

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ САНКТ-ПЕТЕРБУРГСКИЙ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ  
Факультет информационных технологий и программирования

**Лабораторная работа №1**  
«Хостинг веб-приложения на сервисе Heroku»  
по дисциплине «Web программирование»

Выполнил: студент группы М33122 Белоногов Е.В.

Для выполнения работы были установлены:

- NodeJS

<https://nodejs.org/en/download/>

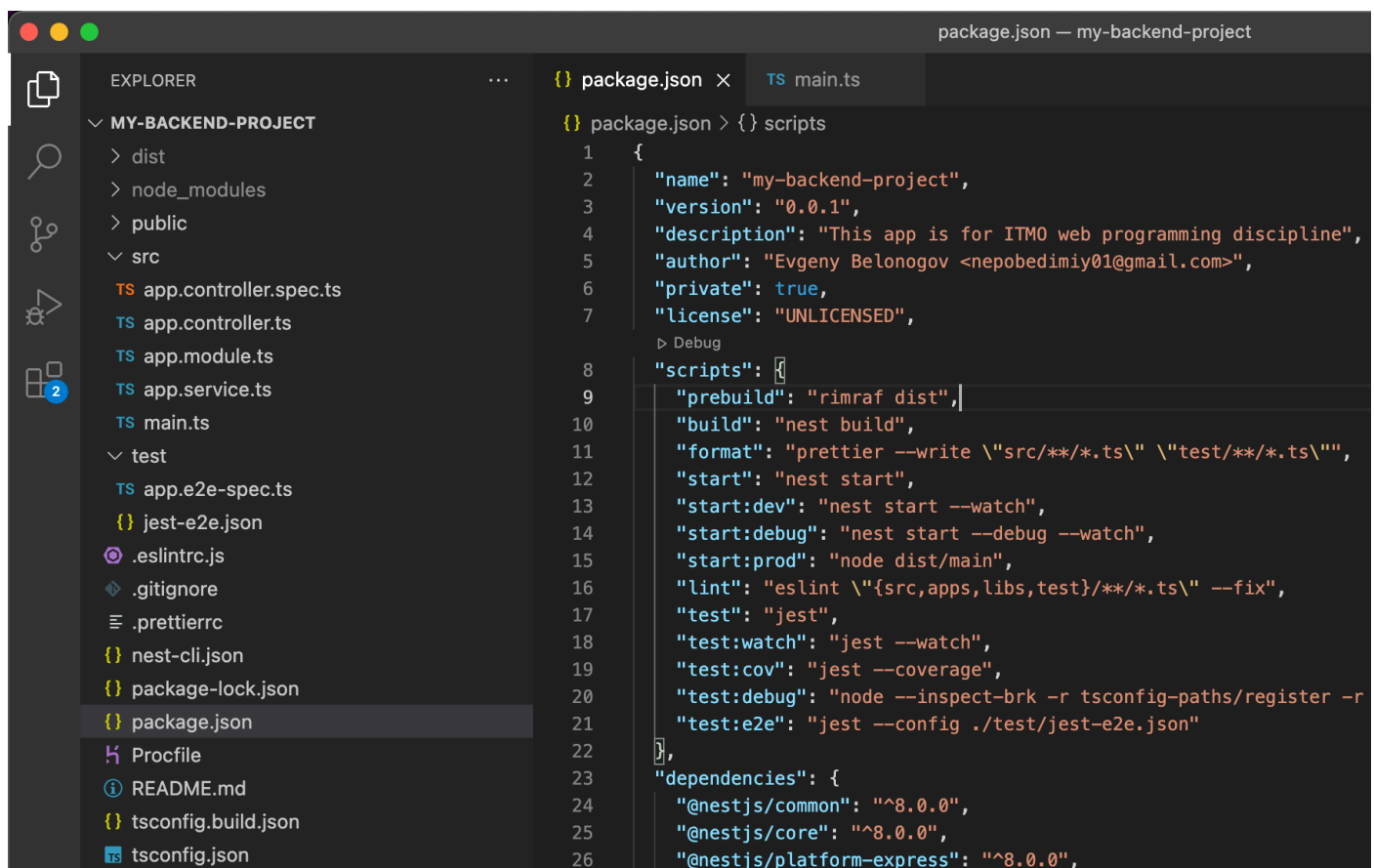
- Пакет NestJS с помощью пакетного менеджера npm:

`npm install -g @nestjs/cli`

- Heroku CLI (command line interface)

<https://devcenter.heroku.com/articles/heroku-cli#install-the-heroku-cli>

Далее с помощью консольной команды `nest new <app name>` был создан новый проект. Редактирование файла `package.json` с указанием авторства:



The screenshot shows the VS Code editor interface. On the left, the Explorer sidebar displays the file structure of a project named 'MY-BACKEND-PROJECT'. The 'src' directory contains several TypeScript files, including 'main.ts'. The 'test' directory contains 'app.e2e-spec.ts' and 'jest-e2e.json'. The 'package.json' file is selected and its content is displayed in the main editor area. The package.json file includes metadata such as name, version, description, author, and license, as well as scripts for building, running, and testing the application. The dependencies section lists '@nestjs/common', '@nestjs/core', and '@nestjs/platform-express'.

```
{
  "name": "my-backend-project",
  "version": "0.0.1",
  "description": "This app is for ITMO web programming discipline",
  "author": "Evgeny Belonogov <nepobedimiy01@gmail.com>",
  "private": true,
  "license": "UNLICENSED",
  "scripts": {
    "prebuild": "rimraf dist",
    "build": "nest build",
    "format": "prettier --write \"src/**/*.ts\" \"test/**/*.ts\"",
    "start": "nest start",
    "start:dev": "nest start --watch",
    "start:debug": "nest start --debug --watch",
    "start:prod": "node dist/main",
    "lint": "eslint \"{src,apps,libs,test}/**/*.ts\" --fix",
    "test": "jest",
    "test:watch": "jest --watch",
    "test:cov": "jest --coverage",
    "test:debug": "node --inspect-brk -r tsconfig-paths/register -r",
    "test:e2e": "jest --config ./test/jest-e2e.json"
  },
  "dependencies": {
    "@nestjs/common": "^8.0.0",
    "@nestjs/core": "^8.0.0",
    "@nestjs/platform-express": "^8.0.0",

```

Запуск проекта локально через команду:

`npm run start`

По умолчанию сервер слушает запросы на 3000 порту. Такое поведение устанавливается в файле `main.ts`. Сделать запрос можно по адресу `localhost:3000`, в результате получить ответ в виде «Hello, world».

Для того, чтобы устанавливать порт согласно значению переменной среды, был изменен файл main.ts:

```
await app.listen(process.env.PORT == undefined ? 3000 : process.env.PORT);
```

Далее создается приложение на heroku командой:

```
Heroku apps:create <app name> --region=eu
```

Создание файла Procfile с содержанием:

```
web: npm run start:prod
```

Этот файл задает heroku то, какой командой надо запускать приложение. Далее необходимо связать git с Heroku:

```
heroku git:remote
```

Далее надо указать, что хотим использовать паттерн MVC. Следуя инструкциям на <https://docs.nestjs.com/techniques/mvc>

main.ts

JS

```
import { NestFactory } from '@nestjs/core';
import { NestExpressApplication } from '@nestjs/platform-express';
import { join } from 'path';
import { AppModule } from './app.module';

async function bootstrap() {
  const app = await NestFactory.create<NestExpressApplication>(
    AppModule,
  );

  app.useStaticAssets(join(__dirname, '..', 'public'));
  app.setBaseViewsDir(join(__dirname, '..', 'views'));
  app.setViewEngine('hbs');

  await app.listen(3000);
}
bootstrap();
```

Закидываем всю статику в папку public. Делаем пуш на heroku.