

Информационная система  
«Комиссия по делам несовершеннолетних и  
защите их прав»

**Инструкция по установке**

Версия 0.1.0

## Оглавление

|                                                                                  |    |
|----------------------------------------------------------------------------------|----|
| Введение .....                                                                   | 3  |
| Создание баз данных .....                                                        | 3  |
| Установка OpenSSL (под пользователем <code>root</code> ) .....                   | 4  |
| Установка интерпретатора Python (под пользователем <code>root</code> ) .....     | 4  |
| Пользователь <code>www-data</code> .....                                         | 5  |
| Создание виртуального окружения (под пользователем <code>www-data</code> ) ..... | 5  |
| Каталог <code>/var/www/childcomission</code> .....                               | 6  |
| Конфигурационные файлы приложения .....                                          | 6  |
| Установка приложения .....                                                       | 7  |
| Установка дистрибутива .....                                                     | 8  |
| Миграции и сбор статистики .....                                                 | 8  |
| Настройка Supervisor .....                                                       | 8  |
| Запуск Supervisor .....                                                          | 9  |
| Настройка nginx .....                                                            | 10 |

## Введение

Данная инструкция описывает процесс развёртывания Подсистемы «Регион КДНиЗП» (далее — «Приложение») в операционной системе Debian версии Buster (далее — «ОС», «система»). При использовании других ОС некоторые этапы установки могут отличаться.

Компьютер должен быть подключён к интернету. В ОС должны быть установлены веб-сервер Nginx и система управления базами данных PostgreSQL, настроен протокол SSH.

Для успешной установки надо, чтобы были открыты следующие сайты:

```
debian.map.fastlydns.net
www.python.org
pypi.org
files.pythonhosted.org
pip.pypa.io
pypi.python.org
pypi.bars-open.ru
postgresql.org
apt.postgresql.org
www.postgresql.org
nginx.org
bootstrap.pypa.io
www.openssl.org
bars-czm.ru
```

## Создание баз данных

В PostgreSQL должен быть создан пользователь, которому будут принадлежать права работы с базами данных Приложения. Имя пользователя и пароль используются в файле настройки Приложения **childcomission.conf** для соединения с базами данных.

Для работы Приложения необходимо создать две базы данных в PostgreSQL: базу для основных данных и базу для служебной информации (логов действий пользователей). Владельцем этих баз данных будет вновь созданный пользователь PostgreSQL. Рекомендуется создавать две отдельные базы, чтобы логи действий пользователей хранились отдельно от основной информации. Имена баз данных могут быть любыми; эти имена необходимо указать в файле настроек Приложения **childcomission.conf** в разделах **[database]** и **[service\_database]**:

```
[database]
...
DATABASE_ENGINE = django.db.backends.postgresql_psycopg2
DATABASE_NAME = childcomission
DATABASE_USER = bars
DATABASE_PASSWORD = bars
...
[service_database]
```

```
ENGINE = django.db.backends.postgresql_psycopg2
NAME = childcomission_service
USER = bars
PASSWORD = bars
```

Здесь:

- **DATABASE\_ENGINE, ENGINE** — система управления базами данных;
- **DATABASE\_NAME, NAME** — название основной и сервисной баз данных;
- **DATABASE\_USER, USER** — пользователь основной и сервисной баз данных;
- **DATABASE\_PASSWORD, PASSWORD** — пароль пользователя основной и сервисной баз данных;

Для создания баз данных запускаем интерпретатор **psql**:

```
sudo su postgres -c psql
```

Создаём пользователя bars, если он не зарегистрирован в PostgreSQL, и базы данных childcomission и childcomission\_service:

```
CREATE ROLE bars WITH PASSWORD 'bars' LOGIN;
CREATE DATABASE childcomission WITH OWNER=bars;
CREATE DATABASE childcomission_service WITH OWNER=bars;
```

После этого можно выйти из интерпретатора psql.

