How to resolve permission configuration issues

Affected version(s): 7.10.4

Currently it is possible to configure or provision permissions via capabilities. E.g. by configuring '*com.openexchange.capability.infostore=false*'. This was unintentional and using this option could lead to unexpected problems. For example a module could be missing because of a deactivated capability. Therefore please be aware that the only supported option to configure permissions is to use permissions itself. For instance by using the user and context command-line tool's *--access-** arguments (e.g. *createuser (...) --access-infostore on/off*).

In order to prevent wrong provisioned users or a bad configuration this option will be deprecated with 7.10.4 and then removed completely with 7.10.5 onwards. The 7.10.4 release will already prohibit the use of permission capabilities by default. This could lead to problems in some existing environment. This document provides help in case you run into those problems because of these new restrictions (see Resolving problems).

General information

It should be mentioned at the beginning that the middleware writes prominent logs for all issues concerning capabilities which are actually permissions. There are actually two scenarios.
The first one concerns the provisioning / configuring of those illegal capabilities. In case an illegal capability is configured - either via property files or via context sets - an error is logged during startup of the server. And in case a provisioning call tries to add an illegal capability to a user the middleware throws an error and also writes an error to the log.

The second one concerns the actual generation of the capability set of an user or context. With 7.10.4 the middleware ignores illegal capabilities by default and always writes a warning to the log for those entries.

Important:
Please note that these restrictions also apply to permissions defined trough the '*permissions*' property, especially since the name of this property can be irritating in this manner.

How to provision permissions the right way

There are actually two possible ways to provision permission. Either with the help moduleaccessdefinitions and access-combination-names or by enabling/disabling the access to modules per user or context.
Which variant you use is up to you, but normally there are three scenarios, each requiring a different approach.

Scenario 1 - All users have the same permissions

This is the simplest scenario. You define a moduleaccessdefinitions for your usecase or use an existing one and provision all users / contexts with the corresponding access-combination-name (e.g. "groupware" or "webmail").

Scenario 2 - You want to offer one or multiple types of premium accounts (e.g. "normal", "premium" and "ultra")

Just like in scenario 1 you define a moduleaccessdefinitions for every type of account you offer and then provision the users/contexts accordingly. For example:

```
normal=webmail,contacts,globaladdressbookdisabled,collectemailaddresses,editpassword
premium=webmail,calendar,contacts,tasks,globaladdressbookdisabled,collectemailaddresses,multiplemailaccounts,
subscription,publication,editpassword
ultra=webmail,calendar,contacts,tasks,infostore,webdav,globaladdressbookdisabled,collectemailaddresses,
multiplemailaccounts,subscription,publication,editpassword
```

Scenario 3 - You want to offer modules independently

This is the most complex scenario. Here it is advicable to change the module access rights per user / context. For example:

```
changeuser (...) --access-infostore on
```

Resolving problems

Like mentioned above it should be easy to identify any problems with the middleware by looking into the logs and checking for errors during startup and for warnings during runtime. If you find any of those errors you need to take action.

The fastest way to resolve any of those issues is to restore the old behaviour with the help of two new properties (7.10.4 only!):

```
com.openexchange.capabilities.allowIllegalPermissionProvisioning=true
com.openexchange.capabilities.applyIllegalPermissions=true
```

For more information about those properties consider the config documentation here.

This is of course not a long term solution because those properties have been removed with the release of App Suite 7.10.5. The proper solution is to change the way you provision permissions. Here are some scenarios:

Scenario 1 - illegal permissions via configuration

Situation:

You configured a permission as a property. For example like this:

```
com.openexchange.capability.infostore = true
```

Solution:

This one is relatively easy to fix. Remove the configuration from your property or yaml files and if necessary adjust the permissions like described in the previous section.

Scenario 2 - illegal permissions via user / context attributes

Situation:

You configured an illegal permission via user or context attributes. For example like this:

```
createuser (...) --config/com.openexchange.capability.infostore=true
```

Solution:

In this scenario you need to remove the user or context attribute with the changeuser or changecontext command-line tool and if necessary you need to adjust the permissions like described in the previous section.

Scenario 3 - illegal permissions via user or context capabilities

Situation:

You configured illegal capabilities on user or context level. For example like this:

```
createuser (...) --capabilities-to-add infostore
```

Solution:

This is similar to scenario 2. Just drop the capability via the *--capabilities-to-drop* argument for every context or user and if necessary you need to adjust the permissions like described in the previous section.

Related articles

- Using browser developer mode
- Disabling the "What's new" Dialog
- Missing configuration options
- Clearing browser cache
- How to resolve permission configuration issues