
20 #
21 # path
22 # =====
23 # If the HTTP is an AJAX API call, it will be replaced by "//module/action", and if not
24 # (e.g. accessing a servlet instead), it will be replaced with the servlet path.
25 #
26 # Examples:
27 # /ajax/folders?action=get&id=1,2,4 -> //folders/get
28 # /rest/api/x/y/z -----> /rest/api/x/y/z
29 #
30 # info
31 # =====
32 # Will be replaced with the servlet path info, i.e. the part of the URL that is behind
33 # the servlet path.
34 #
35 # Examples:
36 # /rest/api/users/john.doe@example.com -> john.doe@example.com
37 #
38 # session
39 # =====
40 # The value "session", "session_id" or "sessionid" will be replaced by the Open-Xchange
    session
41 # identifier, if applicable.
42 # For HTTP operations that are not authenticated, it will be left out.
43 #
44 # context_id
45 # =====
46 # The value "context_id" or "cid" will be replaced by the numeric context identifier of
    the
47 # user, if applicable.
48 # For HTTP operations that are not authenticated, it will be left out.
49 #
50 # user_id
51 # =====
52 # The value "user_id" or "cid" will be replaced by the numeric user identifier of the
53 # user within the context, if applicable.
54 # For HTTP operations that are not authenticated, it will be left out.
55 #
56 # login
57 # =====
58 # The value "login" will be replaced by the login the user entered to authenticate or the
59 # user identifier provided by an SSO mechanism, if applicable.
60 # For HTTP operations that are not authenticated, it will be left out.
61 #
62 # property(module)
63 # =====
64 # Will be replaced by the AJAX API module, if applicable.
65 #
66 # property(action)
67 # =====
68 # Will be replaced by the AJAX API module action, if applicable.
69 #
70 # header(...)
71 # =====
72 # Will be replaced by the value of an HTTP request header, the name of the header
73 # being specified between the parentheses.
74 # Note that header names are case sensitive.
75 #
76 # Example:
77 # header(Host).path -> appsuite01.example.com//folders/list
78 #
79 # parameter(...)
80 # =====
81 # Will be replaced by the value of an HTTP request parameter, the name of the
82 # parameter being specified between the parentheses.
83 #

```

```

84 # Example:
85 # header(Host).parameter(app).path -> appsuite01.example.com.io.ox/mail.//folders/list
86 #
87 # cookie(...)
88 # =====
89 # Will be replaced by the value of a cookie present in the HTTP request, the name of the
90 # cookie being specified between the parentheses.
91 #
92 # session(...)
93 # =====
94 # Will be replaced by the value of a parameter present in the user's Open-Xchange session,
95 # the name of the session parameter being specified between the parentheses.
96 #
97 # text(...)
98 # =====
99 # Specifies text that will be used as-is.
100 #
101 com.openexchange.metrics.http.elements=path.status
102
103 # When aggregation is enabled (by setting this value to true), each element as configured
104 # by the property com.openexchange.metrics.http.elements will be a metric in its own right
105 # and aggregated accordingly to its path.
106 # Without aggregation, each metric is "flat".
107 #
108 # For example, with the following configuration
109 #   com.openexchange.metrics.http.elements=header(Host).path.status
110 #   com.openexchange.metrics.http.aggregation=true
111 # each element will be a metric, namely:
112 # 1. header(Host)
113 # 2. header(Host).path
114 # 3. header(Host).path.status
115 #
116 # Specifically, results will look along the lines of the following, each being a metric:
117 # - appsuite01.example.com
118 # - appsuite01.example.com.//folders/list
119 # - appsuite01.example.com.//folders/list.success
120 #
121 # Each of those metrics except for the last one will be aggregating the measurements
122 # of their parent metrics.
123 #
124 com.openexchange.metrics.http.aggregation=false
125
126 # List of logins for which to create specific metrics.
127 # In order to be able to track and aggregate the metrics of specific users, the
128 # following property can be set to a (full) login name as entered by the user when
129 # authenticating or as provided by an SSO system if applicable.
130 #
131 # For each of the logins specified through this property, an additional set
132 # of metrics will be created, prefixing the elements that are defined in
133 # com.openexchange.metrics.http.elements
134 # with the login value.
135 #
136 # For example, the following configuration
137 #   com.openexchange.metrics.http.elements=header(host).path.status
138 #   com.openexchange.metrics.http.aggregation=true
139 #   com.openexchange.metrics.http.logins=jdoe@example.com
140 # will result in the following list of metrics:
141 # 1. header(Host)
142 # 2. header(Host).path
143 # 3. header(Host).path.status
144 # 4. login
145 # 5. login.header(Host)
146 # 6. login.header(Host).path
147 # 7. login.header(Host).path.status
148 #
149 # Specifically, results will look along the lines of the following, each being a metric:
150 # - appsuite01.example.com
151 # - appsuite01.example.com.//folders/list
152 # - appsuite01.example.com.//folders/list.success
153 # - jdoe@example.com
154 # - jdoe@example.com.appsuite01.example.com

```

```

155 # - jdoe@example.com.appsuite01.example.com.//folders/list
156 # - jdoe@example.com.appsuite01.example.com.//folders/list.success
157 #
158 # Without aggregation, the following configuration
159 #   com.openexchange.metrics.http.elements=header(host).path.status
160 #   com.openexchange.metrics.http.aggregation=false
161 #   com.openexchange.metrics.http.logins=jdoe@example.com
162 # will result in this list of metrics instead:
163 # 1. header(Host).path.status
164 # 2. login.header(Host).path.status
165 #
166 # Note that if this property is commented out (not set) or left empty,
167 # no such additional per-login metrics will be created, which is the default
168 # behavior.
169 #
170 # Multiple logins may be specified, either by separating them with whitespaces
171 # and/or commas, e.g.:
172 # com.openexchange.metrics.http.logins=john.doe@example.com, jane.doe@example.com
173 # or by specifying multiple properties as follows:
174 # com.openexchange.metrics.http.logins.1=john.doe@example.com
175 # com.openexchange.metrics.http.logins.2=jane.doe@example.com
176 # (both may also be combined).
177 #
178 # Furthermore, it is possible to use regular expressions and wildcards:
179 # - if a login contains * or ?, it is understood to be a wildcard
180 # - if a login is enclosed in /.../ or /.../i (case insensitive), it is understood
181 #   to be a regular expression
182 # Examples:
183 # com.openexchange.metrics.http.logins=*@example.com, /~j(ohn|ane)\.doe@example\.cm$/
184 #
185 # Being a wildcard, the following value would match all logins:
186 # com.openexchange.metrics.http.logins=*
187 #
188 com.openexchange.metrics.http.logins=
189
190 # List of paths and path patterns for which to maintain metrics.
191 #
192 # The following property specifies discrete paths, path wildcard patterns, or
193 # regular expressions that will be matched against the HTTP request paths, and
194 # only those that match will have metrics.
195 #
196 # If the property value contains * or ?, it will be understood as a wildcard pattern.
197 # If it starts with / and ends with / or /i (cae insensitive), it will be understood
198 # as a regular expression.
199 # If it is neither of those, it will be interpreted as an exact (string comparison) value.
200 #
201 # To enable metric collection for all URLs, use the following value:
202 # com.openexchange.metrics.http.path=*
203 #
204 # If the value is not defined or empty, no metrics will be collected:
205 # com.openexchange.metrics.http.path=
206 #
207 # Example:
208 # com.openexchange.metrics.http.path.1=/~/appsuite/.+/(boot|precore)\.js$/
209 # com.openexchange.metrics.http.path.2=/appsuite/api/apps/manifests
210 # com.openexchange.metrics.http.path.3=/appsuite/api/mail
211 #
212 com.openexchange.metrics.http.path=
213
214 # The behavior of the path matching above can be configured with the following property.
215 # Possible values:
216 # - whitelist: any URL path that matches one of the URL patterns configured
217 #   using com.openexchange.metrics.http.path will be measured with metrics;
218 #   any URL path that does not, will not be measured with metrics
219 # - blacklist: any URL path that does not matches one of the URL patterns configured
220 #   using com.openexchange.metrics.http.path will be measured with metrics
221 #
222 # When omitted, left empty or invalid, the default mode is whitelist
223 #
224 # Example:
225 # com.openexchange.metrics.http.path.mode=blacklist
226 #

```

```
227 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:

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

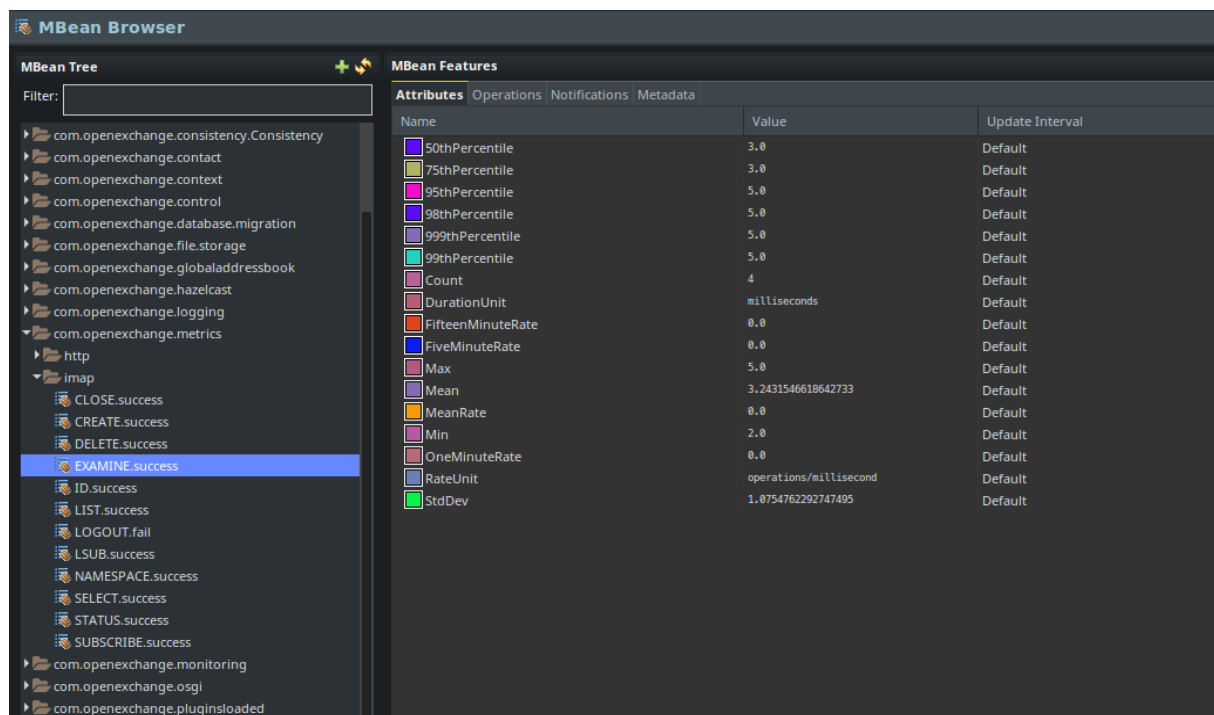


Figure 4: metrics for imap

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
```

```
1 # Configure whether to enable metrics for IMAP operations.
2 # When this property is omitted (commented out) or set to false, or empty,
```