## Установка OpenSSL (под пользователем root)

```
sudo -i
apt-get update
apt-get install curl git wget -y
apt-get install build-essential -y
apt-get install software-properties-common -y
curl https://www.openssl.org/source/openssl-1.1.1g.tar.gz --output openssl-1.1.1g.tar.gz
| tar xz && cd openssl-1.1.1g && ./config shared --prefix=/usr/local/ && make && make
install
```

## Установка интерпретатора Python (под пользователем root)

```
apt-get update
apt-get install libreadline-gplv2-dev libncursesw5-dev -y
apt-get install libgdbm-dev libc6-dev libbz2-dev zlib1g-dev liblcms2-dev zlibc -y
apt-get install libfreetype6-dev libjpeg62-turbo-dev libwebp-dev libpq-dev libxml2-dev
libxmlsec1-dev -y
apt-get install python3-pip python3-setuptools python3-virtualenv virtualenvwrapper
python-elementtidy -y
apt-get install python3-dev libpython3-dev libffi-dev python-smbus libsasl2-dev libldap2-
dev -y
apt-get install libncursesw5-dev libgdbm-dev libc6-dev tk-dev libssl-dev swig -y
apt-get install python3-celery python-celery-common supervisor default-jre redis-server -
y
```

Кроме этих пакетов для работы некоторых плагинов Приложения может потребоваться установить другие пакеты. Для уточнения обратитесь к документации по плагинам.

После установки пакетов следует собрать и установить Python 3.9 из исходников:

```
export LDFLAGS="-L/usr/local/openssl/lib/"
export LD_LIBRARY_PATH="/usr/local/openssl/lib/"
export CPPFLAGS="-I/usr/local/include -I/usr/local/include/openssl"

cd /usr/src/
wget https://www.python.org/ftp/python/3.9.13/Python-3.9.13.tgz
tar xzf Python-3.9.13.tgz
cd Python-3.9.13
./configure --prefix=/usr/local/
sudo make altinstall
```

### Проверка версии питона и ssl:

```
python3.9
>>> import ssl
>>> ssl.OPENSSL_VERSION
'OpenSSL 1.1.1g 21 Apr 2020'
>>> exit()
```

### Установка пакетов в Python:

```
python3.9 -m pip install --upgrade pip wheel virtualenv virtualenvwrapper
```

## Пользователь **www-data**

Работа веб-сервера **nginx** и запуск Приложения должны производиться от имени ограниченного в правах пользователя **www-data**. Для выполнения действий от имени этого пользователя, нужно активировать его командой:

```
sudo su www-data -s /bin/bash
```

## Создание виртуального окружения (под пользователем **www-data**)

Виртуальное окружение для запуска приложений должно создаваться от имени пользователя **www-data**. Созданные виртуальные окружения находятся в папке для виртуальных окружений в домашнем каталоге пользователя **www-data** — **/var/www/.virtualenvs**.

### Создание виртуального окружения **childcomission**:

```
source /usr/local/bin/virtualenvwrapper.sh
```

последнюю команду можно добавить в **~/ .bashrc**

```
python3.9 -m virtualenv ~/.virtualenvs/childcomission
```

Чтобы активировать виртуальное окружение, необходимо выполнить команду:

```
workon childcomission
```

Перед командной строкой появится в скобках название активного виртуального окружения.

Для выхода из активного виртуального окружения используется команда:

```
deactivate
```

**ВАЖНО.** Каждый пользователь имеет свой список виртуальных окружений; для правильной работы создание и активация виртуального окружения должны проводиться от имени пользователя **www-data**.

## Каталог **/var/www/childcomission**

Для хранения настроек, логов, файлов статики и медиа для Приложения используется каталог **/var/www/childcomission**. Следует создать такую структуру каталогов и файлов:

```
/var/www/childcomission
├── conf
│   ├── gunicorn.conf.py
│   ├── childcomission.conf
│   └── plugins.conf
├── log
├── media
├── static
├── var
│   └── run
```

Для создания такой структуры используются команды (запускать от имени пользователя **www-data**):

```
mkdir -p /var/www/childcomission/conf
mkdir -p /var/www/childcomission/static
mkdir -p /var/www/childcomission/media
mkdir -p /var/www/childcomission/log
mkdir -p /var/www/childcomission/var/run
sudo chmod 750 -R /var/www/childcomission/log
sudo chmod 750 -R /var/www/childcomission/var/run
cp ~/repo/childcomission/src/childcomission/childcomission.conf.default
/var/www/childcomission/conf/childcomission.conf
cp ~/repo/childcomission/src/childcomission/plugins.conf.default
/var/www/childcomission/conf/plugins.conf
cp ~/repo/childcomission/src/childcomission/names.conf.default
/var/www/childcomission/conf/names.conf
cp ~/repo/childcomission/src/childcomission/desktop.conf.default
/var/www/childcomission/conf/desktop.conf
```

