Настройка

Содержание

> *Большая часть приведенных настроек автоматизированно выполняется на этапе установки. Здесь приведены примеры самостоятельной настройки приложения.*

Настройка окружения

Перейдите в директорию приложения и откройте файл .env. Если его нет, то создайте копию файла .env.example и переименуйте его в .env.

Если в папке с приложением отсутствуют и файл .env, и файл .env.example, то создайте пустой файл .env и добавьте в него следующие базовые настройки:

```
APP_NAME=
APP_URL=
APP_KEY=
APP_ENV=production
APP_DEBUG=false

LOG_CHANNEL=stack
LOG_LEVEL=error

BROADCAST_DRIVER=redis
CACHE_DRIVER=file
```

```
QUEUE_CONNECTION=sync
SESSION_DRIVER=database
SESSION_LIFETIME=120

MEMCACHED_HOST=127.0.0.1

REDIS_CLIENT=predis
REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

COMPOSER_HOME="/tmp/composer"
```

1. Создайте криптографический ключ:

```
php artisan key:generate
```

2. Откройте файл .env.
3. Настройте название приложения:

```
APP_NAME="Личный кабинет"
APP_ENV=production
APP_KEY=base64:Z234567890900NEb1TmOzWC5673nM5zLHb6h30o4n90c=
APP_DEBUG=FALSE
APP_URL=http://lk.example.com
APP_TIMEZONE=Europe/Moscow
```

4. Настройте параметры соединения с БД:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=dbname
DB_USERNAME=dbuser
DB_PASSWORD=password
```

5. Настройте параметры электронной почты для рассылок:

```
MAIL_MAILER=log
MAIL_HOST=mail.example.ru
MAIL_PORT=25
MAIL_USERNAME=
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
```

```
MAIL_FROM_ADDRESS=no-reply@example.ru
MAIL_FROM_NAME="${APP_NAME}"
```

6. Если используется LDAP введите для него настройки:

```
LDAP_HOST=127.0.0.1
LDAP_USERNAME='cn=admin,dc=example,dc=com'
LDAP_PASSWORD=1
LDAP_BASE_DN='dc=example,dc=com'
LDAP_LOGGING=FALSE
```

Настройка электронной почты

Настройка параметров отправки почты осуществляется в файле .env, который находится в корне директории установки приложения ЛК. Настраиваемые параметры:

- MAIL_MAILER – механизм отправки сообщений. Для отправки через почтовый сервер используется значение smtp.
- MAIL_HOST – адрес почтового сервера.
- MAIL_PORT – порт.
- MAIL_USERNAME – имя пользователя.
- MAIL_PASSWORD – пароль.
- MAIL_ENCRYPTION – режим шифрования. Обычно указывается tls, если шифрование выключено необходимо обязательно указать null.
- MAIL_FROM_ADDRESS – адрес электронной почты, который будет указан в качестве адреса отправителя.
- MAIL_FROM_NAME – имя отправителя. Например Личный кабинет.

Устранение неполадок

В случае если почтовый сервер использует шифрование, но при этом используется самоподписанный сертификат будет возникать ошибка

```
stream_socket_enable_crypto(): SSL operation failed with code 1. OpenSSL Error messages:
error:1416F086:SSL
```

routines:tls_process_server_certificate:certificate verify failed

Для ее устранения без изменения сертификата необходимо добавить в файл конфигурации config/mail.php следующие параметры:

```
'stream' => [
   'ssl' => [
       'allow_self_signed' => true,
       'verify_peer' => false,
       'verify_peer_name' => false,
   ],
],
```

При использовании данных настроек полностью отключается верификация сервера отправки сообщений, и добавляется возможность использования самоподписанных сертификатов. Однако стоит отметить, что в таком случае невозможно гарантировать безопасность доставки сообщений.

Использование обратного прокси

Добавление в список доверенных серверов

Если между сервером, на котором установлено приложение ЛК, и внешней сетью настроен балансировщик нагрузки (обратный прокси/ reverse proxy), то его необходимо добавить в список доверенных серверов.

Перейдите в директорию установки приложения и затем перейдите в директорию app/Http/Middleware. Откройте файл TrustProxies.php и откройте его для редактирования. Найдите переменную $proxies (**не создавайте новую**, переменная уже должна существовать). Измените ее значение следующим образом:

```
protected $proxies = [
```

```
    '192.168.1.1',
    '192.168.1.2',
];
```

Где вместо адресов 192.168.1.1 и 192.168.1.2 необходимо указать корректные значения адресов балансировщиков нагрузки.

Если адреса серверов неизвестны, то можно разрешить все адреса:

```
protected $proxies = '*';
```

Сохраните и закройте файл.

Проверка настроек в файле .env

Проверьте настройки в файле .env. В переменной APP_URL должен быть адрес приложения, включающий протокол https. Напоминаем, что если файл конфигурации был закэширован, то кэш необходимо очистить:

```
php artisan config:clear
```

Принудительное использование https

Откройте файл app/Providers/AppServiceProvider.php. В метод boot необходимо добавить следующую строку:

```
\URL::forceScheme('https');
```

Пример. Если не было ранее внесено других изменений, метод может выглядеть следующим образом:

```
/**
 * Bootstrap any application services.
 *
 * @return void
 */
public function boot()
{
     \URL::forceScheme('https');
}
```

Сохраните и закройте файл.

Проверка передаваемых заголовков

Проверьте, что балансировщик или обратный прокси передает заголовок x-forwarded-proto.

Настройка директории public

Для работы приложения необходимо настроить корневую папку, из которой будет запускаться приложение. В директории приложения находится папка public, а внутри папки находится файл index.php. Необходимо настроить виртуальный хост веб-сервера таким образом, чтобы он указывал на папку public.

Права доступа к директориям

Все вложенные папки и файлы в директориях storage и bootstrap/cache должны быть доступны для чтения и записи веб-сервером.

Настроить пользователя как владельца приложения.

Предполагается, что /var/www/html/laravel/root/directory — путь до директории приложения, имя вашего пользователя tandem, а имя пользователя для веб-сервера www-data.

Замените данные параметры на свои и выполните команды в терминале:

```
sudo usermod -a -G www-data tandem
```

```
sudo chown -R tandem:www-data /var/www/html/laravel/root/directory
```

И затем настройте права доступа:

```
sudo find /path/to/your/laravel/root/directory -type f -exec chmod 664 {} \;
```

