

Настройка сервера, установка и первый запуск

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.

Основные пакеты, используемые в проекте:

- 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

Установка всех необходимых пакетов производится скриптом install.sh

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
Этого пользователя необходимо будет прописать в настройки php-fpm в файле скриптом php.ini.sh

Теперь необходимо создать базу данных для приложения:

```
sudo su – postgres
```

```
psql
```

```
CREATE DATABASE univey_app;
```

```
CREATE USER univey_app WITH password 'password_for_database';
```

```
GRANT ALL ON DATABASE univey_app TO univey_app;
```

Из архива (<https://disk.yandex.ru/d/hbigPshOkq4sHQ>) formalist.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 содержанием:

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

Далее папки backend и frontend из архива помещаем в /var/www и по желанию переименовываем, а так же 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/
```

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/univey
```

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

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

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
```

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
```

И запускаем фронтэнд:

```
zcat univey_test_frontend_mangazeya.gz | docker import -
univey_frontend_test

docker run -d -p 8062:80 --restart=always --name univey_frontend_test -w
/app --add-host=api.formalist.app:172.17.0.1 univey_frontend_test yarn start:prod -
p 80
```

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

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