## Конфигурационные файлы приложения

### Файл **/var/www/childcomission/conf/childcomission.conf**

```
[database]
DATABASE_USER = bars
DATABASE_PASSWORD = bars

[service_database]
USER = bars
PASSWORD = bars

[django-axes]
AXES_LOCK_OUT_AT_FAILURE = True

[celery]
USE_CELERY = True

[paths]
PATH_TO_STATIC = /var/www/childcomission/static
PATH_TO_MEDIA = /var/www/childcomission/media
```

### Файл `/var/www/childcomission/conf/gunicorn.conf.py`

Данный файл – скрипт запуска приложения. Именно его будет выполнять **supervisor** при старте приложения. Содержимое этого файла:

```
# -*- coding: utf-8 -*-
import multiprocessing
import os

os.environ['LD_FLAGS'] = '-L/usr/local/openssl/lib/'
os.environ['LD_LIBRARY_PATH'] = '/usr/local/openssl/lib/'
os.environ['CPPFLAGS'] = '-I/usr/local/include -I/usr/local/include/openssl'
os.environ['DJANGO_SETTINGS_MODULE'] = 'childcomission.settings'
os.environ['CHILDCOMISSION_CONFIG_PATH'] = '/var/www/childcomission/conf/'
bind = 'unix:/var/www/childcomission/var/run/childcomission.sock'
workers = multiprocessing.cpu_count()
user = 'www-data'
group = 'www-data'
errorlog = '/var/www/childcomission/log/gunicorn.log'
loglevel = 'error'
timeout = 180
```

Здесь указывается, какой файл настроек использовать, где находится каталог настроек нашего приложения. Далее указывается способ связи с приложением — через сокеты `unix`. Чуть ниже — имя процесса.

После создания данного файла следует разрешить его выполнение:

```
chmod u+x /var/www/childcomission/conf/gunicorn.conf.py
```

### Файл `/var/www/childcomission/conf/childcomission.conf`

Данный файл хранит в себе настройки приложения. Здесь указываются параметры подключения к базам данных, время жизни сессий и т.д. Как правило, у настройки указан комментарий, чтобы можно было понять, какие параметры использовать в конкретной ситуации.

Исходный файл **childcomission.conf** можно взять в каталоге приложения, там он лежит под именем **childcomission.conf.default**. Рекомендуется быть **КРАЙНЕ** внимательным при заполнении данного файла конфигурации.

### Файл `/var/www/childcomission/conf/plugins.conf`

Список подключенных плагинов. Как правило, подходит для использования поставляемый с приложением файл **plugins.conf.default**, переименованный и положенный в данный каталог.

## Установка приложения

Авторизуйтесь от имени пользователя **www-data** и создайте виртуальное окружение, если оно ещё не создано, так, как это описано выше. Активируйте его. Все дальнейшие действия выполняются в активном виртуальном окружении.

Выполните команды:

```
export LDFLAGS="-L/usr/local/openssl/lib/"
export LD_LIBRARY_PATH="/usr/local/openssl/lib/"
export CPPFLAGS="-I/usr/local/include -I/usr/local/include/openssl"
export DJANGO_SETTINGS_MODULE='childcomission.settings'
export CHILDCOMISSION_CONFIG_PATH='/var/www/childcomission/conf/'
```

## Установка дистрибутива

Выполните команды установки дистрибутива (файл `childcomission.2022.007.xx.tar`, где `xx` — номер релиза:

```
pip install --upgrade --extra-index-url http://pypi.bars-open.ru/simple/ --trusted-host
pypi.bars-open.ru m3_builder
pip install --upgrade virtualenvwrapper
pip install --upgrade --extra-index-url http://pypi.bars-open.ru/simple/ --trusted-host
pypi.bars-open.ru childcomission.2022.007.xx.tar
```

Установите сервер **gunicorn**, который не входит в список зависимостей приложения и не указан в файле **REQUIREMENTS**, но необходим для запуска нашего приложения<sup>1</sup>:

```
pip install gunicorn
```

## Миграции и сбор статики

Выполните команду:

```
django-admin update --noinput
```

Эта команда синхронизирует приложение с базами данных, проведёт по ним миграции и соберёт статику. При выполнении этой команды вам будет предложено создать суперпользователя. Вы можете согласиться, или создать суперпользователя при первом запуске подсистемы.