```

3 # IMAP metrics will not be collected.
4 com.openexchange.metrics.imap.enable=false
5
6 # The number of threads to use to process IMAP operation results,
7 # updating metrics.
8 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

```

1 # Comma separated list of drivers to read into the system
2 # As the sql-client is very early, it may happen that the expected driver is not yet
   registered.
3 # To work around this issue, the following list of drivers will be read before any
   connection is
4 # created.
5 #
6 # Default: com.mysql.jdbc.Driver
7 com.openexchange.sql.client.drivers=com.mysql.jdbc.Driver
8
9 # The sql-client.d folder can be changed via
10 # openexchange.sql.client.dir=

```

14.2 Sample Configuration

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

```

1 # The top-level key is the identifier of the pool, which can be
2 # any string of text and is being used by the bundles and applications
3 # to access that pool configuration.
4 # Typically, those are fixed or need to be configured in the bundles
5 # that use this library.
6 #
7 # When Java Security Manager support is enabled, files that are referenced
8 # in these configuration files must be in a directory that is already
9 # whitelisted, or in a subdirectory thereof, such as
10 # /opt/open-xchange/etc/
11 #
12 # A good candidate would be something along the lines of
13 # /opt/open-xchange/etc/sql-files/
14 #
15 # Otherwise, the filename or its directory must be put into a new .list
16 # file in the folder
17 # /opt/open-xchange/etc/security/
18 # with e.g. the following content:
19 #
20 # file:/etc/trust.jks
21 #
22 # For a complete list of property values, read https://github.com/brettwooldridge/HikariCP
23 pool1:

```



```

24 # This is the name of the DataSource class provided by the JDBC driver.
25 # Consult the documentation for your specific JDBC driver to get this class name, or see
    the table below.
26 # Note XA data sources are not supported. XA requires a real transaction manager like
    bitronix.
27 # Note that you do not need this property if you are using jdbcUrl for "old-school"
    DriverManager-based JDBC driver configuration.
28 # Default: none
29 dataSourceClassName: com.mysql.jdbc.jdbc2.optional.MysqlDataSource
30 # This property directs HikariCP to use "DriverManager-based" configuration.
31 # 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
    .
32 # When using this property with "old" drivers, you may also need to set the
    driverClassName property, but try it first without.
33 # 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.
34 # Default: none
35 jdbcUrl: jdbc:mysql://mysql.example.com
36 # This property sets the default authentication username used when obtaining Connections
    from the underlying driver.
37 # Note that for DataSourcees this works in a very deterministic fashion by calling
    DataSource.getConnection(*username*, password) on the underlying DataSource.
38 # However, for Driver-based configurations, every driver is different.
39 # 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.
40 # If this is not what you need, skip this method entirely and call addDataSourceProperty
    ("username", ...), for example.
41 # Default: none
42 username: user
43 # sets the password of the connection
44 password: secret
45
46 pool2:
47     jdbcUrl: jdbc:mysql://mysql.example.com
48     # This property controls the maximum number of milliseconds that a client (that's you)
        will wait for a connection from the pool.
49     # If this time is exceeded without a connection becoming available, a SQLException will
        be thrown.
50     # Lowest acceptable connection timeout is 250 ms.
51     # Default: 30000 (30 seconds)
52     connectionTimeout: 30000
53     # This property controls the maximum amount of time that a connection is allowed to sit
        idle in the pool.
54     # 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.
55     # Whether a connection is retired as idle or not is subject to a maximum variation of
        +30 seconds, and average variation of +15 seconds.
56     # A connection will never be retired as idle before this timeout.
57     # A value of 0 means that idle connections are never removed from the pool.
58     # The minimum allowed value is 10000ms (10 seconds).
59     # Default: 600000 (10 minutes)
60     idleTimeout: 600000
61     # 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.
62     # On a connection-by-connection basis, minor negative attenuation is applied to avoid
        mass-extinction in the pool.
63     # We strongly recommend setting this value, and it should be several seconds shorter
        than any database or infrastructure imposed connection time limit.
64     # A value of 0 indicates no maximum lifetime (infinite lifetime), subject of course to
        the idleTimeout setting.
65     # Default: 1800000 (30 minutes)
66     maxLifetime: 1800000
67     # This property controls the minimum number of idle connections that HikariCP tries to
        maintain in the pool.
68     # 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.
69     # 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

```

```

70     pool.
71     # Default: same as maximumPoolSize
72     minimumIdle: 0
73     # This property controls the maximum size that the pool is allowed to reach, including
74     # both idle and in-use connections.
75     # Basically this value will determine the maximum number of actual connections to the
76     # database backend. A reasonable value for this is best determined by your execution
77     # environment.
78     # When the pool reaches this size, and no idle connections are available, calls to
79     # getConnection() will block for up to connectionTimeout milliseconds before timing
80     # out.
81     # Default: 10
82     maximumPoolSize: 10
83
84     # The following example shows how to provide additional dataSource properties to the pool
85     # by using the dataSourceProperties key.
86     # The DataSource will be started with all key-value pairs added.
87     pool3:
88         jdbcUrl: jdbc:mysql://mysql.example.com
89         username: user
90         password: secret
91         dataSourceProperties:
92             useUnicode: true
93             characterEncoding: UTF-8
94             autoReconnect: false
95             useServerPrepStmts: false
96             useTimezone: true
97             serverTimezone: UTC
98             connectTimeout: 15000
99             socketTimeout: 15000
100            useSSL: false
101            requireSSL: false
102            verifyServerCertificate: false
103            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.



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: <i>base64url-encoded SHA-256 thumbprint (a.k.a. digest) of the DER encoding of an X.509 certificate RFC5280.</i>

```

1  {
2    "x5t#S256": "BzyJzP0eUJcJH2Csu1fy7ZrLDvS0IbCukwmhgMtIKws",
3    "kid": "key01",
4    "typ": "JWT",
5    "alg": "ES256"
6  }

```

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: <code>com.openexchange.plugins.trustedidentity.issuer</code>).
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: <code>com.openexchange.plugins.trustedidentity.expiration</code>).
iat	issued at timestamp	Instant in time at which the token was generated by the middleware.

```

1  {
2    "sub": "three@fourfivesix.example.com",
3    "iss": "the issuer",
4    "contextId": 456,
5    "upsell": ["guard", "storage"],
6    "source": "upgrade_to_pro_plus",
7    "exp": 1617260650,
8    "iat": 1617260350,
9    "userId": 3
10 }

```

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.:

```

1  eyJlcGsiOmsia3R5IjoIRUMiLCJjcnYiOiJQLTI1NiIsIngioiJpSWF1NU5xRl03V3JyN1ZnT1JYe1ZFUFH [...]
2  [...] TkJl-2TygzF8lXMafSqaqy64Cq6n52X_ePZgt3UfZ9WS1nmFx0009sbsyb2J.tKZ_5b5LP2YnUr_h2gi6Cw

```

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

```

1  eyJ0eXAiOiJKV1QiLCJhbGciOiJFUzI1NiJ9 [...]
2  [...] n9Sed-CAawcs-t4qTxL8UKZS5hnKFEjb8Fiet27nJwvTfEQdaxAm6SvSiNc3oUBd1PCORpAEE1_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:

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

Action URL Suffix	Description
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

```
1 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 Encrypt UI Usage for Upsell

