

Документация по деплою серверной части проекта

Предполагается, что ОС сервера, на который происходит деплой проекта, - **Linux Debian 6.1.+**. Версия Python для запуска проекта - **3.10.+**, PostgreSQL - **15.+**.

Клонирование репозитория проекта с github [↗](#)

На выбранной машине выберите желаемую область, куда желаете скопировать репозиторий проекта и пропишите:

```
1 git clone https://github.com/RHDV-Freelance/CRM-Rovesnik-Doroshka-Screpka.git
2 # https-ссылка варьируется от места к месту, откуда происходит её копирование
```

Установка python и postgresql [↗](#)

Установка python [↗](#)

Выполняйте команды в порядке их следования для установки python3.10:

```
1 1. sudo apt update
2 2. sudo apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev libssl-dev libreadline-dev
3 3. wget https://www.python.org/ftp/python/3.10.6/Python-3.10.6.tgz
4 4. tar -xvf Python-3.10.6.tgz
5
6 5. cd Python-3.10.6
7 6. sudo ./configure --enable-optimizations
8
9 # для проверки правильности установки пакетов пропишите следующее
10 *7. python3.10 --version
11 # вывод должен быть следующим
12 Python 3.10.6
```

Установка и конфигурация PostgreSQL [↗](#)

Установка пакетов postgresql:

```
1 1. sudo apt update
2 2. sudo apt install postgresql-client
3
4 # для проверки правильности установки пакетов пропишите следующее
5 *3. psql --version
6 # вывод должен быть следующим
7 psql (PostgreSQL) 15.6 (Debian 15.6-0+deb12u1)
```

Конфигурация postgresql, создание базы данных и юзера, имеющего права на неё:

```
1 # для конфигурации зайдем в командную строку psql
2 1. sudo -u postgres psql # postgres - суперюзер, создающийся по умолчанию на Linux системах
3
4 # создадим юзера с именем dev и паролем 1234, который и будет владеть базой данных и через которого будет происх
5 2. CREATE USER dev WITH ENCRYPTED PASSWORD '1234';
6
7 # создадим базу данных db
8 3. CREATE DATABASE db;
```

```

9
10 # подключимся к только что созданной базе данных db
11 4. \c db
12
13 # выдадим все права на взаимодействие с базой данных юзеру dev и любые действия с public схемой
14 5. GRANT ALL PRIVILEGES ON DATABASE db TO dev;
15 6. GRANT ALL ON SCHEMA public TO dev;
16
17 # выйдем из командной строки, конфигурация закончена
18 7. \q

```

Установка виртуальной среды [↗](#)

Чтобы создать виртуальную среду для проекта, нужно установить соответствующую библиотеку в python:

```

1 python3.10 -m pip install python-dotenv

```

После установки пропишем команду:

```

1 python3.10 -m venv venv

```

Чтобы активировать среду и установить в нее необходимые библиотеки нужно прописать:

```

1 1. source venv/bin/activate
2 2. python3.10 -m pip install -r requirements.txt

```

Конфигурация .env [↗](#)

Для создания файла окружения пропишите:

```

1 touch .env

```

Вставьте в него `.env-example` из репозитория и заполните переменные в соответствии с их значением:

```

1 # .env-example
2 HEAD_TELEGRAM_BOT_TOKEN=
3 HEAD_BOT_PAYMENT_TOKEN=
4 HEAD_BOT_NAME=
5
6 ROVESNIK_TELEGRAM_BOT_TOKEN=
7 ROVESNIK_PAYMENT_TOKEN=
8 ROVESNIK_BOT_NAME=
9
10 SKREPKA_TELEGRAM_BOT_TOKEN=
11 SKREPKA_PAYMENT_TOKEN=
12 SKREPKA_BOT_NAME=
13
14 DOROSHKHA_TELEGRAM_BOT_TOKEN=
15 DOROSHKHA_PAYMENT_TOKEN=
16 DOROSHKHA_BOT_NAME=
17
18 # IIKO LOGIN
19 API_LOGIN=
20
21 # DATABASE CONFIG (значения подставлены по умолчанию для данного мануала)
22 POSTGRES_USER=dev
23 POSTGRES_PASSWORD=1234
24 POSTGRES_DB=db

