

Troubleshooting Guide

For Xentral users with on-premise instances upgrading Xentral from version 21.x to version 22.x

Document version 2

Copyright © 2022 Xentral ERP Software GmbH

The information contained in this document is confidential, privileged, and only for the information of the intended recipient and may not be used, published, or redistributed without the prior written consent of Xentral ERP Software GmbH.

Table of Contents

Target audience	4
Known issues and workarounds	5
API calls not working properly	5
Login not possible with "419" error	6
No access to update server and issue with license update	7
Cronjobs not running, update not possible	8
Login fails due to infinite redirect loop to time tracking module	9
Error 500 after update due to missing tables	10
Error 500 after update "Access denied for user @localhost"	11
Outdated license information	12
Missing tables after an update	13
Paid module not usable although active and connected to license	14
Update fails with fatal error	15
Updating process complete, but nothing was updated	16
All process starters result in an error	17

Target audience

This guide is intended for users or administrators of Xentral who:

- work with an on-premise instance
- wish to upgrade from Xentral version 21.x (e.g., 21.1) to Xentral version 22.x (e.g., 22.1 or 22.3)

This guide describes:

- known issues that you might encounter during upgrading to Xentral version 22.x
- solutions for these issues

Known issues and workarounds

In the sections below you will find descriptions of problems that you might encounter during the upgrade and solutions to them.

API calls not working properly

Problem

API calls are not working properly at on-premise instances.

Solution

Please check if the correct `.env` file was used: Sometimes `.env.example` was used, whereas `.env.prod` should have been used. The `.env.example` file is not meant to be used in a production environment as it contains a debug configuration which does not function properly together with the API.

To solve this, execute the following commands in the Xentral directory. Remember to replace `www-data` with your actual web server user if applicable.

```
# Remove faulty .env
rm .env

# Create correct .env
sudo -u www-data php artisan dotenv:migrate

# Apply new .env
sudo -u www-data php artisan xentral:update
```

Login not possible with "419" error

Problem

You cannot log into your instance. You receive a "419 CSRF mismatch" error.

Solution

Make sure that you use SSL. Otherwise, Xentral will not work.

If you use SSL and the problem persists, execute the following commands in the Xentral root directory. Remember to replace `www-data` with your actual web server user if applicable.

```
sed -i "s|^#*SESSION_SECURE_COOKIE.*|SESSION_SECURE_COOKIE=false|g" .env
sudo -u www-data php artisan optimize:clear
```

If the problem then still persists, verify that the `.env` file exists and has read-write mode enabled for the web server user. To this end, compare the owner `user:group` of the `.env` file with that of other files in the Xentral root directory.

After you have confirmed the read-write mode on the `.env` file, or if there is no `.env` file, execute the update again. If possible, execute it from the Xentral user interface; otherwise execute it in the command line interface as shown below. Again, remember to replace `www-data` with your actual web server user if applicable.

```
sudo -u www-data php upgradesystem.php
```

No access to update server and issue with license update

Problem

You can't access the update server because it's expired and you can't update your license in the basic settings in Xentral anymore.

Solution

If you can't change the license data in the basic settings in Xentral anymore, you need to update it directly in your database by using the following commands.

```
# xentral <= 21.1
SELECT f.lizenz, f.schluesel FROM firmendaten f;

# xentral > 21.1
SELECT f.* FROM firmendaten_werte f WHERE f.name IN('lizenz',
'schluesel');
```

Cronjobs not running, update not possible

Problem

The update does not run properly on your Xentral instance and cronjobs might not be running either.

Solution

You need ionCube v. 11 running in your PHP CLI on the server hosting your Xentral instance. If ionCube is missing or outdated, this will lead to issues when you run an update of Xentral. Newer versions of Xentral will automatically check if a suitable ionCube version is running before starting an update.

To find out which ionCube version is running in your environment, you can execute the following command.

```
php -v
```

You will get a response in the command line interface. A valid output of a PHP CLI instance with ionCube installed looks as follows.

```
$ php -v
PHP 7.4.28 (cli) (built: Mar 29 2022 03:26:50) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with the ionCube PHP Loader + ionCube24 v11.0.1, Copyright (c)
    2002-2022, by ionCube Ltd.
```

If your output deviates, you need to load or update the ionCube PHP extension.

Login fails due to infinite redirect loop to time tracking module

Problem

You try to log into your Xentral, but your Xentral gets stuck in an infinite redirect loop to the time tracking ("Zeiterfassung") module. This is a known issue with mandatory time tracking in older Xentral versions (before 22.1).

