

**Общество с ограниченной ответственностью “Бионика Диджитал”**

*ОГРН 1127747266998 ИНН 7726711455*

**BDigital CMS**

**Технические характеристики и установка ПО**

2023 г.

<b>1. Технологии и требования.....</b>	<b>2</b>
1.1. Сервер.....	2
1.2. Клиент.....	2
<b>2. Установка.....</b>	<b>3</b>
2.1. Установка PHP 8.1.....	3
2.2. Установка и настройка Nginx.....	4
2.3. Установка и настройка PostgreSQL или MySQL.....	5
2.4. Извлечение архива с ПО в целевую директорию.....	5
2.5. Запуск скрипта установки.....	6
2.6. Установка Composer *.....	8
<b>3. Настройка Cron.....</b>	<b>8</b>
<b>4. Обновление.....</b>	<b>9</b>
4.1 Ручное обновление.....	9
4.2 Полуавтоматическое обновление.....	9
<b>5. Используемые библиотеки и компоненты.....</b>	<b>10</b>
5.1. Backend.....	10
5.2. Frontend.....	13

# 1. Технологии и требования

## 1.1. Сервер

<b>Linux (Ubuntu 18.04+, Debian 10+)</b>	Операционная система
<a href="#">Nginx</a>	Веб-сервер
<a href="#">PHP 8.1+</a>	Язык backend
<a href="#">PostgreSQL 14+</a> / <a href="#">MySQL 8+</a> / <a href="#">Percona 8+</a>	Реляционная СУБД для хранения информации

## 1.2. Клиент

Любой браузер актуальной версии (**Google Chrome, Mozilla Firefox, Yandex Browser, Opera, Safari, Microsoft Edge**) с поддержкой **Javascript**.

Верстка адаптивна и работает как в десктопных, так и в мобильных браузерах.

BDigital CMS не будет работать в браузере **Internet Explorer** любой версии.

## 2. Установка

ПО распространяется в виде готового архива, содержащего в себе все необходимые зависимости. Их обновление можно осуществлять, имея доступ к внешним репозиториям с помощью Composer.

В дальнейшем предполагается, что у вас уже имеется настроенный сервер с операционной системой **Linux** (Ubuntu, Debian, и т.п.) с установленным пакетным менеджером **APT**.

### 2.1. Установка PHP 8.1

Добавьте репозиторий с актуальными обновлениями:

```
sudo add-apt-repository ppa:ondrej/php
```

Установите PHP со всеми необходимыми для работы пакетами:

```
sudo apt install php8.1 php8.1-common php8.1-cli php8.1-fpm php8.1-gd  
php8.1-mbstring php8.1-curl php8.1-xml php8.1-zip php8.1-bcmath  
php8.1-intl
```

В зависимости от используемой СУБД установите расширения:

- `sudo apt install php8.1-pgsql` для PostgreSQL
- `sudo apt install php8.1-mysql` для MySQL / Percona

### 2.2. Установка и настройка Nginx

Установите Nginx и текстовый редактор Nano (или воспользуйтесь существующим редактором):

```
sudo apt install nginx nano
```

Создайте файл с настройкой (имя файла может быть любым):

```
sudo nano /etc/nginx/sites-available/example.com.conf
```

И добавьте туда содержимое следующего вида:

```

server {
    listen 80;

    root /var/www/web/example/public;
    index index.html index.htm index.php;

    server_name example.com;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    location ~ /\.ht {
        deny all;
    }

    location ~ \.php$ {
        include /etc/nginx/fastcgi_params;
        fastcgi_index index.php;
        fastcgi_pass unix:/run/php/php8.1-fpm.sock;
        fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        fastcgi_param SCRIPT_NAME $fastcgi_script_name;
    }

    access_log /var/log/nginx/example.com_access.log;
    error_log /var/log/nginx/example.com_error.log;
}

```

Обратите внимание, что корневой должна являться подпапка **public**.

