

**Community Tools Against Corruption**

**Installation Guide**

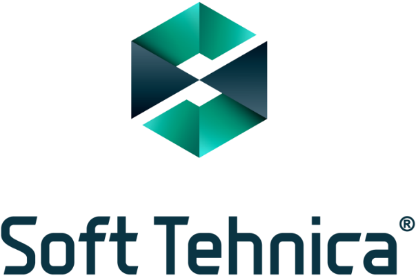


image Documentation licensed under the Creative Commons Attribution-ShareAlike 3.0 License: <http://creativecommons.org/licenses/by-sa/3.0/>

Contents

[**Server configuration:** 3](#_Toc5974609)

[**Database installation:** 3](#_Toc5974610)

[**Source code installation:** 4](#_Toc5974611)

[**ERROR DEBUGGING** 4](#_Toc5974612)

[**Platform development guide** 5](#_Toc5974613)

# **Server configuration:**

This document describes the installation of the CPR platform on a webserver that has at least the following configuration:

1. Apache 2+ functioning as a web server with mod\_rewrite on. It’s possible to user Nginx or other web-servers, but development was done using Apache, so we can’t assure a flawless flight on other system (**Apache 2.4.10**).
2. PHP 7.2+ installed with the following extensions: 'date', 'dom', 'filter', 'gd', 'hash', 'json', 'pcre', 'pdo', 'session', 'SimpleXML', 'SPL', 'tokenizer', 'xml'. A minimum of 64MB should be configured PHP memory limit (**PHP 7.2.7, 512MB**)
3. MySQL 5.5.3/MariaDB 5.5.20 as a Database engine (**MariaDB 5.5.5**).

We have indicated **in bold** the settings used on our test system, which is accessible currently here: <http://cpr.cs1.soft-tehnica.com> and which we used when we developed the platform and for acceptance tests. CPR being a platform for reports collection that is based on Drupal 8 (<https://www.drupal.org/> ), you can check this link for more information about Drupal 8 and its requirements: <https://www.drupal.org/docs/8/system-requirements> for a more complete view and possible fixes for most often encountered errors. After the site will be installed, you can visit ‘*admin/reports/status*’ to check that what configuration is used by Drupal (in case you have multiple sites on the same host, PHP versions, etc).

**About platform security:**  
We also assume that you have root access to the webserver where you’ll be configuring the system, because you will have to unzip the package, import the database dump, change some directory permissions, edit some configurational files, change access rights for various users in order to secure the webserver. We’ll be providing the latest Drupal core and updated contrib modules that we have used as for the moment of writing of this guide – 11.04.2019, but please remember to check periodically the report page (‘*admin/reports/updates*’) and update to at least the latest security-recommended version your core + modules.

**NOTE:** The only custom module that does the platform fine-tuning and had to be written from scratch is ‘**cpr\_extension**’, so you can ignore the warning next to it.

# **Database installation:**

1. Unzip the sources into the directory that will be hosting the site.
2. Create an empty database on your DB server. It’s also recommended to create a separate user for it.
3. Note down the database name, username, password, server IP, port (if it differs from the standard 3306 used by MySQL/MariaDB) – you will need them further.
4. Import the file named d8\_cpr.sql into the database that you have created at step 2.

# **Source code installation:**

1. Unzip the sources into the directory that will be hosting the site
2. Configure the root of the files to be served by the webserver to point to the ‘**web**’ directory that will be unzipped.
3. Make sure the webserver is the main user of the platform’s files (usually it’s named ‘**www-data**’), read here for how to secure your Drupal installation: <https://www.drupal.org/node/244924> .
4. Configure the settings.php file available at ‘**web/sites/default/settings.php**’:
5. correct database settings (row 771 – you should enter there the database name, username, password, host IP for the DB you’ve just configured)
6. uncomment lines 782-784 and enter there the server’s address – don’t forget to escape the dots
7. Refresh the web-address of the site where you’re deploying. You should have a copy of our test-site up and running.

# **ERROR DEBUGGING**

In the event that the site is still not working – don’t worry. As long as you have backups of your previous site or our archive, you can start again from Database installation step 1 (in case you’ve missed some steps or are not sure about one of them). Please check web server error logs/php error logs for more details on what went wrong on your system. Most of the error regarding Drupal installation/migration are already solved and posted on the internet, Drupal being such a popular platform. Below we’re presenting the ones we’ve encountered more often during our deploy process:

1. Check that your webserver (Apache) is up and running and that it is processing .htaccess files. To do this you can enter some bogus strings in the .htaccess file from the ‘*web*’ subdirectory. The server should be presenting a 501/503 error because it will not be able to interpret the file. You can afterwards revert it back by removing the bogus lines.
2. Check that your server is interpreting .php files correctly. You can create a file named ‘test.php’ with only the following text inside ‘<?php phpinfo();’. If ‘yourservername.com/test.php’ opens and you can see information related to your PHP installation - all is fine. You should remove this file afterwards.
3. Check that you DB server is up and running by connecting manually to it using a console window / other means. If there are errors in connecting to the DB, you will have an error page / error log entry specifying that the site was unable to connect to the database. You should double-check your MySQL installation, that the DB exists, the user has enough privileges to access the DB and is allowed to connect from your webserver’s IP (in case the DB server is on another machine). Have in mind internal/external IP addresses.
4. Sometimes you can get a ‘PHP class/file not found’ error – it can be due to pre-filled cache tables. Double-check that the ‘cachetags’ table and other tables with names starting with ‘cache\_’ are empty – if not you should truncate them. Their content will be re-generated with correct filesystem settings.

# **Platform development guide**

With the archive + DB we’ve provided, you can not only deploy multiple copies of the platform, bur also continue developing it. In order to do so, you should just follow the deploy guide and set up a version of the platform on your local machine. We have also included all .scss files, files used for theme development, docker-compose.yml file used for overall development (it will set up a docker based environment that you can use to continue developing the site). Developing the platform is equivalent to enlarging and already existing Drupal site and can be done via manual site configuration in the administrative interface, downloading new modules, patching already downloaded modules, writing custom modules. We advise the person that will extend the platform to be at least fluent in PHP and know what is Drupal 8 and at least be able to write clean code.

After you have your local environment ready, we recommend you log in to the site using ‘**drush uli**’ command that will grant you access to the superadmin Drupal user. You will afterwards be able to adjust the site’s configuration for your needs.

We used **git** during CPR development and strongly recommend you to import the existing code into a git repository, then share it to whatever team of developers will be working on it in order to have the possibility to separate various versions that could differ from project to project.

The delivery package contains this guide, the source code and a DB dump with initial data (live data taken from <https://faracoruptie.md/> + <https://cpr.md/> during import phase).