Solution

As a workaround in versions before 22.1, to enable you to log in to your Xentral again, you can disable the mandatory time tracking directly in the database. Use the following command to disable the mandatory time tracking.

```
UPDATE `firmendaten_werte` SET `wert` = 0 WHERE `name` = 'zeiterfassung_pflicht';
```

In versions 22.1 and higher, the issue has been fixed. Therefore we recommend updating to the latest version.

You can start the update either from the user interface in Xentral or by entering the following command into the command line interface. Remember to replace `www-data` with your actual web server user if applicable.

```
sudo -u www-data php upgradesystem.php
```

Error 500 after update due to missing tables

Problem

After you have run the upgrade from a 21.x version to a 22.x version, an error 500 occurs due to missing tables.

Solution

Typically, this problem is related to large databases. Sometimes, on large database, not all tables are re-created properly on the first update attempt. To fix this, follow the description below.

1. Run the Xentral update by entering the following command into the command line interface and check for any errors.

```
sudo -u www-data php artisan xentral:update
```

Errors will be listed in the command line interface.

2. Follow the hints for resolving the displayed errors.
3. Repeat the steps above: Run the update command again and resolve any errors until all errors have been resolved.
4. If the steps above do not resolve the issue and there are still tables missing, force a full re-run of all legacy database migrations by connecting to the Xentral database and executing the following commands.

```
# Force re-run of all legacy migrations
DELETE FROM migrations WHERE migration =
'2022_02_28_165358_rerun_legacy_migrations_v22_m1';
```

5. Execute the update again and check the results. All migrations should have been executed successfully. You can verify this by entering the following command into the command line interface. Remember to replace `www-data` with the actual web server user if applicable.

```
# Lists missing migrations & should return an empty result
sudo -u www-data php artisan migrate:status | grep " No "
```

Error 500 after update "Access denied for user @localhost"

Problem

After the update to version 22.x, you receive an error 500 "Access denied for user @localhost" when you try to access your Xentral instance.

Solution

The `.env` file is needed for all versions 22.1 and higher. There used to be a bug in the update process that has already been fixed in version 22.1.1. If you nonetheless get this error message, it is most likely due to wrong permissions on the `.env` file.

Verify that the file is owned by the web server user and group: On our cloud instances, this is `www-data:www-data`.

Then enter the following commands into the command line interface. Remember to replace `www-data` with the actual web server user if applicable.

```
# Create .env file
sudo -u www-data php artisan dotenv:migrate

# Clear file caches
sudo -u www-data php artisan optimize:clear

# Update DB
sudo -u www-data php artisan xentral:update
```

Outdated license information

Problem

The license information on your Xentral instance is outdated. For example, your Xentral instance is indicated as a demo instance instead of a rental instance; or your Xentral instance shows a wrong expiry date.

Solution

This is usually caused by wrong permissions on the license file so that `key.php` is read-only and therefore cannot be updated.

To fix this, enter the following commands into the command line interface. Remember to replace `www-data` with your actual web server user, if applicable.

```
# Make the license file writable by the web server user
sudo chown www-data:www-data key.php

# Execute an update to retrieve the new license file
sudo -u www-data php upgradesystem.php
```

Missing tables after an update

Problem

After you upgrade to version 22.x, tables such as "sessions" are missing. This should not happen anymore if you update to version 22.1.17 or higher. It is good practice to always update to the highest available version.

Solution

Missing tables are caused by failed database migrations. Execute the following commands to create the missing tables. Remember to replace `www-data` with the actual web server user, if applicable.

```
# Mandatory: create the sessions table
sudo -u www-data php artisan migrate --force --path=/database/
migrations/legacy/2021_04_21_183354_create_sessions_table.php

# Optional, but recommended: create other missing tables as well
sudo -u www-data php artisan xentral:update
```

If you still run into problems such as unusual system behavior or SQL errors, you can check the migration status for missing tables as follows. Again, if needed, replace `www-data` with your actual web server user.

```
# List missing migrations
sudo -u www-data php artisan migrate:status | grep " No "

# Try to execute them all again
sudo -u www-data php artisan migrate

# If execution fails, move faulty migrations (causing e.g. "Row size
too large" errors) and re-run the migrate command
# Usual suspects:
mv database/migrations/legacy/
2021_04_22_183354_create_adresse_table.php database/migrations/
legacy/2021_04_22_183354_create_adresse_table.php.bak
mv database/migrations/legacy/
2021_04_22_183354_create_projekt_table.php database/migrations/
legacy/2021_04_22_183354_create_projekt_table.php.bak
mv database/migrations/legacy/
2021_04_22_183354_create_verbindlichkeit_table.php
database/migrations/legacy/
2021_04_22_183354_create_verbindlichkeit_table.php.bak
```