Сделайте симлинк в директорию **/etc/nginx/sites-enabled/**:

```

sudo ln -s /etc/nginx/sites-available/example.com.conf
/etc/nginx/sites-enabled/

```

Перезагрузите Nginx для применения изменений:

```

sudo service nginx restart

```

## 2.3. Установка и настройка PostgreSQL или MySQL

Из-за большого количества вариантов установки и настройки СУБД, описание данного процесса выходит за рамки данной документации. Вы можете воспользоваться следующими ресурсами:

- **PostgreSQL:**  
<https://mcs.mail.ru/docs/additionals/cases/cases-db-config/case-psql-create>
- **MySQL или Percona:**  
<https://mcs.mail.ru/docs/additionals/cases/cases-db-config/case-mysql-create>

## 2.4. Извлечение архива с ПО в целевую директорию

В пункте 2.2 (Установка Nginx) по умолчанию настройки были для директории `/var/www/web/example/`, далее для наглядности целевая директория будет обозначаться шаблоном `<dir>`.

Архив можно загрузить по **FTP** или **SFTP**, либо скачать по прямой ссылке (ссылку на скачивание можно получить у менеджера компании).

```
cd <dir> && wget <ссылка_на_архив>/cms.zip
```

Распакуйте архив с последующим удалением:

```
sudo apt install unzip && unzip cms.zip && rm cms.zip
```

## 2.5. Запуск скрипта установки

```
php setup.php
```

Значение в квадратных скобках является значением по умолчанию, достаточно просто нажимать Enter для его применения. В случае необходимости ввода определенных значений, значение по умолчанию указывается в верхнем регистре.

Сначала необходимо указать хост (доменное имя) сайта.

```
Enter application host [localhost]: example.com
```

Затем необходимо указать внутреннее название сайта. Это необходимо в случае размещения нескольких сайтов в одном контуре. По умолчанию необходимо

использовать значение **default** (все первичные настройки и шаблоны указаны именно для сайта default).

```
Enter application name [default]:
```

Затем скрипт предложит настроить окружение для production (готового сайта, к которому имеют доступ все пользователи). Для production настраивается кэширование и отключается подробное отображение ошибок.

```
Setup environment for the production? [y/N]:
```

Настройки окружения в BDigital CMS могут храниться во внутренней структуре папок в отдельных конфигурационных файлах (**.local.php**), либо в файлах [.env](#), размещенных в корне.

```
Store config in .env file instead of .local files? [y/N]:
```

Далее происходит настройка соединения с базой данных. Укажите настройки в зависимости от используемой СУБД.

```
Configuring "default" connection...  
Driver [mysql/PGSQL]: pg
```

Для выбора **PostgreSQL** можно просто нажать Enter, либо ввести **pgsql**, **pg** или **postgres**.

Укажите хост или ip-адрес используемой СУБД.

```
Host [localhost]:
```

Укажите имя пользователя и его пароль.

```
Username [root]: username  
Password []: password
```

Укажите название используемой базы данных.

```
Database [database]:
```

Порт для соединения с СУБД.

```
Port [5432]:
```

Затем на экране будут отображены все указанные ранее настройки с предложением подтвердить их правильность.

```
Array
(
    [driver] => pgsql
    [host] => localhost
    [username] => username
    [password] => password
    [dbname] => database
    [port] => 5432
)

Are these settings correct? [Y/n]:
```

После подтверждения будет произведена начальная настройка сайта (миграция базы данных и заполнение тестовыми данными, генерация кэша и т.п.).

```
Updating applications (cache, migrate)...
DEFAULT ... ✓
```

## 2.6. Установка Composer \*

\* Менеджер пакетов **Composer** устанавливается опционально для последующей возможности обновления зависимостей и компонентов ПО.

Документация по различным вариантам установки Composer содержится на официальном сайте: <https://getcomposer.org/download/>. Ниже будет приведен пример локальной установки. Предполагается, что вы находитесь в директории, содержащей файлы ПО (<dir> из предыдущих шагов).

```
wget https://getcomposer.org/download/2.6.2/composer.phar && sudo mv
composer.phar /usr/local/bin/composer
```

После указанных действий можно выполнять команду `composer update` для обновления.