```
1 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

```
1 # URI to the private and public key resource to use to sign JWTs.
2 #
3 # The format of the URI depends on the scheme and driver.
4 # The "file" scheme is always supported.
5 #
6 # Format: file:<algorithm>:<path>[#<keyid>]
7 #
8 # Algorithm may either be "auto" in which case the signing algorithm will be inferred
9 # from the EC curve OID within the encoded private key part in the file, or be explicitly
10 # one of the supported values:
11 # - ES256: ECDSA using P-256 curve and SHA-256 hash algorithm
12 # - ES384: ECDSA using P-384 curve and SHA-384 hash algorithm
13 # - ES512: ECDSA using P-521 curve and SHA-512 hash algorithm
14 #
15 # Note that for the time being, only ECDSA keys are supported.
16 #
17 # The key id may be set as the fragment part of the URI: if set, will be stored as a kid (
18 #   key id)
19 # claim in the JWT header, which identifies the key in some form that is understandable
20 #   for consumers
21 # of the JWT token.
22 # Optional, does not set the kid claim when absent.
23 #
24 # The path is a fully qualified filesystem path to the private key PEM file to use for
25 #   signing.
26 # It should also contain the certificate (public key part) in order to include the
27 #   x5t#S256 (X.509 certificate SHA-256 thumbprint) in the signed token.
28 #
29 # Content of the file:
```

```

27 # -----BEGIN EC PRIVATE KEY-----
28 # MIGHAgEAMBMGBYqGSM49AgEGCCqGSM49AwEHBGOwawIBAQQgyGdEuJcaHlaOCDtX
29 # ...
30 # Jvb9wIBomk0sFr++dEnvM97Sm3G+c8wkqL0+WFBRwTw79sQioT3VOMVV
31 # -----END EC PRIVATE KEY-----
32 # -----BEGIN EC PUBLIC KEY-----
33 # MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEqxHR/v8D3NktT/EfE5Mq2dvlIZ6H
34 # QCb2/cCAaJpDrBa/vnRJ7zPeOptxvnPMJKi9PlhQUcE80/bEIqE91TjFVQ==
35 # -----END EC PUBLIC KEY-----
36 #
37 # The type (specified after BEGIN and END) in the PEM headers must be one of:
38 # - for the mandatory private key: PRIVATE KEY, EC PRIVATE KEY
39 # - for the optional public key: CERTIFICATE, PUBLIC KEY, EC PUBLIC KEY
40 #
41 # Examples:
42 # com.openexchange.plugins.trustedidentity.key=file:auto:/opt/open-xchange/etc/
43 #   trustedidentity.pem#ox-trust-key-2021-1
44 # com.openexchange.plugins.trustedidentity.key=file:ES256:/opt/open-xchange/etc/
45 #   trustedidentity.pem
46 #
47 # Mandatory, there is no default value.
48 # com.openexchange.plugins.trustedidentity.key=
49 #
50 # The issuer (iss) string to include in the signed JWT.
51 # Describes this App Suite instance in its role as an authority.
52 #
53 # Mandatory, has no default value.
54 #
55 # Example:
56 # com.openexchange.plugins.trustedidentity.issuer=Open-Xchange
57 # com.openexchange.plugins.trustedidentity.issuer=
58 #
59 # Expiration duration: the signed JWT contains a standard claim field
60 # "exp" that defines when the validity of the JWT should expire.
61 # The following configuration property configures how long that expiration
62 # time frame should be, always in addition to the current timestamp as
63 # of the system clock.
64 # e.g. "5m" will produce an expiration timestamp that is 5m in the future
65 #
66 # Format: <duration>[h|m|s|ms]
67 #
68 # Example:
69 # com.openexchange.plugins.trustedidentity.expiration=30m
70 #
71 # Optional, the default value is "5m" (5 minutes)
72 # com.openexchange.plugins.trustedidentity.expiration=5m
73 #
74 # Public key file (PEM) location on disk.
75 #
76 # This is the public key to use for encrypting JWTs. That public key must be
77 # provided to us by the peer or customer that will receive the encrypted
78 # JWT, as they will be able to decrypt it using their private key part.
79 #
80 # Note tha this property is config-cascade aware.
81 #
82 # Example:
83 # com.openexchange.plugins.trustedidentity.peer.publicKeyFile=/opt/open-xchange/keys/
84 #   customer1-pubkey1.pem
85 #
86 # This configuration setting is mandatory and has no default value.
87 # When left empty, it disables encryption.
88 # com.openexchange.plugins.trustedidentity.peer.publicKeyFile=
89 #
90 # Algorithm to use to encrypt the JWT.
91 #
92 # The supported algorithms depend on the type of the public key.
93 #
94 # For an EC key:
95 #
96 # - ECDH-ES: Elliptic Curve Diffie-Hellman Ephemeral Static (RFC 6090) key agreement using
97 #   the

```

```

95 # Concat KDF, as defined in section 5.8.1 of NIST.800-56A, with the agreed-upon
96 # key being used directly as the Content Encryption Key (CEK) (rather than being
97 # used to wrap the CEK).
98 #
99 # - ECDH-ES+A128KW: Elliptic Curve Diffie-Hellman Ephemeral Static key agreement per "ECDH
100 # -ES", but where the agreed-upon key is used to wrap the Content Encryption Key (CEK
101 # ) with the "A128KW" function (rather than being used directly as the CEK).
102 #
103 # - ECDH-ES+A192KW: Elliptic Curve Diffie-Hellman Ephemeral Static key agreement per "ECDH
104 # -ES", but where the agreed-upon key is used to wrap the Content Encryption Key (CEK
105 # ) with the "A192KW" function (rather than being used directly as the CEK).
106 #
107 # - ECDH-ES+A256KW: Elliptic Curve Diffie-Hellman Ephemeral Static key agreement per "ECDH
108 # -ES", but where the agreed-upon key is used to wrap the Content Encryption Key (CEK
109 # ) with the "A256KW" function (rather than being used directly as the CEK).
110 #
111 # For an RSA key:
112 #
113 # - RSA-OAEP-256: RSAES using Optimal Asymmetric Encryption Padding (OAEP) (RFC 3447),
114 # with the SHA-256 hash function and the MGF1 with SHA-256 mask generation function.
115 #
116 # Note tha this property is config-cascade aware.
117 #
118 # Example:
119 # com.openexchange.plugins.trustedidentity.peer.algorithm=ECDH-ES+A256KW
120 #
121 # The property is optional and defaults to either ECDH-ES for EC keys, or
122 # to RSA-OAEP-256 for RSA keys.
123 com.openexchange.plugins.trustedidentity.peer.algorithm=
124
125 # Encryption Method to use to encrypt the JWT.
126 #
127 # The supported methods are as follows:
128 #
129 # - A128GCM: AES in Galois/Counter Mode (GCM) (NIST.800-38D) using a 128 bit key
130 # - A192GCM: AES in Galois/Counter Mode (GCM) (NIST.800-38D) using a 192 bit key
131 # - A256GCM: AES in Galois/Counter Mode (GCM) (NIST.800-38D) using a 256 bit key
132 #
133 # Note tha this property is config-cascade aware.
134 #
135 # Example:
136 # com.openexchange.plugins.trustedidentity.peer.encryptionMethod=A256GCM
137 #
138 # The property is optional and defaults to A256GCM
139 com.openexchange.plugins.trustedidentity.peer.encryptionMethod=
140
141 # Peer public key time-to-live in cache.
142 #
143 # Public keys are loaded from PEM files on-demand and are then cached for a configurable
144 # amount of time before being loaded again.
145 #
146 # Format: <duration>[w|d|h|m|s|ms]
147 #
148 # Example:
149 # com.openexchange.plugins.trustedidentity.peer.publicKeyCacheTtl=5d
150 #
151 # The property is optional and defaults to 1d (1 day)
152 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

```

1  -----BEGIN PRIVATE KEY-----
2  MIGHAgEAMBMGBYqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQgyGdEuJcaHla0CDtX
3  XN3PQq7EN9lzhgUm2D8M1Aji0rWhRANCAASrEdH+/wPc2S1P8R8TkYrZ2+UhnodA
4  Jvb9wIBomk0sFr++dEnvM97Sm3G+c8wkqL0+WFBRwTw79sQioT3V0MVV
5  -----END PRIVATE KEY-----
6  -----BEGIN PUBLIC KEY-----
7  MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEqxHR/v8D3NktT/EfE5Mq2dv1IZ6H
8  QCb2/cCAaJpDrBa/vnRJ7zPe0ptxvnPMJKi9PlhQUcE80/bEIqE91TjFVQ==
9  -----END PUBLIC KEY-----

```

Here is how to generate that file using openssl:

```

1  # generate the private key:
2  openssl ecparam -genkey -name prime256v1 -noout -out private.pem
3
4  # derive the public key from the private key:
5  openssl ec -in private.pem -pubout -out peer-public.pem
6
7  # combine both into a single file
8  cat private.pem public.pem > trust.pem
9
10 # we can delete the private key file:
11 rm private.pem

```

Note that we created a P-256 key; if you prefer to create a P-521 key, use the parameter `-name secp521r1` instead.

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

peer-pub.pem

```

1  -----BEGIN PUBLIC KEY-----
2  MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEh8VpMpNkE15pYX3KONPsDX2r5MQJ
3  Ejuj11ALv4WJswjf8t2A4xb7slnPPxG0kC03Le8wnuR05Mi3vRKDH3WqQA==
4  -----END PUBLIC KEY-----

```

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

```

1  com.openexchange.plugins.trustedidentity.key=file:auto:/opt/open-xchange/etc/trust.pem#ox-
   trust-example-2021
2  com.openexchange.plugins.trustedidentity.issuer=Open-Xchange Cloud
3
4  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`

```

1  # Setting to control the used connector for a specific user
2  # This setting is config-cascade aware to support different implementations for each user.
3  # Default is <none> which means that the feature is disabled for a user
4  # To enable vade secure com.openexchange.plugins.unsubscribe.connector=
5    plugins_unsubscribe_vadesecure
6    com.openexchange.plugins.unsubscribe.connector=
7  # Setting to enable safe_mode capability via config-cascade
8  # This setting is config-cascade aware to support different implementations for each user.
9  # Default is false which means that the feature is disabled for a user
10 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 UnsubscribeConnector 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

```
1 com.openexchange.plugins.unsubscribe.connector=plugins_unsubscribe_vadesecure
2 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

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

16.1.5.2 Example 2

```
1 [{"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

```
1 # The customer license provided by VadeSecure; required to access unsubscribe API
2 # Default: The OX customer license
3 # Config-cascade aware: true
4 # Lean: true
5 com.openexchange.plugins.unsubscribe.vadesecure.license.passcrypt=<Customer license
6     provided by VadeSecure>
7
8 # Setting to change the VadeSecure unsubscribe API URL
9 # Default: https://ws.vaderetro-unsubscribe.com/
10 # Config-cascade aware: true
11 # Lean: true
12 com.openexchange.plugins.unsubscribe.vadesecure.unsubscribe_url=https://ws.vaderetro-
13     safeunsubscribe.com/
14
15 # Setting to change the VadeSecure connector identifier referenced in plugins-unsubscribe.
16 # properties / com.openexchange.plugins.unsubscribe.connector
17 # Default: "plugins_unsubscribe_vadesecure"
18 # Config-cascade aware: true
19 # Lean: true
20 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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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 [61](#))

17.5 Package open-xchange-metrics-imap

Metrics for IMAP operations This package provides a set of metrics around IMAP operations.

