

# Бэкенд на Node.js

Продолжительность - 3,5 месяца

## 00

Входное  
тестирование

Как устроено  
обучение

1,5 часа

## 01

Основы Node.js,  
Express и MongoDB

4 недели, 80 часов, 1 проект

## 02

PostgreSQL и Nest.js

6 недель, 80 часов, 1 проект

## 03

Деплой  
и автоматизация

4 недели, 60 часов, 2 проекта

00

0,5 часа

## Входное тестирование

Чтобы вы проверили знания и объективно оценили силы, мы предлагаем ответить на 11 вопросов. Потом предложим интерпретировать результаты и перейти к обучению.

Темы

1. Что вас ждёт в тестировании
2. Входное тестирование
3. Как интерпретировать результаты

1,5 часа

## Как устроено обучение

Вы узнаете, как устроены программа и обучение. Ознакомитесь с организационными деталями и работой команды сопровождения, а в конце мы синхронизируемся по ожиданиям.

Темы

1. Введение. Что предстоит в платной части?
2. Структура и сроки
3. Что такое переход и зачем он нужен
4. Как устроена поддержка
5. Правила
6. Убедимся, что правильно друг друга поняли
7. Что для вас важно?
8. Что получится в результате

## 01

4 недели,  
80 часов,  
1 проект

# Основы Node.js, Express и MongoDB

Вы разберётесь, как выполняется код бэкенда. Познакомитесь со средой выполнения серверного JavaScript – Node.js – и узнаете, как она работает изнутри. Затем освоите сторонние библиотеки и научитесь добавлять их в проект при помощи пакетного менеджера NPM.

После этого вы разработаете собственный веб-сервис с применением фреймворка Express, подключите к нему авторизацию и реализуете корректную обработку ошибок.

Для хранения данных приложения вы воспользуетесь гибкой и производительной NoSQL базой данных MongoDB.

### Темы

1. Введение в бэкенд-разработку
2. Node.js. Экосистема NPM
3. Компьютерные сети
4. Принципы построения API REST
5. Основы Express
6. Введение в базы данных. MongoDB
7. Работа с БД и авторизация. JWT
8. Логирование и продвинутая обработка ошибок
9. Обеспечение безопасности приложения

### Проект

#### Веб-сервис Mesto

Вы напишете бэкенд для приложения Mesto, реализуете авторизацию, валидацию данных и корректно и безопасно обработаете ошибки, которые возникают в процессе работы.

## 02

6 недель,  
80 часов,  
1 проект

# PostgreSQL и Nest.js

Вы освоите фреймворк Nest.js, который позволяет разрабатывать веб-приложения любой сложности, разбивая их на небольшие и независимые модули. Научитесь применять основные компоненты фреймворка, добавлять авторизацию и логирование, разберётесь в концепции внедрения зависимостей.

Затем вы познакомитесь с реляционными базами данных и языком SQL.

Подключите базу данных PostgreSQL к проекту с использованием библиотеки TypeORM.

### Темы

1. Введение в реляционные БД и SQL
2. Разработка бэкенда на Nest.js
3. PostgreSQL на практике
4. Passport.js. Авторизация через соцсети
5. Рутинные задачи

### Проект

#### Приложение «КупиПодариДай»

Вы разработаете модульный API сервис с использованием Nest.js, опишите маршруты и контроллеры приложения, подключите к нему базу данных PostgreSQL.

# Деплой и автоматизация

Вы научитесь создавать удалённый сервер и работать с ним при помощи терминала. После этого разместите и запустите приложение на удалённом сервере и автоматизируете доставку изменений кода.

Вы узнаете, что скрывается за концепцией контейнеров и Docker, создадите собственный Docker-контейнер, опубликуете его в хранилище DockerHub и запустите на вашем сервере.

А ещё познакомитесь с Docker Compose, который позволяет управлять группой контейнеров как единым приложением.

## Темы

1. Тестирование бэкенда
2. Создаем собственный сервер
3. Linux на практике
4. Деплой приложения на сервер
5. Погружение в мир контейнеров и Docker
6. Хранение и распространение Docker-образов. DockerHub
7. Запуск приложения из нескольких контейнеров

## Проект

### Веб-сервис Mesto

Развернёте проект на удалённом сервере с применением Git и процессного менеджера pm2. Затем автоматизируете выгрузку изменений из репозитория на сервер при помощи pm2.

### Приложение «КупиПодариДай»

Упакуете сервис в Docker и запустите его на удалённом сервере при помощи Docker Compose.