### 3. Настройка Cron

Cron (**Command Run ON**) — система для автоматического запуска программ и скриптов на сервере в определённое время.

Cron (планировщик) используется для отложенной публикации записей, обновления кэша и очистки мусора в БД.

Выполните следующую команду для добавления нового задания:

```
crontab -e
```

Добавьте следующую запись в конец файла:

```
* * * * * /usr/bin/php8.1 /var/www/web/example/index.php -a=default  
-r=cron -u=1 1>> /dev/null 2>&1
```

Расшифровка переменных:

<b>/var/www/web/example/</b>	Путь до папки с CMS из конфигурации Nginx
<b>-a=default</b>	Название сайта, которое было настроено в setup.php
<b>-u=1</b>	ID пользователя в базе с правами администратора

Задача будет добавлена в очередь автоматически после сохранения файла.

В результате вышеуказанной операции Cron будет каждую минуту обращаться к сайту, а непосредственное выполнение запланированных задач будет осуществлять сама CMS.

## 4. Обновление

Обновление ПО может быть осуществлено 2-мя способами. Все ссылки на архивы можно получить у менеджера компании. Полностью автоматическое обновление не предусмотрено.

### 4.1 Ручное обновление

Для осуществления ручного обновления необходимо получить ссылку на архив, содержащий весь код ПО и его зависимости вида <https://bionika.digital/.../cms.zip>

Затем в консоли в папке с файлами выполнить следующую команду:

```
wget https://bionika.digital/.../cms.zip && unzip -o cms.zip && rm cms.zip
```

После завершения операции произвести миграцию БД и пересборку кэша командой:

```
php cli -a=default -c -m --cache
```

Где **default** - имя вашего приложения.

### 4.2 Полуавтоматическое обновление

Для полуавтоматического обновления необходимо получить ссылку на архив, содержащий только код ПО без включенных зависимостей вида <https://.../prototype-master.zip>, а также логин и пароль для доступа к системе контроля версий.

Затем в консоли в папке с файлами выполните следующую команду:

```
wget https://.../prototype-master.zip && unzip -o prototype-master.zip && mv prototype-master/* . && rm -Rf prototype-master/ && rm prototype-master.zip
```

После завершения операции необходимо произвести обновление зависимостей, миграцию БД и пересборку кэша с помощью **composer** командой:

```
composer install --no-dev
```

## 5. Используемые библиотеки и компоненты

### 5.1. Backend

- 5.1.1. <https://github.com/adhocore/php-jwt> - Ultra lightweight JSON web token (JWT) library for PHP5.5+.
- 5.1.2. <https://github.com/asika32764/php-autolink> - Auto convert url to link anchor.
- 5.1.3. <https://github.com/wikimedia/less.php> - PHP port of the Javascript version of LESS <http://lesscss.org> (Originally maintained by Josh Schmidt)
- 5.1.4. <https://github.com/bocharsky-bw/Arrayzy> - A native PHP arrays easy manipulation library in OOP way.
- 5.1.5. <https://github.com/box/spout> - PHP Library to read and write spreadsheet files (CSV, XLSX and ODS), in a fast and scalable way
- 5.1.6. <https://github.com/brick/varexporter> - A powerful alternative to `var_export()`, which can export closures and objects without `__set_state()`
- 5.1.7. <https://github.com/caxy/php-htmldiff> - A library for comparing two HTML files/snippets and highlighting the differences using simple HTML.
- 5.1.8. <https://github.com/centrifugal/phpcent> - PHP library to communicate with Centrifugo HTTP API
- 5.1.9. <https://github.com/claviska/SimpleImage> - A PHP class that makes working with images as simple as possible.
- 5.1.10. <https://github.com/cocur/background-process> - Start processes in the background that continue running when the PHP process exists.
- 5.1.11. <https://github.com/crazycodr/standard-exceptions> - This project is aimed at providing additional standard exceptions to php.
- 5.1.12. <https://github.com/danielstjules/Stringy> - A string manipulation library with multibyte support
- 5.1.13. <https://github.com/darkfriend/php7-xml> - PHP7 library for encode/decode xml
- 5.1.14. <https://github.com/davidbarratt/custom-installer> - Install custom types into custom locations
- 5.1.15. <https://github.com/digitalnature/php-ref> - A nicer `print_r/var_dump` alternative for PHP 5.3+
- 5.1.16. <https://github.com/erusev/parsedown> - Parser for Markdown.
- 5.1.17. <https://github.com/FakerPHP/Faker> - Faker is a PHP library that generates fake data for you.

