

Инструкция по установке экземпляра программного обеспечения, предоставленного для проведения экспертной проверки Общества с ограниченной ответственностью «Отус онлайн-образование»

Введение.....	1
Docker compose.....	1
Пересборка образа backend.....	1
Миграция БД до текущей версии.....	1
Запуск Django.....	2
Запуск тестов.....	2
Запуск фронта (старый фронт).....	2
Команды фронта.....	2
Запуск фронта (новый фронт).....	2
Node version.....	2
Менеджер пакетов.....	2
Packages.....	3
Запуск.....	3
Команды для dev запуска.....	3
Команды для сборки.....	3
Команды QA.....	4

Введение

Ниже представлена инструкция для локального развертывания экземпляра ПО. Репозитории проекта приложены к заявке в виде архивов.

`otus` - репозиторий back-end и старого фронта

`otus-front` - репозиторий нового фронта

Если у вас возникнут какие-то вопросы по технической части проекта, их можно адресовать нашему разработчику Павлу Глазкову - p.glazkov@otus.ru

Docker compose

В корне проекта находится конфиг для docker compose - `compose.yml`. С его помощью можно билдить и запускать dev-версию проекта со всеми зависимостями - MySQL, RabbitMQ, Elasticsearch, Celery и т.п. Дополнительная документация о работе с docker compose находится здесь: - <https://docs.docker.com/compose/>

Пересборка образа backend

При первом запуске контейнеризованной версии Django, произойдёт автоматически. В дальнейшем, может потребоваться, если изменилось что-то в

файлах, связанных с конфигурацией docker-образа, в проект добавились какие-то зависимости и т.п.

```
docker compose build backend
```

Миграция БД до текущей версии

```
docker compose run backend migrate
```

Запуск Django

```
docker compose up backend
```

Прочие сервисы - mysql, rabbitmq и celery запустятся автоматически.

Запуск тестов

```
docker compose run backend tests
```

Запуск фронта (старый фронт)

1. Установить Node 20
 - для apple Silicon <https://github.com/nvm-sh/nvm#macos-troubleshooting>
 - для windows и linux:
<https://www.geeksforgeeks.org/how-to-install-the-previous-version-of-node-js-and-npm/>
2. `yarn install`
3. `yarn start`

Команды фронта

```
yarn run clean - удаление папки assets
```

```
yarn run clean_sourcemaps - удаление sourcemaps
```

```
yarn start - запуск проекта в development окружении
```

```
yarn run start:with_sourcemaps - запуск проекта в development окружении с sourcemaps
```

```
yarn run build:dev - билд проекта в development окружении
```

```
yarn run build:dev_prod - билд проекта в production окружении
```

```
yarn run build:prod - билд проекта в production окружении с удалением sourcemaps
```

Запуск фронта (новый фронт)

Node version

v20+

Менеджер пакетов

yarn

Packages

external - фронт на Next.js

internal - фронт на React

qa - автотесты

react-static - генерация заглушек в html файлах

shared - общий код

storybook - сторибукком

translation - утилита для переводов

Папка tilda - Скрипты для лендов на тильде

Запуск

yarn - установка модулей.

Так же должно выполняться скрипт prepare - husky install

Если ничего не появилось в папке .husky/_/, то выполнить самостоятельно yarn prepare

Для локальной разработки копируем .env.example - cp .env.example .env.local

LOG_ENABLE - включение логирования

BACK_URL - url до нужно бэка

Команды для dev запуска

yarn dev:external:otus - запуск otus-a external пакета в режиме разработки

yarn dev:external:ketus - запуск _ketus-a_ external пакета в режиме разработки

yarn dev:internal:otus - запуск otus-a internal пакета в режиме разработки

yarn dev:internal:ketus - запуск _ketus-a_ internal пакета в режиме разработки

Команды для сборки

`yarn build:internal:dev` - сборка `internal` пакета для `dev`, с переменной окружения `NEXT_PUBLIC_CONFIG_ENV=development`

`yarn build:internal:prod` - сборка `internal` пакета для `prod`, с переменной окружения `NEXT_PUBLIC_CONFIG_ENV=production`

`yarn build:external:local:otus` - сборка `otus-a external` пакета с локальной конфигурацией `NEXT_PUBLIC_CONFIG_ENV=local` - Для билда локально

`yarn build:external:local:ketus` - сборка `ketus-a external` пакета с локальной конфигурацией `NEXT_PUBLIC_CONFIG_ENV=local` - Для билда локально

`yarn build:external:dev:otus` - сборка `otus-a external` пакета с локальной конфигурацией `NEXT_PUBLIC_CONFIG_ENV=development`

`yarn build:external:dev:ketus` - сборка `ketus-a external` пакета с локальной конфигурацией `NEXT_PUBLIC_CONFIG_ENV=development`

`yarn build:external:prod:otus` - сборка `otus-a external` пакета с локальной конфигурацией `NEXT_PUBLIC_CONFIG_ENV=production`

`yarn build:external:prod:ketus` - сборка `ketus-a external` пакета с локальной конфигурацией `NEXT_PUBLIC_CONFIG_ENV=production`

Команды QA

`yarn qa:cypress:open` - открытие `cypress`

`yarn qa:cypress:run` - запуск тестов `cypress`

`qa:jest:run` - запуск тестов `jest`