

АКЦИОНЕРНОЕ ОБЩЕСТВО «СЭЙФ КОНТРАКТ»

СИСТЕМА УПРАВЛЕНИЯ САЙТОМ «BOOSTALL»

**ИНСТРУКЦИЯ ПО УСТАНОВКЕ ЭКЗЕМПЛЯРА ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ, ПРЕДОСТАВЛЕННОГО ДЛЯ ПРОВЕДЕНИЯ
ЭКСПЕРТНОЙ ПРОВЕРКИ**

Системные требования серверного ПО

1. Web-сервер **Nginx** — рекомендован для производительности и безопасности (Apache можно использовать как альтернативу, поддерживая совместимость с Laravel)
2. Поддержка языка **PHP 8.1** и выше (обязательные расширения, так называемые модули языка – **BCMath** (математические операции с большими числами), **Ctype** (проверка типов символов), **Fileinfo** (определение типов файлов), **JSON** (работа с данными в формате JSON), **Mbstring** (обработка многобайтовых строк), **OpenSSL** (шифрование и работа с SSL/TLS), **PDO** (универсальная работа с базами данных), **Tokenizer** (анализ кода, работа с токенами PHP), **XML** (обработка XML-данных)). Эти расширения необходимы для обеспечения функциональности Laravel
3. Сервер СУБД **MySQL версии 8.0** и выше (альтернативы **PostgreSQL версии 13** и выше, а также в локальной среде при разработке допустимо использование облегченной среды SQLite)
4. Операционная система: **Linux** (например, **Ubuntu 22.04**) для лучшей производительности и поддержки
5. **Composer**: инструмент управления зависимостями, версия 2.5+
6. **Node.js** и **NPM/Yarn**: для работы с frontend-зависимостями, Node.js версии 18+

Инструкция по установке CMS BoostAll

Перенесите архив проекта на сервер (ссылку на проект предоставляет менеджер), используя SFTP/FTP или scp. Распакуйте архив в директорию, где будет размещаться проект, например:

```
unzip cms_src.zip -d /var/www/laravel
```

Убедитесь, что файлы распакованы корректно. Настройка прав доступа.

Назначьте правильные права для папок *storage* и *bootstrap/cache*:

```
sudo chown -R www-data:www-data /var/www/laravel
sudo chmod -R 775 /var/www/laravel/storage
/var/www/laravel/bootstrap/cache
```

Настройка переменных окружения. В корне проекта должен быть файл *.env* (или *.env.example*). Если файл отсутствует, создайте его:

```
cp .env.example .env
```

Откройте файл *.env* и настройте подключение к базе данных, указав:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=имя_базы_данных
DB_USERNAME=пользователь
DB_PASSWORD=пароль
```

Установка зависимостей. Убедитесь, что *Composer* установлен на сервере.

Если нет, установите его:

```
php -r
"copy('https://getcomposer.org/installer',
'composer-setup.php');"
php composer-setup.php
mv composer.phar /usr/local/bin/composer
```

Установите зависимости проекта:

```
cd /var/www/laravel
composer install --no-dev --optimize-autoloader
```

Генерация ключа приложения. Сгенерируйте ключ приложения, если он не был сгенерирован ранее:

```
php artisan key:generate
```

Миграции и сидирование базы данных. База данных требует миграций и сидирования, выполните команды:

```
php artisan migrate --force
```

```
php artisan db:seed --force
```

Настройка веб-сервера. Для Nginx создайте конфигурационный файл:

```
server {
    listen 80;
    server_name example.com;
    root /var/www/laravel/public;

    index index.php index.html;

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

    location ~ /\.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.1-
        fpm.sock;
        fastcgi_param SCRIPT_FILENAME
        $document_root$fastcgi_script_name;
        include fastcgi_params;
    }

    location ~ /\.ht {
        deny all;
    }
}
```

Перезапустите Nginx:

```
sudo systemctl restart nginx
```

Для Apache включите модуль *mod_rewrite* и настройте *.htaccess*

```
sudo a2enmod rewrite
sudo systemctl restart apache2
```

Очистка и кэширование. Оптимизируйте конфигурацию и кэш проекта:

```
php artisan config:cache
php artisan route:cache
php artisan view:cache
```