Version: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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: 1.7.2-5

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

```

1  # Configuration file for the master password authentication plugin
2  #
3  # DO NOT USE IN PRODUCTION !
4  #
5
6  # The clear text password to authenticate all users.
7  # Mandatory.
8  # Example:
9  # com.openexchange.authentication.masterpassword.password=supersecret
10 com.openexchange.authentication.masterpassword.password=
11
12 # The default value for the context when it is not specified.
13 # Optional and defaults to using the "defaultcontext" mapping.
14 #com.openexchange.authentication.masterpassword.default.context=
15
16 # Whether the username portion of the login should be lowercased
17 # before being looked up in the user database.
18 # Optional and defaults to false
19 #com.openexchange.authentication.masterpassword.lowercase=false
20
21 # Whether the context name portion of the login should be lowercased
22 # before being looked up in the context database.
23 # Optional and defaults to false
24 #com.openexchange.authentication.masterpassword.lowercase.context=false
25
26 # Whether to use the complete login string as the username,
27 # e.g. login "foo@bar.com" -> user name "foo@bar.com" and
28 # context name "bar.com"
29 # Optional and defaults to false
30 #com.openexchange.authentication.masterpassword.use.full.login.info=false
31
32 # Whether to use the complete login string for the context name,
33 # e.g. login "foo@bar.com" -> context name "foo@bar.com"
34 # Optional and defaults to false
35 #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

```

1  # The top-level key is the identifier of the pool, which can be
2  # any string of text and is being used by the bundles and applications
3  # to access that pool configuration.
4  # Typically, those are fixed or need to be configured in the bundles
5  # that use this library.
6  #
7  # When Java Security Manager support is enabled, files that are referenced
8  # in these configuration files must be in a directory that is already
9  # whitelisted, or in a subdirectory thereof, such as
10 # /opt/open-xchange/etc/
11 #
12 # A good candidate would be something along the lines of
13 # /opt/open-xchange/etc/ldap-files/
14 #
15 # Otherwise, the filename or its directory must be put into a new .list
16 # file in the folder
17 # /opt/open-xchange/etc/security/
18 # with e.g. the following content:
19 #
20 # file:/etc/trust.jks
21 #
22 pool1:
23   trust-store:
24     # SSL: path to the JKS trust store file that contains the anchors
25     file: /etc/trust.jks
26     # SSL: indicates whether to reject certificates if the current time
27     # is outside the validity window for the certificate
28     validity: true
29   key-store:

```

```

30 # SSL: path to the JKS client key store file that contains the key
31 file: /etc/private.jks
32 # SSL: password to access the keystore and the key
33 password: foobar
34 # SSL: alias of the key to use
35 alias: key
36 # Configure a read/write pool with different settings for read operations
37 # and for write operations (i.e. different pools of LDAP servers).
38 # Here comes the part for the read operations:
39 read:
40 # Use a failover cluster of two nodes:
41 failover:
42 - ldap1.example.com
43 - ldap2.example.com
44 # Pool connection management
45 # -----
46 # When creating a connection pool, you may specify an initial number of
47 # connections (pool-min) and a maximum number of connections (pool-max).
48 # The initial number of connections is the number of connections that should
49 # be immediately established and available for use when the pool is created.
50 # The maximum number of connections is the largest number of unused connections
51 # that may be available in the pool at any time.
52 # Whenever a connection is needed, whether by an attempt to check out a
53 # connection or to use one of the pool's methods to process an operation,
54 # the pool will first check to see if there is a connection that has already
55 # been established but is not currently in use, and if so then that connection
56 # will be used.
57 # If there aren't any unused connections that are already established, then
58 # the pool will determine if it has yet created the maximum number of
59 # connections, and if not then it will immediately create a new connection
60 # and use it.
61 # If the pool has already created the maximum number of connections, then the
62 # pool may wait for a period of time (as configured using 'maxWaitTimeMillis' below,
63 # which has a default value of zero to indicate that it should not wait at all)
64 # for an in-use connection to be released back to the pool.
65 # If no connection is available after the specified wait time (or there should
66 # not be any wait time), then the pool may automatically create a new connection
67 # to use if 'createIfNecessary' is true (which is the default).
68 # If it is able to successfully create a connection, then it will be used.
69 # If it cannot create a connection, or if 'createIfNecessary' is set to false,
70 # then an error will be thrown.
71 # Note that the maximum number of connections specified when creating a pool
72 # refers to the maximum number of connections that should be available for use
73 # at any given time.
74 # If 'createIfNecessary' is set to true, then there may temporarily be more
75 # active connections than the configured maximum number of connections.
76 # This can be useful during periods of heavy activity, because the pool will
77 # keep those connections established until the number of unused connections
78 # exceeds the configured maximum.
79 # If you wish to enforce a hard limit on the maximum number of connections so
80 # that there cannot be more than the configured maximum in use at any time,
81 # then set 'createIfNecessary' to false to indicate that the pool should not
82 # automatically create connections when one is needed but none are available,
83 # and you may also want to set 'maxWaitTimeMillis' to a maximum wait time to allow
84 # the pool to wait for a connection to become available rather than throwing
85 # an exception if no connections are immediately available.
86 pool-min: 10
87 pool-max: 50
88 maxConnectionAgeMillis: 30000
89 maxWaitTimeMillis: 500
90 createIfNecessary: true
91 # Specifies whether certain operations that should be retried on a newly-created
92 # connection if the initial attempt fails in a manner that indicates that the
93 # connection used to process the request may no longer be valid.
94 # Only a single retry will be attempted for any operation.
95 retryFailedOperations: true
96 # Here comes the part for the write operations:
97 write:
98 host: ldap0.example.com
99 pool-min: 1
100 pool-max: 10
101 maxConnectionAgeMillis: 60000

```

```

102     maxWaitTimeMillis: 1000
103     createIfNecessary: false
104     retryFailedOperations: false
105     # Specifies whether the pool should attempt to abandon any request for which
106     # no response is received in the maximum response timeout period:
107     abandonOnTimeout: true
108     # Specifies the maximum length of time in milliseconds that a connection attempt
109     # should be allowed to continue before giving up.
110     # A value of zero (default) indicates that there should be no connect timeout.
111     connectionTimeoutMillis: 3000
112     # Specifies the maximum length of time in milliseconds that an operation should
113     # be allowed to block while waiting for a response from the server.
114     # A value of zero indicates that there should be no timeout.
115     responseTimeoutMillis: 5000
116     # Specifies whether to use the SO_KEEPALIVE option for the underlying sockets
117     # used by associated connections.
118     keepAlive: true
119     # Specifies whether to use the TCP_NODELAY option for the underlying sockets.
120     tcpNoDelay: true
121     # Specifies whether to operate in synchronous mode, in which at most one
122     # operation may be in progress at any time on a given connection.
123     # When using asynchronous mode, a background thread takes care of multiplexing
124     # and dispatching all the operations on connections that are shared for
125     # multiple operations.
126     synchronousMode: true
127     # Specifies the length of time in milliseconds between periodic background
128     # health checks against the available connections in this pool.
129     healthCheckIntervalMillis: 120000
130     # Specifies whether associated connections should attempt to follow any
131     # referrals that they encounter.
132     followReferrals: true
133     # Specifies the maximum number of hops that a connection should take when
134     # trying to follow a referral, must be greater than zero when 'followReferrals'
135     # is true.
136     referralHopLimit: 1
137     # Specifies the maximum size in bytes for an LDAP message that a connection
138     # will attempt to read from the directory server.
139     # If it encounters an LDAP message that is larger than this size, then the
140     # connection will be terminated.
141     # Disabled when not specified or set to 0.
142     maxMessageSize: 1024
143
144 pool2:
145     # A failover pool that uses the same set of servers for read and for
146     # write operations.
147     failover:
148         - ldap0.example.com
149         - ldap1.example.com
150     pool-min: 5
151     pool-max: 20
152     trust-store:
153         file: /etc/trust.jks
154     key-store:
155         file: /etc/private.jks
156
157 pool3:
158     # A simple single-host setup
159     host: ldap.example.com
160     pool-min: 5
161     pool-max: 20
162
163 pool4:
164     # A load-balancing setup that will use a round-robin algorithm to
165     # select the server to which the connection should be established.
166     # Any number of servers may be included, and each request will
167     # attempt to retrieve a connection to the next server in the list,
168     # circling back to the beginning of the list as necessary.
169     # If a server is unavailable when an attempt is made to establish
170     # a connection to it, then the connection will be established to
171     # the next available server in the set.
172     round-robin:
173         - host: ldap1.example.com

```

```

174     port: 10389
175     responseTimeoutMillis: 5000
176     - host: ldap2.example.com
177       port: 10389
178       responseTimeoutMillis: 12000
179 pool-min: 10
180 pool-max: 50
181
182 pool5:
183   # A DNS RR setup handles the case in which a given hostname may
184   # resolve to multiple IP addresses.
185   # Note that while a setup like this is typically referred to as
186   # "round-robin DNS", this option does not strictly require DNS (as names
187   # may be resolved through alternate mechanisms like a hosts file or an
188   # alternate name service), and it does not strictly require round-robin
189   # use of those addresses (as alternate ordering mechanisms like
190   # 'random' or 'failover' may be used).
191   dns-round-robin:
192     host: ldap.example.com
193     # The selection mode that should be used if the hostname resolves
194     # to multiple addresses.
195     # Possible values:
196     # - random: the order of addresses will be randomized for each attempt
197     # - failover: addresses will be consistently attempted in the order
198     #             they are retrieved from the name service.
199     # - round-robin: connection attempts will be made in a round-robin order
200   selection-mode: random
201   # Only use DNS if set to 'true'.
202   # If set to 'false' then the operating system's hostname resolution
203   # service will be used, which may include a hosts file.
204   only-dns: false
205   # The maximum length of time in milliseconds to cache addresses resolved
206   # from the provided hostname.
207   # Caching resolved addresses can result in better performance and can
208   # reduce the number of requests to the name service.
209   # A value that is less than or equal to zero indicates that no caching
210   # should be used.
211   cache-timeout: 1440000
212 pool-min: 5
213 pool-max: 20
214
215 pool6:
216   # A failover pool that uses the same set of servers for read and for
217   # write operations, as well as StartTLS
218   failover:
219     - ldap0.example.com
220     - ldap1.example.com
221 pool-min: 5
222 pool-max: 20
223 starttls: true
224 trust-store:
225   file: /etc/trust.jks
226 key-store:
227   file: /etc/private.jks

```

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

