

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего
образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №1
по дисциплине «Web-программирование»

Автор: Шарапков Егор Викторович М33081

Преподаватель: Приискалов Роман Андреевич



УНИВЕРСИТЕТ ИТМО

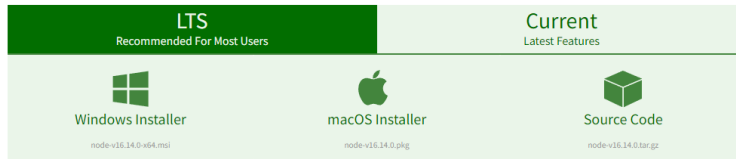
Санкт-Петербург 2022

Скачиваем инсталлятор NodeJS с официального сайта.

Downloads

Latest LTS Version: 16.14.0 (includes npm 8.3.1)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.



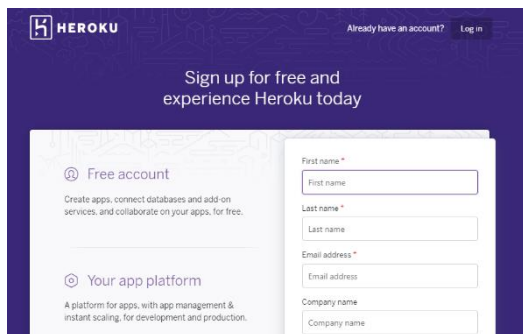
Node.js — это платформа с открытым исходным кодом для работы с языком JavaScript, построенная на движке Chrome V8. Она позволяет писать серверный код для веб-приложений и динамических веб-страниц, а также программ командной строки.

Устанавливаем пакет NestJS CLI глобально через пакетный менеджер.

```
$ npm install -g @nestjs/cli
```

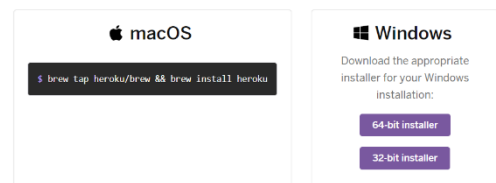
NestJS — это фреймворк для создания эффективных масштабируемых Node.js серверные приложения, созданные с использованием TypeScript и полностью поддерживающие его.

Регистрируемся на хостинг-сервисе Heroku, устанавливаем Heroku CLI.



Install with an Installer

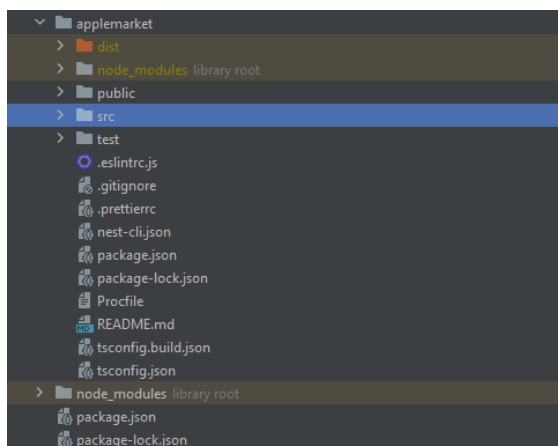
The Windows installers display a warning titled "Windows protected your PC" to some users. To run the installation when this warning shows, click "More info", verify the publisher as "salesforce.com, inc", then click the "Run anyway" button.



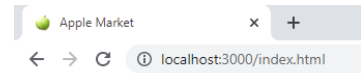
Heroku — облачная мультязычная платформа как услуга (PaaS), основанная на управляемой контейнерной системе, с интегрированными службами передачи данных и развитой экосистемой для развертывания и запуска приложений.

Создаем приложение Nest. `nest new applemarket`

В результате сгенерирован проект в новой директории с именем проекта.



Запускаем, приложение запускается на 3000-м порту.



```
PS C:\Users\Egor\Desktop\web 6 sem\applemarket> npm run start

> applemarket@0.0.1 start
> nest start

[Nest] 13652 - 12.03.2022, 23:23:22 LOG [NestFactory] Starting Nest application...
[Nest] 13652 - 12.03.2022, 23:23:22 LOG [InstanceLoader] AppModule dependencies initialized +26ms
[Nest] 13652 - 12.03.2022, 23:23:22 LOG [RoutesResolver] AppController {/}: +5ms
[Nest] 13652 - 12.03.2022, 23:23:22 LOG [RouterExplorer] Mapped {/, GET} route +2ms
[Nest] 13652 - 12.03.2022, 23:23:22 LOG [NestApplication] Nest application successfully started +2ms
```

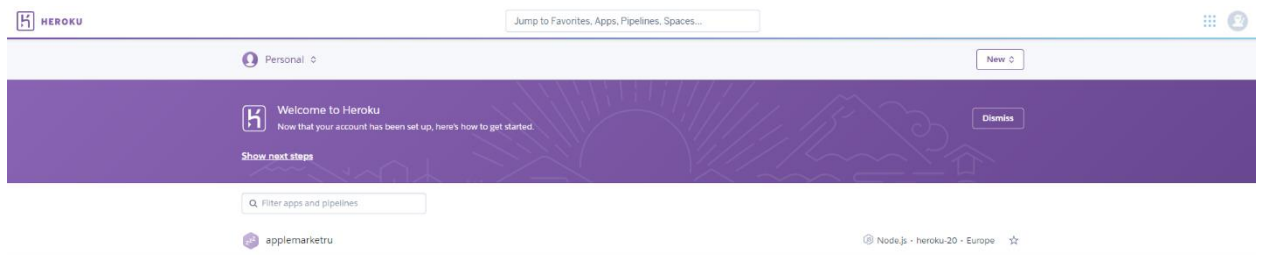
Т. к. Heroku сообщает нам порт через переменную окружения с именем PORT номер порта, на котором необходимо принимать соединения, отредактируем файл main.ts

```
const port = process.env.PORT || 3000;
await app.listen(port);
```

Создаем инстанс приложения на хостинге.

```
PS C:\Users\Egor\Desktop\web 6 sem\applemarket> heroku apps:create applemarketru --region eu
```

Убеждаемся, что приложение создалось.



Указываем входную точку (предоставляем команду, с помощью которой должно запускаться приложение) в файле Procfile.

```
main.ts x Procfile x app.controller.ts x
1 web: npm run start:prod
```

Инициализируем репозиторий и связываем с Heroku.

Create a new Git repository

Initialize a git repository in a new or existing directory

```
$ cd my-project/
$ git init
$ heroku git:remote -a boiling-inlet-60608
```

Deploy your application

Commit your code to the repository and deploy it to Heroku using Git.

```
$ git add .
$ git commit -am "make it better"
$ git push heroku master
```

Теперь надо разместить все статические ресурсы. Для этого укажем в момент запуска нашего Nest-приложения, что мы хотим воспользоваться шаблоном MVC, основанном на фреймворке Express. Укажем, что хотим иметь возможность отдавать статические ресурсы, расположенные в папке public.

Чтобы создать приложение MVC, нам также нужен механизм шаблонов для отображения наших HTML-представлений:

```
$ npm install --save hbs
```

```
const app = await NestFactory.create<NestExpressApplication>(AppModule);

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

Мы сказали Express, что public каталог будет использоваться для хранения статических ресурсов, views будет содержать шаблоны, а hbs механизм шаблонов должен использоваться для вывода HTML.

С помощью команды heroku apps:open открываем приложение в браузере.