```
sudo find /path/to/your/laravel/root/directory -type d -exec chmod 775 {} \;
```

Настройка Supervisor

В рамках платформы реализована возможность работы с заданиями jobs и очередями queue, которые могут обрабатываться в фоновом режиме *(все трудоемкие задания перемещаются в очередь и выполняться в фоне, что позволяет быстрее обрабатывать веб-запросы и отвечать клиенту)*.

Для работы очередей требуется настроить Supervisor:

- Если Supervisor не установлен, требуется выполнить его установку *(пример для ОС Ubuntu и Debian – https://www.digitalocean.com/community/tutorials/how-to-install-and-manage-supervisor-on-ubuntu-and-debian-vps)*.
- В файле .env для переменной QUEUE_CONNECTION установить значение database:
  QUEUE_CONNECTION=database

- Очистить кэш конфигурационного файла php *{путь до корня проекта}*/artisan config:cache.
- Создать конфигурационный файл для запуска очереди: nano /etc/supervisor/conf.d/base.conf со следующим примерным содержимым:

[program:*{НАЗВАНИЕ, например:* lk-base*}***]**
process_name=%(program_name)s_%(process_num)02d
command=php *{ПУТЬ К ПРОЕТУ, например:* /var/www/html*}***/artisan queue:work --queue=default,base --timeout=3600**
autostart=true
autorestart=true

stopasgroup=true
killasgroup=true
user=*{ПОЛЬЗОВАТЕЛЬ КОТОРЫЙ БУДЕТ ВЫПОЛНЯТЬ ПРОЦЕСС, например:* tandem*}*
numprocs=1
startsecs=0
redirect_stderr=true
environment=HOME="*{ДОМАШНЯЯ ДИРЕКТОРИЯ ИСПОЛНЯЕМОГО ПОЛЬЗОВАТЕЛЯ, например:* /home/tandem*}",USER="{ПОЛЬЗОВАТЕЛЬ КОТОРЫЙ БУДЕТ ВЫПОЛНЯТЬ ПРОЦЕСС, например:* tandem*}"*
stdout_logfile=*{ПУТЬ ДЛЯ ХРАНЕНИЯ ЛОГОВ, например:* /home/tandem*}/lk-base-worker.log*
stopwaitsecs=3600

- Сохранить файл base.conf
- Создать конфигурационный файл для запуска системной очереди: nano /etc/supervisor/conf.d/system-base.conf со следующим примерным содержимым:

[program:*{НАЗВАНИЕ, например:*lk-system-base*}]*
process_name=%(program_name)s_%(process_num)02d
command=php *{ПУТЬ К ПРОЕТУ, например:* /var/www/html*}/artisan* **queue:work --queue=system --force --timeout=3600**
autostart=true
autorestart=true
stopasgroup=true
killasgroup=true
user=*{ПОЛЬЗОВАТЕЛЬ КОТОРЫЙ БУДЕТ ВЫПОЛНЯТЬ ПРОЦЕСС, например:* tandem*}*
numprocs=1
startsecs=0
redirect_stderr=true
environment=HOME="*{ДОМАШНЯЯ ДИРЕКТОРИЯ ИСПОЛНЯЕМОГО ПОЛЬЗОВАТЕЛЯ, например:* /home/tandem*}",USER="{ПОЛЬЗОВАТЕЛЬ КОТОРЫЙ БУДЕТ ВЫПОЛНЯТЬ ПРОЦЕСС, например:* tandem*}",PATH=/usr/local/bin:%(ENV_PATH)s*
stdout_logfile=*{ПУТЬ ДЛЯ ХРАНЕНИЯ ЛОГОВ, например:* /home/tandem*}/lk-base-system-worker.log*
stopwaitsecs=3600

- Сохранить файл system-base.conf
- Создать конфигурационный файл для запуска очереди

уведомлений: nano /etc/supervisor/conf.d/event-notification-worker.conf со следующим примерным содержимым:

[program:*{НАЗВАНИЕ, например:* event-notification-worker*}***]**
process_name=%(program_name)s_%(process_num)02d
command=php *{ПУТЬ К ПРОЕТУ, например:* /var/www/html*}***/artisan queue:work --queue=event_notification**
autostart=true
autorestart=true
stopasgroup=true
killasgroup=true
user=www-data
numprocs=1
startsecs=0
redirect_stderr=true
stdout_logfile=*{ПУТЬ ДЛЯ ХРАНЕНИЯ ЛОГОВ, например:* /home/tandem*}***/event_notification.log**
stopwaitsecs=3600

- Сохранить файл event-notification-worker.conf
- Создать 2-й конфигурационный файл для запуска очереди уведомлений: nano /etc/supervisor/conf.d/delivery-notification-worker.conf со следующим примерным содержимым:

[program:*{НАЗВАНИЕ, например:* delivery-notification-worker*}***]**
process_name=%(program_name)s_%(process_num)02d
command=php *{ПУТЬ К ПРОЕТУ, например:* /var/www/html*}***/artisan queue:work --queue=delivery_notification**
autostart=true
autorestart=true
stopasgroup=true
killasgroup=true
user=www-data
numprocs=1
startsecs=0
redirect_stderr=true
stdout_logfile=*{ПУТЬ ДЛЯ ХРАНЕНИЯ ЛОГОВ, например:* /home/tandem*}***/delivery_notification.log**
stopwaitsecs=3600

- Сохранить файл delivery-notification-worker.conf
- Проверить файлы конфигураций supervisor:

```
sudo supervisorctl reread
```

- Обновить процессы и запустить новый процесс:

```
sudo supervisorctl update && sudo supervisorctl start {НАЗВАНИЕ, например: lk-base}:* && sudo supervisorctl start {НАЗВАНИЕ, например: lk-system-base}:* && sudo supervisorctl start {НАЗВАНИЕ, например: event-notification-worker}:* && sudo supervisorctl start {НАЗВАНИЕ, например: delivery-notification-worker}:*
```

Пример: sudo supervisorctl update && sudo supervisorctl start lk-base:* && sudo supervisorctl start lk-system-base:* && sudo supervisorctl start event-notification-worker:* && sudo supervisorctl start delivery-notification-worker:*

- Важно! Проверить, что функция exec в php включена.

На этом процесс донастройки модуля tandem/base завершен – очередь будет автоматически браться в обработку.

Настройка планировщика задач

Для выполнения команд получения, актуализации данных из интегрируемых систем и выполнения других системных действий по расписанию на сервере ЛК необходимо в планировщике задач (cron) прописать следующую команду:

```
* * * * * cd /{путь до корня проекта} && php artisan schedule:run >> /dev/null 2>&1
```

Например: * * * * * cd /var/www/html/lk && php artisan schedule:run >> /dev/null 2>&1

Задача cron должна создаваться под тем пользователем, под которым работает приложение (не под root, например, tandem).

Добавить в crontab задачу на очистку логов (по умолчанию очищаются все логи старше 30 дней):

```
php artisan tandem-notification:log-clear [--days=30]
```

Настроить сброс неудавшихся очередей (например, более старые, чем 3 дня):

```
php artisan queue:prune-failed --hours=72
```