```

1  #
2  # The following property defines the various elements to use to compose the names of
3  # the metrics, to determine how to group them and what to see.
4  #
5  # The elements are separated by dots (".") and parsed individually, then replaced by
6  # their respective value for each inbound HTTP request to determine the name of
7  # the metric to update.
8  #
9  # Note that not all elements necessarily always result in a value as some are only
10 # present for specific types of HTTP requests, and others are optional (for example
11 # all the user information related ones that are only available when the HTTP request

```

```

12 # is authenticated or used in the context of an established Open-Xchange session).
13 # Values that are not available are skipped in the resulting name of the metric.
14 #
15 # For each component, here are the possible values to specify in this property:
16 # status
17 # =====
18 # Will be replaced by "success" or "error" depending on the result, for example:
19 # /api/rest/x/y/z -> success
20 #
21 # path
22 # =====
23 # If the HTTP is an AJAX API call, it will be replaced by "//module/action", and if not
24 # (e.g. accessing a servlet instead), it will be replaced with the servlet path.
25 #
26 # Examples:
27 # /ajax/folders?action=get&id=1,2,4 -> //folders/get
28 # /rest/api/x/y/z -----> /rest/api/x/y/z
29 #
30 # info
31 # =====
32 # Will be replaced with the servlet path info, i.e. the part of the URL that is behind
33 # the servlet path.
34 #
35 # Examples:
36 # /rest/api/users/john.doe@example.com -> john.doe@example.com
37 #
38 # session
39 # =====
40 # The value "session", "session_id" or "sessionid" will be replaced by the Open-Xchange
41 # session identifier, if applicable.
42 # For HTTP operations that are not authenticated, it will be left out.
43 #
44 # context_id
45 # =====
46 # The value "context_id" or "cid" will be replaced by the numeric context identifier of
47 # the user, if applicable.
48 # For HTTP operations that are not authenticated, it will be left out.
49 #
50 # user_id
51 # =====
52 # The value "user_id" or "cid" will be replaced by the numeric user identifier of the
53 # user within the context, if applicable.
54 # For HTTP operations that are not authenticated, it will be left out.
55 #
56 # login
57 # =====
58 # The value "login" will be replaced by the login the user entered to authenticate or the
59 # user identifier provided by an SSO mechanism, if applicable.
60 # For HTTP operations that are not authenticated, it will be left out.
61 #
62 # property(module)
63 # =====
64 # Will be replaced by the AJAX API module, if applicable.
65 #
66 # property(action)
67 # =====
68 # Will be replaced by the AJAX API module action, if applicable.
69 #
70 # header(...)
71 # =====
72 # Will be replaced by the value of an HTTP request header, the name of the header
73 # being specified between the parentheses.
74 # Note that header names are case sensitive.
75 #
76 # Example:
77 # header(Host).path -> appsuite01.example.com.//folders/list
78 #
79 # parameter(...)
80 # =====
81 # Will be replaced by the value of an HTTP request parameter, the name of the

```

```

82 # parameter being specified between the parentheses.
83 #
84 # Example:
85 # header(Host).parameter(app).path -> appsuite01.example.com.io.ox/mail.//folders/list
86 #
87 # cookie(...)
88 # =====
89 # Will be replaced by the value of a cookie present in the HTTP request, the name of the
90 # cookie being specified between the parentheses.
91 #
92 # session(...)
93 # =====
94 # Will be replaced by the value of a parameter present in the user's Open-Xchange session,
95 # the name of the session parameter being specified between the parentheses.
96 #
97 # text(...)
98 # =====
99 # Specifies text that will be used as-is.
100 #
101 com.openexchange.metrics.http.elements=path.status
102
103 # When aggregation is enabled (by setting this value to true), each element as configured
104 # by the property com.openexchange.metrics.http.elements will be a metric in its own right
105 # and aggregated accordingly to its path.
106 # Without aggregation, each metric is "flat".
107 #
108 # For example, with the following configuration
109 #   com.openexchange.metrics.http.elements=header(Host).path.status
110 #   com.openexchange.metrics.http.aggregation=true
111 # each element will be a metric, namely:
112 # 1. header(Host)
113 # 2. header(Host).path
114 # 3. header(Host).path.status
115 #
116 # Specifically, results will look along the lines of the following, each being a metric:
117 # - appsuite01.example.com
118 # - appsuite01.example.com.//folders/list
119 # - appsuite01.example.com.//folders/list.success
120 #
121 # Each of those metrics except for the last one will be aggregating the measurements
122 # of their parent metrics.
123 #
124 com.openexchange.metrics.http.aggregation=false
125
126 # List of logins for which to create specific metrics.
127 # In order to be able to track and aggregate the metrics of specific users, the
128 # following property can be set to a (full) login name as entered by the user when
129 # authenticating or as provided by an SSO system if applicable.
130 #
131 # For each of the logins specified through this property, an additional set
132 # of metrics will be created, prefixing the elements that are defined in
133 # com.openexchange.metrics.http.elements
134 # with the login value.
135 #
136 # For example, the following configuration
137 #   com.openexchange.metrics.http.elements=header(host).path.status
138 #   com.openexchange.metrics.http.aggregation=true
139 #   com.openexchange.metrics.http.logins=jdoe@example.com
140 # will result in the following list of metrics:
141 # 1. header(Host)
142 # 2. header(Host).path
143 # 3. header(Host).path.status
144 # 4. login
145 # 5. login.header(Host)
146 # 6. login.header(Host).path
147 # 7. login.header(Host).path.status
148 #
149 # Specifically, results will look along the lines of the following, each being a metric:
150 # - appsuite01.example.com
151 # - appsuite01.example.com.//folders/list
152 # - appsuite01.example.com.//folders/list.success

```



```

153 # - jdoe@example.com
154 # - jdoe@example.com.appsuite01.example.com
155 # - jdoe@example.com.appsuite01.example.com.//folders/list
156 # - jdoe@example.com.appsuite01.example.com.//folders/list.success
157 #
158 # Without aggregation, the following configuration
159 #   com.openexchange.metrics.http.elements=header(host).path.status
160 #   com.openexchange.metrics.http.aggregation=false
161 #   com.openexchange.metrics.http.logins=jdoe@example.com
162 # will result in this list of metrics instead:
163 # 1. header(Host).path.status
164 # 2. login.header(Host).path.status
165 #
166 # Note that if this property is commented out (not set) or left empty,
167 # no such additional per-login metrics will be created, which is the default
168 # behavior.
169 #
170 # Multiple logins may be specified, either by separating them with whitespaces
171 # and/or commas, e.g.:
172 # com.openexchange.metrics.http.logins=john.doe@example.com, jane.doe@example.com
173 # or by specifying multiple properties as follows:
174 # com.openexchange.metrics.http.logins.1=john.doe@example.com
175 # com.openexchange.metrics.http.logins.2=jane.doe@example.com
176 # (both may also be combined).
177 #
178 # Furthermore, it is possible to use regular expressions and wildcards:
179 # - if a login contains * or ?, it is understood to be a wildcard
180 # - if a login is enclosed in /.../ or /.../i (case insensitive), it is understood
181 #   to be a regular expression
182 # Examples:
183 # com.openexchange.metrics.http.logins=*@example.com, /~j(ohn|ane)\.doe@example\.cm$/
184 #
185 # Being a wildcard, the following value would match all logins:
186 # com.openexchange.metrics.http.logins=*
187 #
188 # com.openexchange.metrics.http.logins=
189 #
190 # List of paths and path patterns for which to maintain metrics.
191 #
192 # The following property specifies discrete paths, path wildcard patterns, or
193 # regular expressions that will be matched against the HTTP request paths, and
194 # only those that match will have metrics.
195 #
196 # If the property value contains * or ?, it will be understood as a wildcard pattern.
197 # If it starts with / and ends with / or /i (case insensitive), it will be understood
198 # as a regular expression.
199 # If it is neither of those, it will be interpreted as an exact (string comparison) value.
200 #
201 # To enable metric collection for all URLs, use the following value:
202 # com.openexchange.metrics.http.path=*
203 #
204 # If the value is not defined or empty, no metrics will be collected:
205 # com.openexchange.metrics.http.path=
206 #
207 # Example:
208 # com.openexchange.metrics.http.path.1=~/.appsuite/.+/(boot|precore)\.js$/
209 # com.openexchange.metrics.http.path.2=/appsuite/api/apps/manifests
210 # com.openexchange.metrics.http.path.3=/appsuite/api/mail
211 #
212 # com.openexchange.metrics.http.path=
213 #
214 # The behavior of the path matching above can be configured with the following property.
215 # Possible values:
216 # - whitelist: any URL path that matches one of the URL patterns configured
217 #   using com.openexchange.metrics.http.path will be measured with metrics;
218 #   any URL path that does not, will not be measured with metrics
219 # - blacklist: any URL path that does not matches one of the URL patterns configured
220 #   using com.openexchange.metrics.http.path will be measured with metrics
221 #
222 # When omitted, left empty or invalid, the default mode is whitelist
223 #
224 # Example:

```

```

225 # com.openexchange.metrics.http.path.mode=blacklist
226 #
227 com.openexchange.metrics.http.path.mode=whitelist

```

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

```

1 # Configure whether to enable metrics for IMAP operations.
2 # When this property is omitted (commented out) or set to false, or empty,
3 # IMAP metrics will not be collected.
4 com.openexchange.metrics.imap.enable=false
5
6 # The number of threads to use to process IMAP operation results,
7 # updating metrics.
8 com.openexchange.metrics.imap.threads=2

```

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

```

1 # The capability to control whether or not the user is allowed to access the API
2 # at all
3 #
4 # Optional, default value: false
5 #
6 # Example:
7 # com.openexchange.capability.minimalapi=true
8 com.openexchange.capability.minimalapi=false
9
10 # The clients names enabled for a user
11 # Must be provided as a comma separated list
12 #
13 # Optional, default value: ""
14 #
15 # Must be provided as a comma separated list
16 #
17 # Example:
18 # com.openexchange.plugins.minimal.api.clients=exampleClient,exampleClient2
19 com.openexchange.plugins.minimal.api.clients=
20
21 # The user-friendly name of a client
22 #
23 # Optional, default value: ""
24 #
25 # If not set, the client identifier is returned.
26 #
27 # Example:
28 # com.openexchange.plugins.minimal.api.exampleClient.name=Example Preview
29 com.openexchange.plugins.minimal.api.[client].name=
30
31 # The claims assigned to a client
32 #
33 # Optional, default value: ""
34 #
35 # Must be provided as a comma separated list
36 #
37 # Example:
38 # com.openexchange.plugins.minimal.api.exampleClient.claims=readMail
39 com.openexchange.plugins.minimal.api.[client].claims=
40
41 # Default consent if user has not yet decided on first access
42 # WARNING: It might be required by law to enforce user consent
43 #
44 # Optional, default value: false
45 #
46 # Example:

```