- 5.1.18. <https://github.com/FoolCode/SphinxQL-Query-Builder> - A PHP query builder for SphinxQL. Uses MySQLi to connect to the Sphinx server.
- 5.1.19. <https://github.com/guzzle/guzzle> - Guzzle is a PHP HTTP client library
- 5.1.20. <https://github.com/vinkla/hashids> - Generate short, unique, non-sequential ids (like YouTube and Bitly) from numbers
- 5.1.21. <https://github.com/IcyApril/CryptoLib> - CryptoLib: secure, free & open-source PHP cryptography library for everyone.
- 5.1.22. <https://github.com/ihor/Cachalot> - Cache a lot in a proper way (APC, XCache, Memcached, Redis, Couchbase)
- 5.1.23. <https://github.com/j4mie/idiorm> - A lightweight nearly-zero-configuration object-relational mapper and fluent query builder for PHP5
- 5.1.24. <https://github.com/jasonlewis/expressive-date> - A fluent extension to PHP's DateTime class.
- 5.1.25. <https://github.com/kevinlebrun/colors.php> - Colors for PHP CLI scripts
- 5.1.26. <https://github.com/kint-php/kint> - Kint - debugging tool for PHP developers
- 5.1.27. <https://github.com/thephpleague/flysystem-sftp> - SFTP filesystem adapter for Flysystem.
- 5.1.28. <https://github.com/matthiasmullie/minify> - CSS & JavaScript minifier, in PHP. Removes whitespace, strips comments, combines files
- 5.1.29. [https://github.com/mk-j/PHP\\_XLSXWriter](https://github.com/mk-j/PHP_XLSXWriter) - PHP Library to write XLSX files
- 5.1.30. <https://github.com/serbanghita/Mobile-Detect> - Mobile\_Detect is a lightweight PHP class for detecting mobile devices.
- 5.1.31. <https://github.com/moneyphp/money> - PHP implementation of Fowler's Money pattern
- 5.1.32. <https://github.com/Seldaek/monolog> - Sends your logs to files, sockets, inboxes, databases and various web services
- 5.1.33. <https://github.com/mpdf/mpdf> - PHP library generating PDF files from UTF-8 encoded HTML
- 5.1.34. <https://github.com/mpdf/qrcode> - QR code generator for mPDF
- 5.1.35. <https://github.com/bobthecow/mustache.php> - A Mustache implementation in PHP.
- 5.1.36. <https://github.com/myclabs/DeepCopy> - Create deep copies (clones) of your objects
- 5.1.37. <https://github.com/nette/php-generator> - Nette PHP Generator: generates neat PHP code for you
- 5.1.38. <https://github.com/nikic/PHP-Parser> - A PHP parser written in PHP