Самый долгий процесс – миграции. Обязательно дождитесь его завершения, т.к. в противном случае данные могут быть повреждены либо приложение будет работать неправильно!

## Настройка Supervisor

Настройки для supervisor хранятся в каталоге **/etc/supervisor**. В нём содержится файл **supervisord.conf** и каталог **conf.d**, в который следует помещать настройки для приложений. Нужно под пользователем **root** перейти в этот каталог и создать файл настроек:

```
sudo -i
cd /etc/supervisor/conf.d/
touch childcomission.conf
```

---

<sup>1</sup> Вместо Gunicorn вы можете использовать uWSGI. Настройки для работы с uWSGI здесь не рассматриваются.

## Файл /etc/supervisor/conf.d/childcomission.conf:

```
[program:childcomission]
environment=CHILDCOMMISSION_CONFIG_PATH="/var/www/childcomission/conf/",DJANGO_SETTINGS_MODULE="childcomission.settings",LANG="ru_RU.UTF-8",LC_ALL="ru_RU.UTF-8",LC_LANG="ru_RU.UTF-8",PYTHONIOENCODING="UTF-8"
command=/var/www/.virtualenvs/childcomission/bin/gunicorn childcomission.wsgi:application -c /var/www/childcomission/conf/gunicorn.conf.py
user=www-data
group=www-data
daemon=false
debug=false
autostart=true
autorestart=true
redirect_stderr=true
stdout_logfile=/var/www/childcomission/log/supervisor.log

[program:childcomission_celery]
environment=CHILDCOMMISSION_CONFIG_PATH='/var/www/childcomission/conf/',DJANGO_SETTINGS_MODULE='childcomission.settings',PYTHONIOENCODING=UTF-8,LANG=ru_RU.UTF-8,LC_ALL=ru_RU.UTF-8,LC_LANG=ru_RU.UTF-8,PATH='/usr/local/openssl/bin:/usr/local/bin:/usr/bin:/bin',OPENSSL_CONF='/usr/local/openssl/openssl.cnf',
command=/var/www/.virtualenvs/childcomission/bin/celery worker -A childcomission.celery_app -l debug
#пользователь и группа от чьего имени будет запускаться приложение
user=www-data
group=www-data
daemon=False
debug=False
stopasgroup = True
autostart=true
autorestart=true
redirect_stderr=True
stdout_logfile=/var/www/childcomission/log/celery.log

[program:childcomission_celery_beat]
environment=CHILDCOMMISSION_CONFIG_PATH='/var/www/childcomission/conf/',DJANGO_SETTINGS_MODULE='childcomission.settings',PYTHONIOENCODING=UTF-8,LANG=ru_RU.UTF-8,LC_ALL=ru_RU.UTF-8,LC_LANG=ru_RU.UTF-8,PATH='/usr/local/openssl/bin:/usr/local/bin:/usr/bin:/bin',OPENSSL_CONF='/usr/local/openssl/openssl.cnf',
command=/var/www/.virtualenvs/childcomission/bin/celery beat -A childcomission.celery_app -l debug --pidfile /var/www/childcomission/var/run/celerybeat.pid -s /var/www/childcomission/var/run/celerybeat-schedule
#пользователь и группа от чьего имени будет запускаться приложение
user=www-data
group=www-data
daemon=False
debug=False
stopasgroup = True
autostart=true
autorestart=true
redirect_stderr=True
stdout_logfile=/var/www/childcomission/log/celery_beat.log
```

## Запуск Supervisor

Теперь приложение можно запустить с помощью Supervisor. Для этого выполните следующую команду от имени **root**:

```
supervisorctl update
```

Эту же команду следует выполнить при любом изменении в файлах конфигурации приложений, расположенных в каталоге **/etc/supervisor/conf.d/**.

Управление приложениями с помощью Supervisor производится с помощью утилиты **supervisorctl** с различными параметрами:

Табл. 1.