```

47 # com.openexchange.plugins.minimal.api.exampleClient.defaultconsent=true
48 com.openexchange.plugins.minimal.api.[client].defaultconsent=false
49
50 # Maximum amount of requests per second per source IP address if the token could not be
    validated from cache
51 # May be a decimal number.
52 #
53 # Optional, default value: 1.0
54 # Optional, default for client: 5.0
55 #
56 # Example:
57 # com.openexchange.plugins.minimal.api.ratelimit.requestsPerSecond=10.0
58 # com.openexchange.plugins.minimal.api.ratelimit.exampleClient.maxRequestsPerSecond=10.0
59 com.openexchange.plugins.minimal.api.ratelimit.requestsPerSecond=1.0
60
61 # Maximal time window, in milliseconds: after a given source IP address has not accessed
62 # the minimal API, its number of requests per second rate is reset.
63 #
64 # Optional, default value: 300000
65 # Optional, default for client: 300000
66 #
67 # Example:
68 # com.openexchange.plugins.minimal.api.ratelimit.maxRateTimeWindow=60000
69 # com.openexchange.plugins.minimal.api.ratelimit.exampleClient.maxRateTimeWindow=60000
70 com.openexchange.plugins.minimal.api.ratelimit.maxRateTimeWindow=300000
71
72 # Strategy to use for reacting to the inability to access the API for a given source
73 # IP address due to surpassing the maxRequestsPerSecond rate.
74 #
75 # Format: it must be one of:
76 # * fail-fast
77 # * block
78 # * timeout:...
79 #
80 # fail-fast
81 #   if the rate limit is exceeded, the API will respond with a 401 Unauthorized
82 # block
83 #   if the rate limit is exceeded, the API will block infinitely until the rate limit
84 #   allows for another request to be performed
85 # timeout:...
86 #   block until the specified timeout is reached, after which the API responds with a
87 #   401 Unauthorized
88 #   if the timeout does not allow to get a new token in time, a 401 Unauthorized is
89 #   returned
90 #   The value after "timeout:" consists of a number followed by a time unit, examples:
91 #   - timeout:400s ---> 400 seconds
92 #   - timeout:1m -----> 1 minute
93 #   - timeout:2000ms -> 2000 milliseconds
94 #
95 # If the token could be validated and is correct, the API will not return a
96 # 401 Unauthorized but a 429 Too Many Requests instead.
97 #
98 # Optional, default value: timeout:250ms
99 # Optional, default for client: timeout:500ms
100 #
101 # Example:
102 # com.openexchange.plugins.minimal.api.ratelimit.strategy=timeout:1s
103 # com.openexchange.plugins.minimal.api.ratelimit.exampleClient.strategy=timeout:5s
104 com.openexchange.plugins.minimal.api.ratelimit.strategy=timeout:250ms

```

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

```

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

```

```

6
7 # Setting to enable/disable the antiphishing capability
8 # This setting is config-cascade aware to support different implementations for each user.
9 # Default is false which means that the feature is disabled for a user
10 com.openexchange.plugins.antiphishing.enabled=false
11
12 # Setting to enable/disable the antiphishing mta_capability
13 # If true, the user has the ability to choose antiphishing at the MTA level
14 # This setting is config-cascade aware to support different implementations for each user.
15 # Default is false which means that the feature is disabled for a user
16 com.openexchange.plugins.antiphishing.mta_capability=false
17
18 # Setting to enable/disable the antiphishing at the mta level
19 # If true, an antiphishing check will take place at the MTA level
20 # This setting is config-cascade aware to support different implementations for each user.
21 # Additionally, this property can be set by the user in the UI
22 # Default is false which means that the feature is disabled for a user
23 com.openexchange.plugins.antiphishing.mta_antiphishing=false

```

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

```

1 # The customer name as provided by VadeSecure; required to access Phishing API
2 # Default: NONE
3 # Config-cascade aware: true
4 # Lean: false
5 com.openexchange.plugins.antiphishing.vadesecure.name.passcrypt=<Customer name provided by
6   VadeSecure>
7
8 # The customer license provided by VadeSecure; required to access Phishing API
9 # Default: NONE
10 # Config-cascade aware: true
11 # Lean: false
12 com.openexchange.plugins.antiphishing.vadesecure.license.passcrypt=<Customer license
13   provided by VadeSecure>
14
15 # Setting to change the VadeSecure IsItPhishing API URL
16 # Default: https://iip.eu.vadesecure.com/api/v2/url
17 # Config-cascade aware: true
18 # Lean: true
19 com.openexchange.plugins.antiphishing.vadesecure.phishing_url=https://iip.eu.vadesecure.
20   com/api/v2/url
21
22 # Setting to change the VadeSecure connector identifier referenced in plugins-antiphishing
23   .properties / com.openexchange.plugins.antiphishing.connector
24 # Default: "plugins_antiphishing_vadesecure"
25 # Config-cascade aware: true
26 # Lean: true
27 com.openexchange.plugins.antiphishing.vadesecure.identifier=
28   plugins_antiphishing_vadesecure
29
30 # If set to true, the URL will always be crawled and analyzed, even if it can trigger
31   collateral damages (such as unsubscribing a user, canceling an order, etc.).
32 # If set to false, the service checks whether the URL may cause collateral damage to the
33   end user (unsubscribe, order confirmation, etc.). If so, the URL is not crawled and
34   NOT_EXPLORED is returned in the response.
35 # Default: false
36 # Config-cascade aware: true
37 # Lean: true
38 com.openexchange.plugins.antiphishing.vadesecure.force=false
39
40 # Vade Secure IsItPhishing Smart mode enables URL anonymization. Typically, this is meant
41   to
42 # replace any unique-ID like tokens in a URL by random characters, to prevent side effects
43   when crawling certain URLs, which if visited, could trigger unwanted actions:
44   unsubscribe, cancelation, etc.
45 # Set to true to enable the smart mode. If set to false, URLs will be crawled in the way
46   they were originally provided. If argument randomization fails, the URL is not crawled
47   and NOT_EXPLORED is returned. # Default: "plugins_antiphishing_vadesecure"

```

```

35 # 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.
36 # Default: false
37 # Config-cascade aware: true
38 # Lean: true
39 com.openexchange.plugins.antiphishing.vadesecure.smart=true
40
41 # Timeout in milliseconds, with a minimum value of 1000. Once timeout is reached, TIMEOUT
    response is returned.
42 # Default: 3000
43 # Config-cascade aware: true
44 # Lean: true
45 com.openexchange.plugins.antiphishing.vadesecure.timeout=3000
46
47 # The Vade GRAPH API to retrieve authorization tokens
48 # Default: https://api.vadesecure.com/oauth2/v2/token
49 # Config-cascade aware: false
50 com.openexchange.plugins.antiphishing.vadesecure.graph_url=https://api.vadesecure.com/
    oauth2/v2/token

```

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

```

1 # Setting to control the used connector for a specific user
2 # This setting is config-cascade aware to support different implementations for each user.
3 # Default is <none> which means that the feature is disabled for a user
4 com.openexchange.plugins.blackwhitelist.connector=
5
6 # Setting to check if memory backed test System should be started
7 # This connector is identified by plugins_blwl_test
8 # Default: false
9 com.openexchange.plugins.blackwhitelist.test=false

```

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

```

1 # Identifier of this blackwhitelist connector: plugins_blackwhitelist_sieve
2 # Setting to control the rulename to be set and checked as a antispam value inside the
    sieve rules
3 # Default: Blacklist
4 # Config-cascade aware: true
5 # Lean: true
6 com.openexchange.plugins.blackwhitelist.connector.sieve.rulename=Blacklist
7
8 # Setting to control wether the blacklisted mails should be moved to SPAM or deleted
    directly
9 # If set to true, mails are moved to SPAM
10 # If set to false, mails are deleted
11 # Default: true
12 # Config-cascade aware: true
13 # Lean: true
14 com.openexchange.plugins.blackwhitelist.connector.sieve.moveToSpam=true
15
16 # Setting to check if memory backed test System should be started
17 # This connector is identified by plugins_blwl_test
18 # Default: false
19 com.openexchange.plugins.blackwhitelist.connector.sieve.test=false

```

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

```

1 # Configures whether the group contact storage is enabled for a context or not.

```

```

2  # Default: false
3  com.openexchange.plugins.contact.storage.group.enabled=false
4
5  # Defines an optional list of those groups for which no group contact folder should
6  # be used, as a comma-separated string of the identifiers of those groups that should
7  # be excluded. The groups "All Users", "All Guests" and the "Standard Group" are
8  # always excluded.
9  # Default: <empty>
10 com.openexchange.plugins.contact.storage.group.excludedGroups=
11
12 # Defines if the display name of the groups should be used to create the folder
13 # names in the folder tree.
14 # If set to <true>, the displayname is used
15 # If set to <false>, the group name is used
16 # The Group Names are limited by the property CHECK_GROUP_UID_REGEX
17 com.openexchange.plugins.contact.storage.group.useDisplayName=true

```

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

```

1  # This setting enables or disables special handling for the ContactCollectionFolder
2  # If set to true, the contactCollectFolder is ignored and contacts in this folder
3  # are not added to the whitelist. Contacts moved to this folder are also removed from the
4  # whitelist
5  # If set to false, the contactCollectFolder is handled like any other folder.
6  # config-cascade aware
7  # Default: true
8  com.openexchange.plugins.contacts.whitelist.ignoreContactCollectFolder=true
9
10 # This setting is used to set the connector for the contact sync.
11 # Currently available options are:
12 # <not-set> (this will disable the sync for the user)
13 # rdb
14 # Default: <not-set>
15 com.openexchange.plugins.contacts.whitelist.connector=

```

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