- 5.1.39. <https://github.com/nullivex/lib-array2xml> - Array2XML conversion library credit to lalit.org
- 5.1.40. <https://github.com/opis/closure> - A library that can be used to serialize closures (anonymous functions) and arbitrary objects.
- 5.1.41. [https://github.com/paragonie/random\\_compat](https://github.com/paragonie/random_compat) - PHP 5.x polyfill for random\_bytes() and random\_int() from PHP 7
- 5.1.42. <https://github.com/peppeocchi/php-cron-scheduler> - PHP Cron Job Scheduler
- 5.1.43. <https://github.com/PHPMailer/PHPMailer> - PHPMailer is a full-featured email creation and transfer class for PHP
- 5.1.44. <https://github.com/phpseclib/phpseclib> - PHP Secure Communications Library - Pure-PHP implementations of RSA, AES, SSH2, SFTP, X.509 etc.
- 5.1.45. <https://github.com/sebastianbergmann/phpunit> - The PHP Unit Testing framework.
- 5.1.46. <https://github.com/cakephp/phinx> - Phinx makes it ridiculously easy to manage the database migrations for your PHP app.
- 5.1.47. <https://github.com/Setasign/FPDI> - FPDF is a collection of PHP classes facilitating developers to read pages from existing PDF documents
- 5.1.48. <https://github.com/tedious/Stash> - The place to keep your cache
- 5.1.49. <https://github.com/tdlib/telegram-bot-api> - PHP Wrapper for Telegram Bot API
- 5.1.50. <https://github.com/tijsverkoyen/CssToInlineStyles> - CssToInlineStyles is a class that enables you to convert HTML-pages/files into HTML-pages/files with inline styles.
- 5.1.51. <https://github.com/vlucas/phpdotenv> - Loads environment variables from `.env` to `getenv()`, `\$\_ENV` and `\$\_SERVER` automatically.
- 5.1.52. <https://github.com/voku/HtmlMin> - HTML Compressor and Minifier
- 5.1.53. [https://github.com/voku/simple\\_html\\_dom](https://github.com/voku/simple_html_dom) - Simple HTML DOM package.
- 5.1.54. <https://github.com/yoomoney/yookassa-sdk-php> - This is a developer tool for integration with YooMoney
- 5.1.55. <https://github.com/zordius/lightncandy> - An extremely fast PHP implementation of handlebars ( <http://handlebarsjs.com/> ) and mustache ( <http://mustache.github.io/> ).

## 5.2. Frontend

- 5.2.1. <https://github.com/mateuszkocz/3l>
- 5.2.2. <https://github.com/animate-css/animate.css>
- 5.2.3. <https://github.com/blueimp/jQuery-File-Upload>
- 5.2.4. <https://github.com/centrifugal/centrifuge>

- 5.2.5. <https://github.com/chartist-js/chartist>
- 5.2.6. <https://github.com/jackmoore/colorbox>
- 5.2.7. <https://github.com/fengyuanchen/cropperjs>
- 5.2.8. <https://github.com/Mikhus/domurl>
- 5.2.9. <https://github.com/Studio-42/eIFinder>
- 5.2.10. <https://www.npmjs.com/package/ev-emitter>
- 5.2.11. <https://github.com/fomantic/Fomantic-UI>
- 5.2.12. <https://github.com/headjs/headjs>
- 5.2.13. <https://github.com/desandro/imagesloaded>
- 5.2.14. <https://github.com/jquery/jquery>
- 5.2.15. <https://github.com/jquery-form/form>
- 5.2.16. <https://www.npmjs.com/package/js-cookie>
- 5.2.17. <https://github.com/vakata/jstree>
- 5.2.18. <https://github.com/aFarkas/lazysizes>
- 5.2.19. <https://www.npmjs.com/package/moment>
- 5.2.20. <https://github.com/needim/noty>
- 5.2.21. <https://www.npmjs.com/package/rangeslider-js>
- 5.2.22. <https://www.npmjs.com/package/sockjs-client>
- 5.2.23. <https://github.com/tinymce/tinymce>
- 5.2.24. <https://github.com/buonguyen/topbar>
- 5.2.25. <https://github.com/alpinejs/alpine>
- 5.2.26. <https://www.npmjs.com/package/downloadjs>
- 5.2.27. <https://github.com/onury/geolocator>
- 5.2.28. <https://github.com/uNmAnNeR/imaskjs>
- 5.2.29. <https://github.com/metafizzy/infinite-scroll>
- 5.2.30. <https://github.com/hilios/jQuery.countdown>
- 5.2.31. <https://github.com/hgoebl/mobile-detect.js>
- 5.2.32. <https://github.com/marcuswestin/store.js>
- 5.2.33. <https://github.com/elbywan/wretch>