

Инструкция по установке экземпляра мобильного приложения, предоставленного для проведения экспертной проверки

Монорепозиторий проекта на базе [NX](#) состоит из нескольких приложений:

- API - Nest graphql backend приложение (используется единый NodeJS бэк для всех приложений системы).
- Client Mobile - мобильное React Native приложение для выпускников.
- Admin UI - web-приложение панели администратора, предназначено для управления пользователями/событиями/публикациями и прочим.
- Client UI - посадочная страница и публичная часть в виде веб-приложения

commitizen friendly

Прежде чем начать. Установка проекта локально

Перед запуском любого из клиентских приложений необходимо развернуть API
Далее разворачиваем клиентские приложения.

Запуск API

Выполняем последовательность команд:

- `npm install` - устанавливаем зависимости,
- `docker-compose up -d` - поднимаем базу данных,
- `npx prisma migrate dev` - выполняем миграции (только для dev-режима разработки!)
- `npm start api` - запускаем graphql api
Дополнительные возможности:
- `prisma/schema.prisma` - изменяет и сохраняет схему
- `npx prisma migrate dev --name=<migration name>` - создать новую миграцию из изменений схемы, обязательно укажите имя для миграции, например `add-user` or `add-order-total-price`

Как перезапустить бэк

Внимание! Приведет к потере данных до стартовых значений миграции и сидов

- `npx prisma migrate reset` - сбросит данные, сгенерирует схему, накатит сиды и выполнит миграции

Как обновить Apollo Studio subgraph

Мы используем Apollo Studio для playground API, периодически нужно синхронизировать схему

- install [Rover CLI](#)
- ```
rover subgraph publish empm@main --schema
./libs/graphql/src/lib/generated/schema.graphql --name api -
```

  
синхронизирует локальную схему с сабграфом Apollo Studio

## Как применить seeds

Seeds это сконфигурированные данные для базы данных для новой установки.

Следуйте шагами 1-3 из [Запуск API](#) Запустите команду:

```
$ npx prisma db seed
```

## Запуск клиентских приложений

### Как начать

Для начала, вы должны скопировать и переименовать `.env.example` to `.env` и скорректировать значения переменных окружения, при необходимости. После этого, вы можете запустить приложение при помощи команды `npm start app-name`, вы можете найти больше информации о старте приложений в `app/project.json`

### Запуск мобильного приложения

Перед началом работы необходимо настроить окружение. Для запуска эмулятора и сборки IOS необходим MAC. Последовательность шагов для настройки окружения, можно увидеть по [ссылке на официальную документацию](#)

### iOs

- Открываем в xCode папку `apps/client-mobile/ios`
- Переходим в директорию iOS и из командной строки запускаем `pod install`
- Скопировать `.env.example` и переименовать в `.env`
- Запускаем команду `nx run client-mobile:run-ios`

### Android

- Скопировать `.env.example` и переименовать в `.env`
- Запускаем команду `nx run client-mobile:run-android`

### Обновление версии мобильного приложения

Вы можете использовать стандартную команду в директории проекта (apps/client-mobile):

- `cd ./apps/client-mobile 2npm version patch -m "Upgrade to %s for reasons"`  
`or npm version prerelease -m "Upgrade to %s for reasons"`

## Сборка

Для сборки Android приложения выполняем команду `client-mobile:build-android` в `empm/nx.json` далее в папке

`apps/client-mobile/android/app/build/outputs/bundle/release/app-release.aab` появится скомпилированный файл для отправки в стор. Внимание! Необходимо помнить о значениях переменных окружения перед запуском. Далее, проверяем наше приложение в [Android App Sharing](#) и отправляем на тестирование.

Для сборки IOS необходимо использовать Xcode. Открываем проект, настраиваем сертификаты и через меню Product выполняем Archive. Внимание! Необходимо помнить о значениях переменных окружения перед запуском. А сборка должна быть в режиме Any IOS Device(arm64). Далее доставляем приложение при помощи Transporter в Test Flight и публикуем его для тестирования. Аналогично, делаем сборку для прода, изменив переменные окружения и доставляем приложение в стор.

## Публикация

Публикация [Android](#) Публикация [IOS](#)

## Как работать с next app в nx

Информация по ссылке [cheat sheet for getting starting with Next.js in this repo](#).  
Пожалуйста ознакомьтесь с [документацией](#) для next с nx.

## Для получения дополнительной информации по работе с монорепозиториями NX

Посетите [Nx Documentation](#) и изучите подробнее.

Если Вы столкнулись с проблемами во время локального развертывания ПО, сообщите нам об ошибке по электронной почте [diway.rus@gmail.com](mailto:diway.rus@gmail.com) или по номеру телефона нашего технического специалиста +7 926 288 1319.