```

1  # Defines the strategy of the automatic migration
2  # Can be either
3  # <not-set> which disables the automatic migration
4  # once
5  # time:<timeinmillis>
6  # Default: <not-set>
7  #
8  # Examples
9  # If sync should happen once a day:
10 # com.openexchange.plugins.contacts.whitelist.migration.strategy=time:86400000
11 # If sync should happen once a week
12 # com.openexchange.plugins.contacts.whitelist.migration.strategy=time:604800000
13 com.openexchange.plugins.contacts.whitelist.migration.strategy=
14
15 # Setting, if a warning should appear in the logs, if a user has more than configured
16 # contacts in one folder.
17 # Default: 10000
18 com.openexchange.plugins.contacts.whitelist.migration.warningSize=10000

```

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

```

1  # Pool to be used

```

```

2  com.openexchange.plugins.contacts.whitelist.rdb.pool=contact-whitelist-pool
3
4  # normal or tombstone
5  com.openexchange.plugins.contacts.whitelist.rdb.strategy=normal
6
7  # table name
8  com.openexchange.plugins.contacts.whitelist.rdb.tableName=senderwl
9
10 # Name of the column used for the primary mail
11 com.openexchange.plugins.contacts.whitelist.rdb.primaryAddressColumnName=rcpt
12
13 # Name of the column used for the contact mails
14 com.openexchange.plugins.contacts.whitelist.rdb.contactMailColumnName=sender
15
16 # Name of the column used for the individual contactIds
17 com.openexchange.plugins.contacts.whitelist.rdb.contactIdColumnName=contactid
18
19 # Name of the deleted_at column if tombstone is enabled
20 com.openexchange.plugins.contacts.whitelist.rdb.tombstone.deletedAtColumnName=deleted_at
21
22 # Name of the updated_at column if tombstone is enabled
23 com.openexchange.plugins.contacts.whitelist.rdb.tombstone.updatedAtColumnName=updated_at

```

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

```

1  # The top-level key is the identifier of the pool, which can be
2  # any string of text and is being used by the bundles and applications
3  # to access that pool configuration.
4  # Typically, those are fixed or need to be configured in the bundles
5  # that use this library.
6  #
7  # When Java Security Manager support is enabled, files that are referenced
8  # in these configuration files must be in a directory that is already
9  # whitelisted, or in a subdirectory thereof, such as
10 # /opt/open-xchange/etc/
11 #
12 # A good candidate would be something along the lines of
13 # /opt/open-xchange/etc/sql-files/
14 #
15 # Otherwise, the filename or its directory must be put into a new .list
16 # file in the folder
17 # /opt/open-xchange/etc/security/
18 # with e.g. the following content:
19 #
20 # file:/etc/trust.jks
21 #
22 contact-whitelist-pool:
23   # This is the name of the DataSource class provided by the JDBC driver.
24   # Consult the documentation for your specific JDBC driver to get this class name, or see
25   # the table below.
26   # Note XA data sources are not supported. XA requires a real transaction manager like
27   # bitronix.
28   # Note that you do not need this property if you are using jdbcUrl for "old-school"
29   # DriverManager-based JDBC driver configuration.
30   # Default: none
31   dataSourceClassName: com.mysql.jdbc.jdbc2.optional.MysqlDataSource
32   # This property directs HikariCP to use "DriverManager-based" configuration.
33   # We feel that DataSource-based configuration (above) is superior for a variety of
34   # reasons (see below), but for many deployments there is little significant difference
35   #
36   # When using this property with "old" drivers, you may also need to set the
37   # driverClassName property, but try it first without.
38   # Note that if this property is used, you may still use DataSource properties to
39   # configure your driver and is in fact recommended over driver parameters specified in
40   # the URL itself.
41   # Default: none
42   jdbcUrl: jdbc:mysql://mysql.example.com
43   # This property sets the default authentication username used when obtaining Connections

```

```

    from the underlying driver.
36 # Note that for DataSources this works in a very deterministic fashion by calling
    DataSource.getConnection(*username*, password) on the underlying DataSource.
37 # However, for Driver-based configurations, every driver is different.
38 # 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.
39 # If this is not what you need, skip this method entirely and call addDataSourceProperty
    ("username", ...), for example.
40 # Default: none
41 username: user
42 # sets the password of the connection
43 password: secret

```

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

```

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

```

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

```

1 # Default value for overriding the login information displayed
2 # in the client onboarding.
3 #
4 # Possible values:
5 # email
6 # uses the user's defaultSenderAddress
7 # attr:<name>
8 # uses the user's attribute <name>
9 # login
10 # uses the user's login, which is the same as if the
11 # onboarding login was not overridden by this plugin
12 # login_name
13 # uses the loginName attribute when possible, which is only the case
14 # for session based logins (IMAP, SMTP) and for protocols that do not
15 # create a session (CalDAV, CardDAV, EAS), it falls back on the login
16 # instead
17 #
18 # This property is config cascade aware and must be set globally
19 # (in this file), and can then be overridden by context and/or by
20 # user.
21 #
22 # Note that for this feature to be enabled, one is also required
23 # to set one or more the following properties, depending on the
24 # client onboarding dialogs that need the login information to
25 # be overridden by this plugin:
26 # com.openexchange.client.onboarding.caldav.login.customsource=true
27 # com.openexchange.client.onboarding.carddav.login.customsource=true
28 # com.openexchange.client.onboarding.mail.imap.login.customsource=true
29 # com.openexchange.client.onboarding.mail.smtp.login.customsource=true
30 #
31 com.openexchange.plugins.onboarding.login=login

```

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

```

1 # URI to the private and public key resource to use to sign JWTs.
2 #

```



```

3 # The format of the URi depends on the scheme and driver.
4 # The "file" scheme is always supported.
5 #
6 # Format: file:<algorithm>:<path>[#<keyid>]
7 #
8 # Algorithm may either be "auto" in which case the signing algorithm will be inferred
9 # from the EC curve OID within the encoded private key part in the file, or be explicitly
10 # one of the supported values:
11 # - ES256: ECDSA using P-256 curve and SHA-256 hash algorithm
12 # - ES384: ECDSA using P-384 curve and SHA-384 hash algorithm
13 # - ES512: ECDSA using P-521 curve and SHA-512 hash algorithm
14 #
15 # Note that for the time being, only ECDSA keys are supported.
16 #
17 # The key id may be set as the fragment part of the URI: if set, will be stored as a kid (
    key id)
18 # claim in the JWT header, which identifies the key in some form that is understandable
    for consumers
19 # of the JWT token.
20 # Optional, does not set the kid claim when absent.
21 #
22 # The path is a fully qualified filesystem path to the private key PEM file to use for
    signing.
23 # It should also contain the certificate (public key part) in order to include the
24 # x5t#S256 (X.509 certificate SHA-256 thumbprint) in the signed token.
25 #
26 # Content of the file:
27 # -----BEGIN EC PRIVATE KEY-----
28 # MIGHAgEAMBMGBYqGSM49AgEGCCqGSM49AwEHBGOwawIBAQQgyGdEuJcaHlaOCdtX
29 # ...
30 # Jvb9wIBomk0sFr++dEnvM97Sm3G+c8wkqLO+WFBRwTw79sQioT3VOMVV
31 # -----END EC PRIVATE KEY-----
32 # -----BEGIN EC PUBLIC KEY-----
33 # MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEqxHR/v8D3NktT/EfE5Mq2dvlIZ6H
34 # QCb2/cCAaJpDrBa/vnRJ7zPeOptxvnPMJKi9PlhQUcE80/bEIqE91TjFVQ==
35 # -----END EC PUBLIC KEY-----
36 #
37 # The type (specified after BEGIN and END) in the PEM headers must be one of:
38 # - for the mandatory private key: PRIVATE KEY, EC PRIVATE KEY
39 # - for the optional public key: CERTIFICATE, PUBLIC KEY, EC PUBLIC KEY
40 #
41 # Examples:
42 # com.openexchange.plugins.trustedidentity.key=file:auto:/opt/open-xchange/etc/
    trustedidentity.pem#ox-trust-key-2021-1
43 # com.openexchange.plugins.trustedidentity.key=file:ES256:/opt/open-xchange/etc/
    trustedidentity.pem
44 #
45 # Mandatory, there is no default value.
46 com.openexchange.plugins.trustedidentity.key=
47
48 # The issuer (iss) string to include in the signed JWT.
49 # Describes this App Suite instance in its role as an authority.
50 #
51 # Mandatory, has no default value.
52 #
53 # Example:
54 # com.openexchange.plugins.trustedidentity.issuer=Open-Xchange
55 com.openexchange.plugins.trustedidentity.issuer=
56
57 # Expiration duration: the signed JWT contains a standard claim field
58 # "exp" that defines when the validity of the JWT should expire.
59 # The following configuration property configures how long that expiration
60 # time frame should be, always in addition to the current timestamp as
61 # of the system clock.
62 # e.g. "5m" will produce an expiration timestamp that is 5m in the future
63 #
64 # Format: <duration>[h|m|s|ms]
65 #
66 # Example:
67 # com.openexchange.plugins.trustedidentity.expiration=30m
68 #
69 # Optional, the default value is "5m" (5 minutes)

```