```

```

25 POSTGRES_HOST=localhost
26 POSTGRES_PORT=5432
27
28 # PGADMIN CONFIG
29 PGADMIN_DEFAULT_EMAIL=root@root.com
30 PGADMIN_DEFAULT_PASSWORD=root
31
32 # TINKOFF TERMINAL
33 TERMINAL_NAME=
34 TERMINAL_PASSWORD=
35
36 # ADMIN CHAT ID
37 SUPPORT_ADMIN_ID=
38
39 # REVIEWS GROUP CHAT ID
40 REVIEW_GROUP_CHAT_ID=
41
42 JWT_SECRET=

```

Конфигурация `alembic.ini`, `env.py` и создание миграций [↗](#)

Перейдем в папку проекта `BackendApp/Database` и инициализируем миграции:

```

1 # используется асинхронный драйвер для взаимодействия с базой данных, поэтому прописываем флаг -t async
2 alembic init -t async migrations

```

Настроим конфигурацию файлов `alembic.ini` и `env.py`:

```

1 # alembic.ini
2 # доходим до 63 строки в файле, где написана переменная sqlalchemy.url и меняем её соответственно
3 1. sqlalchemy.url = postgresql+asyncpg://dev:1234@127.0.0.1:5432/db
4
5         ↑↑↑↑↑↑↑      ↑↑↑  ↑↑↑  ↑↑↑↑  ↑↑↑↑ ↑↑
6         asynchronous-driver  user  pass-   IP    port dbname
7                                word
7 # тут дан пример в соответствии с конфигурацией базы данных по умолчанию в мануале
8
9 # env.py
10 1. Прописать в самом начале файла import from BackendApp.Models import Base
11 2. Дойти до строки с metadata, поменять значение None -> Base.metadata

```

Создадим миграции в базе данных:

```

1 alembic revision --autogenerate
2 alembic upgrade head

```

Запуск ботов, шедулера и API [↗](#)

Конфигурация запуска `.py` скриптов в разных областях будет производиться с помощью утилиты `screen`. Для её установки пропишем:

```

1 sudo apt update
2 sudo apt install screen

```

```

1 screen -S <screen_name> - создание скрина с screen_name
2 screen -x <screen_name> - присоединение к скрину с screen_name
3 screen -XS <screen_name> quit - удаление скрина
4 screen -ls - просмотр всех созданных скринов

```

```
5
6 Сочетание клавиш Ctrl + A + D - выход из скрина
7 Сочетание клавиш Ctrl + A + [ - вход в copy-mode
8 Сочетание клавиш Ctrl + A + ] - выход из copy-mode
```

Запуск ботов: [↗](#)

```
1 1. screen -S head_bot
2 2. source venv/bin/activate
3 3. export PYTHONPATH=$PYTHONPATH:$(pwd)
4 4. python3.10 BackendApp/TelegramBots/HeadBot/main.py
5 5. Ctrl + A + D
6
7 1. screen -S rovesnik_bot
8 2. source venv/bin/activate
9 3. export PYTHONPATH=$PYTHONPATH:$(pwd)
10 4. python3.10 BackendApp/TelegramBots/Rovesnik/main.py
11 5. Ctrl + A + D
12
13 1. screen -S skrepka_bot
14 2. source venv/bin/activate
15 3. export PYTHONPATH=$PYTHONPATH:$(pwd)
16 4. python3.10 BackendApp/TelegramBots/Skrepka/main.py
17 5. Ctrl + A + D
18
19 1. screen -S doroshka_bot
20 2. source venv/bin/activate
21 3. export PYTHONPATH=$PYTHONPATH:$(pwd)
22 4. python3.10 BackendApp/TelegramBots/Doroshka/main.py
23 5. Ctrl + A + D
```

Запуск API: [↗](#)

```
1 1. screen -S API
2 2. source venv/bin/activate
3 3. export PYTHONPATH=$PYTHONPATH:$(pwd)
4 4. python3.10 BackendApp/API/app.py
5 5. Ctrl + A + D
```

Запуск шедулера: [↗](#)

```
1 1. screen -S scheduler
2 2. source venv/bin/activate
3 3. export PYTHONPATH=$PYTHONPATH:$(pwd)
4 4. python3.10 BackendApp/TelegramBots/tasks_for_scheduling/main_scheduler.py
5 5. Ctrl + A + D
```