Paid module not usable although active and connected to license

Problem

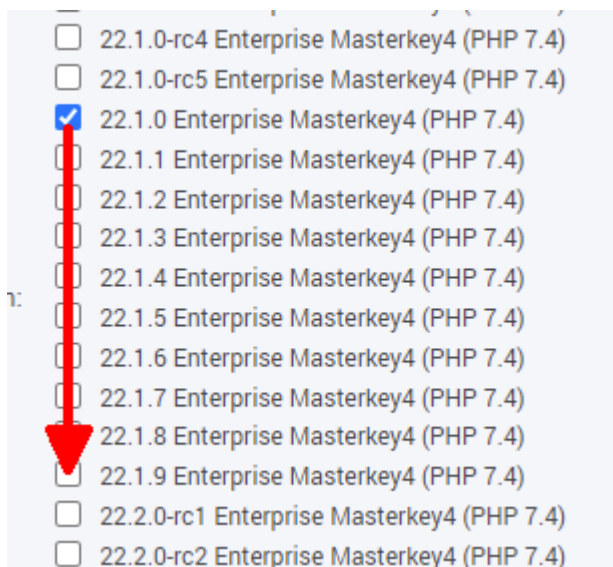
You can't use a paid module (such as Amazon Seller App or Production) even though it is active and properly connected to the license.

Solution

Even though a paid module appears under Modules in a license on the update server, the module file's size is 0 bytes on your instance and therefore cannot be used. You can see this by executing a command such as the following sample command. Remember to replace `www-data` with your actual web server user if applicable.

```
ubuntu@ip-172-31-24-131:~$ ll xentral/download/www/pages/amazon.php
-rw-rw-r-- 1 www-data www-data 0 Feb 16 16:50 xentral/download/www/
pages/amazon.php
```

Usually this is caused by a version mismatch of the MD5 file hash on the update server. To fix the issue temporarily, you must upgrade the license manually to match your exact update version (usually the latest version available) via command or update button.



Afterwards, update your Xentral again.

Update fails with fatal error

Problem

The update fails with a fatal error such as the following.

```
Fatal Error: Class App\Modules\Legacy\CoppersurchargeModule
contains 1 abstract method and must therefore be
declared abstract or implement the remaining methods
(App\Core\Module\Contracts\AppStoreModuleContract::getAppEdition) in
app/Modules/Legacy/CoppersurchargeModule.php on line 9
```

This error is related to modules that have been removed in versions 22.x.

Solution

To fix the problem, enter the commands shown below. Remember to replace `www-data` with your actual web server user if applicable.

```
# Remove the obsolete module file that's stated in the Fatal Error
sudo rm -f app/Modules/Legacy/CoppersurchargeModule.php

# Clear the cache
sudo rm -f bootstrap/cache/*.php

# Re-run the update
sudo -u www-data php artisan xentral:update
```

Updating process complete, but nothing was updated

Problem

The updating process completes, but the version has not changed and no files were updated.

Solution

A likely cause is that the required heartbeat cronjob is missing or has been set incorrectly.



Tip

There are two cronjobs with very similar names - `starter.php` and `starter2.php`. The update process needs the `starter2.php` to be running.

You can check if the required process `starter2.php` is running by executing the following command. If applicable, remember to replace `www-data` with your actual web server user.

```
sudo -u www-data crontab -l
```

If `cronjobs/starter2.php` is not running, set up and start this heartbeat cronjob as follows.

1. Open the editor with the following command.

```
sudo -u www-data crontab -e
```

2. Enter the following command into the editor.

```
* * * * * php /var/www/html/cronjobs/starter2.php
```

This command sets up the cronjob so that it runs every minute.

3. Save the settings.

The cronjob will start within 1 minute.

All process starters result in an error

Problem

After the update to Xentral versions 22.3.x, all process starters result in errors.

Solution

This happens when the Xentral instance is in debug mode instead of in production mode after the update.

The `APP_ENV` setting in the `.env` file determines the mode that the application is running in. In your environment, the `APP_ENV` value must be set to `production`.

To fix this, enter the following commands into the command line interface. Remember to replace `www-data` with your actual web server user, if applicable.

```
sed -i "s|^#*APP_ENV.*|APP_ENV=production|g" .env
2sudo -u www-data php artisan optimize:clear
```