```

70 #
71 com.openexchange.plugins.trustedidentity.expiration=5m
72
73 # Public key file (PEM) location on disk.
74 #
75 # This is the public key to use for encrypting JWTs. That public key must be
76 # provided to us by the peer or customer that will receive the encrypted
77 # JWT, as they will be able to decrypt it using their private key part.
78 #
79 # Note tha this property is config-cascade aware.
80 #
81 # Example:
82 # com.openexchange.plugins.trustedidentity.peer.publicKeyFile=/opt/open-xchange/keys/
    customer1-pubkey1.pem
83 #
84 # This configuration setting is mandatory and has no default value.
85 # When left empty, it disables encryption.
86 com.openexchange.plugins.trustedidentity.peer.publicKeyFile=
87
88 # Algorithm to use to encrypt the JWT.
89 #
90 # The supported algorithms depend on the type of the public key.
91 #
92 # For an EC key:
93 #
94 # - ECDH-ES: Elliptic Curve Diffie-Hellman Ephemeral Static (RFC 6090) key agreement using
    the
95 #         Concat KDF, as defined in section 5.8.1 of NIST.800-56A, with the agreed-upon
    key
96 #         being used directly as the Content Encryption Key (CEK) (rather than being
    used to
97 #         wrap the CEK).
98 #
99 # - ECDH-ES+A128KW: Elliptic Curve Diffie-Hellman Ephemeral Static key agreement per "ECDH
    -ES",
100 #         but where the agreed-upon key is used to wrap the Content Encryption Key (CEK
    ) with
101 #         the "A128KW" function (rather than being used directly as the CEK).
102 #
103 # - ECDH-ES+A192KW: Elliptic Curve Diffie-Hellman Ephemeral Static key agreement per "ECDH
    -ES",
104 #         but where the agreed-upon key is used to wrap the Content Encryption Key (CEK
    ) with
105 #         the "A192KW" function (rather than being used directly as the CEK).
106 #
107 # - ECDH-ES+A256KW: Elliptic Curve Diffie-Hellman Ephemeral Static key agreement per "ECDH
    -ES",
108 #         but where the agreed-upon key is used to wrap the Content Encryption Key (CEK
    ) with
109 #         the "A256KW" function (rather than being used directly as the CEK).
110 #
111 # For an RSA key:
112 #
113 # - RSA-OAEP-256: RSAES using Optimal Asymmetric Encryption Padding (OAEP) (RFC 3447),
    with the
114 #         SHA-256 hash function and the MGF1 with SHA-256 mask generation function.
115 #
116 # Note tha this property is config-cascade aware.
117 #
118 # Example:
119 # com.openexchange.plugins.trustedidentity.peer.algorithm=ECDH-ES+A256KW
120 #
121 # The property is optional and defaults to either ECDH-ES for EC keys, or
122 # to RSA-OAEP-256 for RSA keys.
123 com.openexchange.plugins.trustedidentity.peer.algorithm=
124
125 # Encryption Method to use to encrypt the JWT.
126 #
127 # The supported methods are as follows:
128 #
129 # - A128GCM: AES in Galois/Counter Mode (GCM) (NIST.800-38D) using a 128 bit key
130 # - A192GCM: AES in Galois/Counter Mode (GCM) (NIST.800-38D) using a 192 bit key

```

```

131 # - A256GCM: AES in Galois/Counter Mode (GCM) (NIST.800-38D) using a 256 bit key
132 #
133 # Note tha this property is config-cascade aware.
134 #
135 # Example:
136 # com.openexchange.plugins.trustedidentity.peer.encryptionMethod=A256GCM
137 #
138 # The property is optional and defaults to A256GCM
139 com.openexchange.plugins.trustedidentity.peer.encryptionMethod=
140
141 # Peer public key time-to-live in cache.
142 #
143 # Public keys are loaded from PEM files on-demand and are then cached for a configurable
144 # amount of time before being loaded again.
145 #
146 # Format: <duration>[w|d|h|m|s|ms]
147 #
148 # Example:
149 # com.openexchange.plugins.trustedidentity.peer.publicKeyCacheTtl=5d
150 #
151 # The property is optional and defaults to 1d (1 day)
152 com.openexchange.plugins.trustedidentity.peer.publicKeyCacheTtl=

```

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

```

1 # Setting to control the used connector for a specific user
2 # This setting is config-cascade aware to support different implementations for each user.
3 # Default is <none> which means that the feature is disabled for a user
4 # To enable vade secure com.openexchange.plugins.unsubscribe.connector=
   plugins_unsubscribe_vadesecure
5 com.openexchange.plugins.unsubscribe.connector=
6
7 # Setting to enable safe_mode capability via config-cascade
8 # This setting is config-cascade aware to support different implementations for each user.
9 # Default is false which means that the feature is disabled for a user
10 com.openexchange.plugins.unsubscribe.safemode=false

```

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

```

1 # The customer license provided by VadeSecure; required to access unsubscribe API
2 # Default: The OX customer license
3 # Config-cascade aware: true
4 # Lean: true
5 com.openexchange.plugins.unsubscribe.vadesecure.license.passcrypt=<Customer license
   provided by VadeSecure>
6
7 # Setting to change the VadeSecure unsubscribe API URL
8 # Default: https://ws.vaderetro-unsubscribe.com/
9 # Config-cascade aware: true
10 # Lean: true
11 com.openexchange.plugins.unsubscribe.vadesecure.unsubscribe_url=https://ws.vaderetro-
   safeunsubscribe.com/
12
13 # Setting to change the VadeSecure connector identifier referenced in plugins-unsubscribe.
   properties / com.openexchange.plugins.unsubscribe.connector
14 # Default: "plugins_unsubscribe_vadesecure"
15 # Config-cascade aware: true
16 # Lean: true
17 com.openexchange.plugins.unsubscribe.vadesecure.identifier=plugins_unsubscribe_vadesecure

```

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

```

1 # Twilio accountSID
2 com.openexchange.plugins.sms.twilio.accountSID.secret=ACCOUNT_SID
3
4 # Twilio auth token
5 com.openexchange.plugins.sms.twilio.authtoken.secret=AUTH_TOKEN
6
7 # Twilio Message Service SID
8 com.openexchange.plugins.sms.twilio.messageservicesid.secret=SERVICE_SID
9
10 # Max message length. 1600 characters is Twilio's maximum
11 com.openexchange.plugins.sms.twilio.maxlength=1600

```

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

```

1 # Comma separated list of drivers to read into the system
2 # As the sql-client is very early, it may happen that the expected driver is not yet
   registered.
3 # To work around this issue, the following list of drivers will be read before any
   connection is
4 # created.
5 #
6 # Default: com.mysql.jdbc.Driver
7 com.openexchange.sql.client.drivers=com.mysql.jdbc.Driver

```

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

```

1 # The top-level key is the identifier of the pool, which can be
2 # any string of text and is being used by the bundles and applications
3 # to access that pool configuration.
4 # Typically, those are fixed or need to be configured in the bundles
5 # that use this library.
6 #
7 # When Java Security Manager support is enabled, files that are referenced
8 # in these configuration files must be in a directory that is already
9 # whitelisted, or in a subdirectory thereof, such as
10 # /opt/open-xchange/etc/
11 #
12 # A good candidate would be something along the lines of
13 # /opt/open-xchange/etc/sql-files/
14 #
15 # Otherwise, the filename or its directory must be put into a new .list
16 # file in the folder
17 # /opt/open-xchange/etc/security/
18 # with e.g. the following content:
19 #
20 # file:/etc/trust.jks
21 #
22 # For a complete list of property values, read https://github.com/brettwooldridge/HikariCP
23 pool1:
24 # This is the name of the DataSource class provided by the JDBC driver.
25 # Consult the documentation for your specific JDBC driver to get this class name, or see
   the table below.
26 # Note XA data sources are not supported. XA requires a real transaction manager like
   bitronix.
27 # Note that you do not need this property if you are using jdbcUrl for "old-school"
   DriverManager-based JDBC driver configuration.
28 # Default: none
29 dataSourceClassName: com.mysql.jdbc.jdbc2.optional.MysqlDataSource
30 # This property directs HikariCP to use "DriverManager-based" configuration.
31 # 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

```

```

32 # When using this property with "old" drivers, you may also need to set the
    driverClassName property, but try it first without.
33 # 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.
34 # Default: none
35 jdbcUrl: jdbc:mysql://mysql.example.com
36 # This property sets the default authentication username used when obtaining Connections
    from the underlying driver.
37 # Note that for DataSources this works in a very deterministic fashion by calling
    DataSource.getConnection(*username*, password) on the underlying DataSource.
38 # However, for Driver-based configurations, every driver is different.
39 # 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.
40 # If this is not what you need, skip this method entirely and call addDataSourceProperty
    ("username", ...), for example.
41 # Default: none
42 username: user
43 # sets the password of the connection
44 password: secret
45
46 pool2:
47     jdbcUrl: jdbc:mysql://mysql.example.com
48     # This property controls the maximum number of milliseconds that a client (that's you)
        will wait for a connection from the pool.
49     # If this time is exceeded without a connection becoming available, a SQLException will
        be thrown.
50     # Lowest acceptable connection timeout is 250 ms.
51     # Default: 30000 (30 seconds)
52     connectionTimeout: 30000
53     # This property controls the maximum amount of time that a connection is allowed to sit
        idle in the pool.
54     # 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.
55     # Whether a connection is retired as idle or not is subject to a maximum variation of
        +30 seconds, and average variation of +15 seconds.
56     # A connection will never be retired as idle before this timeout.
57     # A value of 0 means that idle connections are never removed from the pool.
58     # The minimum allowed value is 10000ms (10 seconds).
59     # Default: 600000 (10 minutes)
60     idleTimeout: 600000
61     # 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.
62     # On a connection-by-connection basis, minor negative attenuation is applied to avoid
        mass-extinction in the pool.
63     # We strongly recommend setting this value, and it should be several seconds shorter
        than any database or infrastructure imposed connection time limit.
64     # A value of 0 indicates no maximum lifetime (infinite lifetime), subject of course to
        the idleTimeout setting.
65     # Default: 1800000 (30 minutes)
66     maxLifetime: 1800000
67     # This property controls the minimum number of idle connections that HikariCP tries to
        maintain in the pool.
68     # 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.
69     # 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.
70     # Default: same as maximumPoolSize
71     minimumIdle: 0
72     # This property controls the maximum size that the pool is allowed to reach, including
        both idle and in-use connections.
73     # 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.
74     # 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.
75     # Default: 10

```

```
76     maximumPoolSize: 10
77
78 # The following example shows how to provide additional dataSource properties to the pool
79   by using the dataSourceProperties key.
80 # The DataSource will be started with all key-value pairs added.
81 pool3:
82   jdbcUrl: jdbc:mysql://mysql.example.com
83   username: user
84   password: secret
85   dataSourceProperties:
86     useUnicode: true
87     characterEncoding: UTF-8
88     autoReconnect: false
89     useServerPrepStmts: false
90     useTimezone: true
91     serverTimezone: UTC
92     connectTimeout: 15000
93     socketTimeout: 15000
94     useSSL: false
95     requireSSL: false
96     verifyServerCertificate: false
97     enabledTLSProtocols: TLSv1,TLSv1.1,TLSv1.2
```