

Локальное развертывание бота

1. Клонирование репозитория

- Клонировать к себе репозиторий https://github.com/Studio-Yandex-Practicum-Hackathons/cafe_azu_bot_1/
- Активировать ветку `release_0_0_2`

2. Установка `poetry`

ПРИМЕЧАНИЕ: Этот пункт больше актуален для разработки. Для развертывания бота локально его можно пока пропустить и вернуться к нему позднее.

Для UNIX-систем и MacOS вводим в консоль следующую команду:

```
curl -sSL https://install.python-poetry.org | python3 -
```

Для WINDOWS PowerShell:

```
(Invoke-WebRequest -Uri https://install.python-poetry.org -  
UseBasicParsing).Content | python -
```

После установки, если установщик попросит, добавить путь к `poetry` в переменную среды `PATH`, перезапустить оболочку и выполнить команду

```
poetry --version
```

Если установка прошла успешно, будет выведена версия `poetry` (может отличаться)

```
Poetry (version 1.6.1)
```

Выполнить команду (что бы виртуальное окружение устанавливалось в папку проекта, а не глобально в систему):

```
poetry config virtualenvs.in-project true
```

3. Установка виртуального окружения

ПРИМЕЧАНИЕ: все дальнейшие пути указываются относительно корневой папки проекта. Т.е. путь вида `/infra/config` означает <папка, куда клонировали репозиторий>/`cafe_azu_bot_1/infra/config`

ЕЩЕ ОДНО ПРИМЕЧАНИЕ: Этот пункт больше актуален для разработки. Для развертывания бота локально его можно пока пропустить и вернуться к нему позднее.

- в терминале переходим в папку `/bot`
- если в системе установлено несколько версий python, то выполняем команду (если одна, то проверяем, что бы это была версия 3.10 и до устанавливаем при необходимости)

```
poetry env use <полный путь к исполняемому файлу python версии 3.10>
```

- устанавливаем виртуальное окружение

```
poetry install --no-root
```

- в папке `/bot` создается папка `.venv`
- переходим в папку `/admin_panel` и повторяем те же действия по установке виртуального окружения

4. Настройка параметров подключения к БД и прочее

В папке `/infra/config` создаем файл `.general` следующего содержания (пример есть в файле `example.general`):

```
# Настройки для бота
BOT_TOKEN=<токен вашего тестового бота>
PAYMENT_PROVIDER_TOKEN=<платежный токен, привязанный к вашему боту>

# Подключение к БД
POSTGRES_DB=azubot
POSTGRES_USER=<имя пользователя для подключения к БД>
POSTGRES_PASSWORD=<пароль пользователя>
POSTGRES_HOST=postgres
POSTGRES_PORT=5432

# Настройки администраторской панели
ENGINE=django.db.backends.postgresql
ALLOWED_HOSTS=django
CSRF_TRUSTED_ORIGINS=http://django:8000
SECRET_KEY=<секретный ключ для джанги>
DEBUG=True
```

5. Поднятие докер-контейнеров

ВАЖНО: При поднятии контейнеров джанги в логах может ругаться, что не выполнена миграция для модели `azucafe`. Это **нормально**. Первичным у нас является бот и именно он создает все

необходимые таблицы в БД для своей работы. Нужные для работы джанги таблицы она создаст сама. Делать миграцию для модели `azucafe` в джанге **НЕ НАДО**.

- переходим в папку `/infra`
- выполняем команду

```
docker compose --file docker-compose_local_dev.yml up --build
```

В результате должна создаться одна группа с 4мя контейнерами:

```
azubot-infra
├─ azubot-infra-django-1      - контейнер с админкой
├─ azubot-infra-telegram_bot-1 - контейнер с ботом
├─ azubot-infra-nginx-1      - контейнер с nginx
└─ azubot-infra-postgres-1   - контейнер с СУБД PostgreSQL
```

- подключаемся к контейнеру `django` (команда `docker compose --file .\docker-compose_local_dev.yml exec -t -i django bash` находясь в папке `/infra`)
- Создаем суперпользователя (`python manage.py createsuperuser`)
- При необходимости загружаем тестовые данные (`python manage.py loaddata base_data.json`)
- выходим из контейнера

Возможные проблемы

Возможно при поднятии контейнеров могут не запуститься контейнеры `django`, `telegram_bot` и `nginx`.

В этом случае проверяем логи этих контейнеров. Если там присутствуют строки

```
/app/dep.sh: line 2: $'\r': command not found
/app/dep.sh: line 13: syntax error: unexpected end of file
```

это значит, что прошел "глюк", связанный с некорректным (для Linux) окончанием строк в файлах `/bot/dep.sh` и `admin_panel/dep.sh`. Вместо юникового конца строки `LF` в них стоит виндовозский конец строки `CRLF`. Нужно его поменять на юниковский `LF` и запустить пересборку контейнеров.