

ИНСТРУКЦИЯ

по установке программного обеспечением

«Телеформ»

АННОТАЦИЯ

Данный документ содержит в себе Инструкцию по установке серверной части программного обеспечения «Телеформ».

Описана последовательность настройки фронтэнда и бэкенда ПО.

ОГЛАВЛЕНИЕ

1.	УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	4
1.1.	Основные пакеты, используемые в проекте	4
1.2.	Установка ПО.....	4
1.3.	Создание базы данных	4
1.4.	Настройка бэкенда.....	5
1.1.	Запуск фронтенда	5

1. УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Установка и настройка всех необходимых модулей будет рассматриваться на примере Ubuntu Server 20.04.4 Для других ОС можно сделать по аналогии, посмотрев документацию соответствующих пакетов.

1.1. Основные пакеты, используемые в проекте

1.1.1. В программном обеспечении используются следующие пакеты:

- nginx;
- php (cli для фоновых задач и fpm для работы апи);
- ionCube loader;
- memcached;
- postgresql (<https://www.postgresql.org/download/linux/ubuntu/>);
- nodejs;
- yarn;
- docker-ce;
- ffmpeg;

1.2. Установка ПО

1.2.1. Установка всех необходимых пакетов производится скриптом `install.sh`.

1.2.2. Создаем пользователя, под которым будет развернут сайт и от имени которого будут запускать скрипты, как из `php-fpm`, так и из `cron`. запускаем скрипт `user.sh`.

1.2.3. Этому пользователю необходимо будет прописать в настройки `php-fpm` в файле скриптом `php.ini.sh`.

1.3. Создание базы данных

1.3.1. Теперь необходимо создать базу данных для приложения. Для этого выполняем команды:

```
sudo su - postgres
psql
CREATE DATABASE teleform app;
CREATE USER teleform app WITH password 'password_for_database';
GRANT ALL ON DATABASE teleform app TO teleform app;
```

1.3.2. Из архива `teleform.tar.gz` берем файл `ioncube_loader_lin_7.2.so` и копируем его в `/usr/lib/php/20170718/`, после чего создаем файлы `/etc/php/7.2/fpm/conf.d/00-ioncube.ini` и `/etc/php/7.2/cli/conf.d/00-ioncube.ini` со следующим содержимым:

```
zend_extension = /usr/lib/php/20170718/ioncube_loader_lin_7.2.so.
```

1.3.3. Далее папки `backend` и `frontend` из архива помещаем в `/var/www` и по желанию переименовываем, а также копируем файл лицензии из архива в папку `backend`, например, `backend` в `api.host.ru`, а `frontend` в `host.ru` и при необходимости меняем владельца

```
sudo chown -R web-user:www-data /var/www/
```

1.3.4. В каждой из папок есть пример конфигурации для `nginx`, вносим в них необходимые изменения - пути к сертификатам, нужные домены и т.д. Создаем папку под кэш (первая строчка в `backend/nginx.conf`):

```
sudo mkdir -p /var/www/cache/teleform.
```

1.4. Настройка бэкенда

1.4.1. Теперь необходимо внести все необходимые настройки в файл `backend/.env` (ср `/var/www/backend/.env.example /var/www/backend/.env`) указываем данные для админа системы, если оставить пустыми то будет `email=admin@teleform.ru password=password` (что не безопасно), настройки соединения с базой данных, при необходимости с smtp сервером, домен апи и домен фронтэнда.

1.4.2. Так же вносим настройки в `frontend/.env.production.local` - домен апи и домен, так же можно включить или выключить некоторые функции или подключить `logrocket`.

1.4.3. После этого запускаем установку бэкенда, все команды установки выполняются из под созданного пользователя `web-user`:

```
cd /var/www/backend
php artisan migrate
php artisan key:generate --ansi
php artisan passport:install
php artisan optimize.
```

1.4.4. Теперь нужно запустить выполнение задач по расписанию, для этого, под тем же пользователем выполняем команду:

```
crontab -e
```

1.4.5. Теперь нужно запустить выполнение задач по расписанию, для этого, под тем же пользователем выполняем команду:

```
crontab -e
```

и добавляем строчку

```
* * * * * php /var/www/backend/artisan
schedule:run >> /dev/null 2>&1
```

1.1. Запуск фронтенда

1.4.6. Запускаем фронтенд:

```
zcat teleform_test_frontend.gz | docker import - teleform_frontend_test
docker run -d -p 8062:80 --restart=always --name teleform_frontend_test -w
/app --add-host=api.teleform.app:172.17.0.1 teleform_frontend_test yarn
start:prod -p 80
```

1.4.7. Поскольку менялись конфигурации для вебсерверов (`php-fpm` и `nginx`) необходимо их перезапустить:

```
sudo service php-fpm restart
sudo service nginx restart
```