### Параметры supervisorctl

| Команда                                           | Результат                                                                       |
|---------------------------------------------------|---------------------------------------------------------------------------------|
| <code>supervisorctl stop childcomission</code>    | Останавливает приложение childcomission                                         |
| <code>supervisorctl stop all</code>               | Останавливает все приложения                                                    |
| <code>supervisorctl start childcomission</code>   | Запускает приложение childcomission                                             |
| <code>supervisorctl start all</code>              | Запускает все приложения                                                        |
| <code>supervisorctl restart childcomission</code> | Перезапускает приложение childcomission                                         |
| <code>supervisorctl restart all</code>            | Перезапускает все приложения                                                    |
| <code>supervisorctl status childcomission</code>  | Выводит информацию о том, в каком состоянии находится приложение childcomission |
| <code>supervisorctl status</code>                 | Выводит информацию о состоянии всех приложений.                                 |

Если приложение успешно запущено (статус приложения *RUNNING*), можно приступить к настройке nginx.

### Настройка nginx

Создайте в каталоге `/etc/nginx/` файл **proxy\_params.conf** со следующим содержанием:

```
proxy_redirect                off;
proxy_set_header Host         $http_host;
proxy_set_header X-Real-IP    $remote_addr:$remote_port;
proxy_set_header X-Url-Scheme $scheme;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
client_max_body_size          32m;
client_body_buffer_size        2m;
proxy_buffering                off;
proxy_send_timeout             180s;
proxy_read_timeout             180s;
proxy_connect_timeout          180s;
proxy_buffer_size              8k;
proxy_buffers                  32 32k;
proxy_busy_buffers_size         64k;
proxy_temp_file_write_size     16m;
add_header X-Frame-Options "SAMEORIGIN";
```

Исходя из мощности сервера, следует установить размеры буферов, тела запросов и т.д.

После этого сделайте резервную копию файла **nginx.conf**, а оригинал приведите к нужному виду:

```
cd /etc/nginx
cp nginx.conf nginx.conf.bak
```

### Файл /etc/nginx/nginx.conf:

```
user nginx;
worker_processes auto;    # Количество воркеров вычисляется автоматически
worker_priority -5;       # Увеличиваем приоритет процесса
worker_rlimit_nofile 2048; # Лимит на число воркеров

error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;

events {
    worker_connections 2048;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer" '
                    '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main; # Настройка основного лога

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;

    keepalive_timeout 180; # Три минуты поддерживать подключение

    client_max_body_size 32m; # Максимальное тело запроса - 32 Мб
    server_tokens off;

    include /etc/nginx/conf.d/*.conf;
}
```

Теперь перейдите в каталог **/etc/nginx/conf.d/** и создайте файл **childcomission.conf**:

```
cd /etc/nginx/conf.d/
touch childcomission.conf
```

### Файл /etc/nginx/conf.d/childcomission.conf:

```
upstream childcomission {
    server unix:/var/www/childcomission/var/run/childcomission.sock fail_timeout=0;
    keepalive 30;
}

server {
    listen 80; # Слушать 80 порт
    server_name 10.252.9.34 childcomission.ru www.childcomission.ru;
    # Откликаться на имена www.childcomission.ru и childcomission.ru
    keepalive_timeout 10;
    access_log off;
}
```

```

error_log /var/www/childcomission/log/nginx_error.log crit;

location / {
    proxy_pass http://childcomission/;
    include /etc/nginx/proxy_params.conf;
}

location /static/ {
    expires 3d;
    alias /var/www/childcomission/static/;
    error_log /var/www/childcomission/log/nginx_static_errors.log;
}

location /media/ {
    expires 3d;
    alias /var/www/childcomission/media/;
    error_log /var/www/childcomission/log/nginx_media_errors.log;
}

location ~* \.(7z|jpg|jpeg|gif|png|ico|css|bmp|swf|js|html|txt|doc|docx|pdf|rar|xls|xlsx|zip)$ {
    root /var/www/childcomission/;
    expires 3d;
    add_header Cache-Control: public;
    access_log off;
    error_log /var/www/childcomission/log/nginx_static_error.log;
}
}

```

В параметре `server_name` указывается IP-адрес и доменное имя, на которое обращается пользователь для доступа к системе.

Когда настройка будет закончена, перезапустите `nginx`:

```
service nginx restart
```

На этом настройка закончена. Если всё сделано правильно, то Приложение будет доступно по тем адресам, что Вы указали в **`/etc/nginx/conf.d/childcomission.conf`** в параметре `server_name` (эти адреса должны быть прописаны в DNS):  
**<http://childcomission.ru>**, **<http://www.childcomission.ru>**.