

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Практическая работа 1

Выполнил:

Никитин Павел

Группа К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2022 г.

Задача

Установка npm (10.2.4), node.js (18.19.1 / 20.11.1)

Инициализация npm-пакета

Установка express

Удаление express/попытка его обновить

Написание кастомной команды для npm

Ход работы

Установил nvm, чтобы можно было автоматически менять версию npm и node.

```
NikitinPavel@LAPTOP-FLIG82Q1 MINGW64 ~/Documents/GitHub/ITMO-ICT-Backend-2024/homeworks/K33402 (master)
$ nvm list

* 21.7.1 (Currently using 64-bit executable)
  21.6.1
  16.13.0
  14.21.3
  13.14.0
```

Рисунок 1 - все установленные версии node

```
NikitinPavel@LAPTOP-FLIG82Q1 MINGW64 ~/Documents/GitHub/ITMO-ICT-Backend-2024/homeworks/K33402 (master)
$ nvm install 20.11.1
Downloading node.js version 20.11.1 (64-bit)...
Extracting node and npm...
Complete
npm v10.2.4 installed successfully.

Installation complete. If you want to use this version, type

nvm use 20.11.1

NikitinPavel@LAPTOP-FLIG82Q1 MINGW64 ~/Documents/GitHub/ITMO-ICT-Backend-2024/homeworks/K33402 (master)
$ nvm use 20.11.1
Now using node v20.11.1 (64-bit)
```

Рисунок 2 - установка и использование нужной версии

В качестве сданной лабораторной хочу показать ранее реализованный проект на express node js и express с кастомными npm скриптами;

<https://github.com/pasha-nikitin-2003/shop-boilerplate-backend>


```


1  {
2    "name": "shop-boilerplate-backend",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.ts",
6    "scripts": {
7      "dev": "nodemon",
8      "build:file": "tsc && esbuild src/index.ts --bundle --minify --platform=node --target=node16.2 --outfile=build.js",
9      "start": "node index.js"
10   },
11   "keywords": [],
12   "author": "",
13   "license": "ISC",
14   "dependencies": {
15     "cors": "^2.8.5",
16     "express": "^4.18.2",
17     "typescript": "^5.1.6"
18   },
19   "devDependencies": {
20     "@types/cors": "^2.8.13",
21     "@types/express": "^4.17.17",
22     "@types/node": "^20.4.2",
23     "esbuild": "^0.18.13",
24     "esbuild-node-tsc": "^2.0.5",
25     "nodemon": "^3.0.1",
26     "ts-node": "^10.9.1"
27   }
28 }

```

Рисунок 3 - файл package.json

В файле package.json мы написали кастомный скрипт build:file который при помощи esbuild компилирует код в 1 файл, это нужно для дальнейшего хостинга на vercel, да туда можно выкладывать и бек, правда присутствуют достаточно серьезные ограничения на мощности.


main
shop-boilerplate-backend / src / routes /


pasha-nikitin-2003
update server

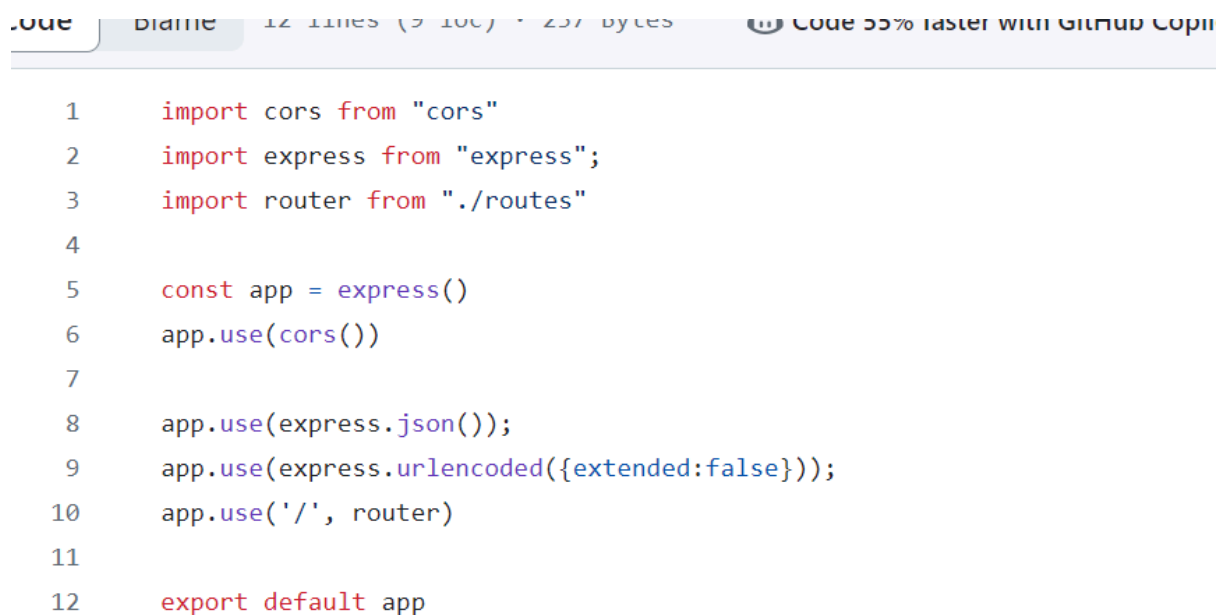
Name	Last commit message
..	
getFilteredProducts.ts	initial commit
getInitialData.ts	update server
getProductInfo.ts	initial commit
index.ts	initial commit

Рисунок 4 - endpoints для запросов

```
import { PRODUCTS } from '../data'
import { Request, Response } from 'express';

/** Отдает данные товара по id */
✓ export function getProductInfo(req: Request, res: Response) {
  const product = PRODUCTS.find((product) => product.id === Number(req.query.id))
  return res.status(200).json({product: product})
}
```

Рисунок 5 - пример контроллера

A screenshot of a code editor interface. At the top, there's a header bar with tabs labeled 'Code' and 'Blame'. To the right of the tabs, it shows '12 lines (9100) · 237 bytes' and a GitHub logo with the text 'Code 55% faster with GitHub Copilot'. The main area displays a JavaScript file with 12 lines of code. The code imports 'cors' from 'cors', 'express' from 'express', and 'router' from './routes'. It then creates an Express app, uses 'cors()', 'express.json()', and 'express.urlencoded({extended: false})'. The router is mounted on the root path ('/'). Finally, the app is exported as the default export.

```
1 import cors from "cors"
2 import express from "express";
3 import router from "./routes"
4
5 const app = express()
6 app.use(cors())
7
8 app.use(express.json());
9 app.use(express.urlencoded({extended:false}));
10 app.use('/', router)
11
12 export default app
```

Рисунок 6 - подключение express

Вывод

В ходе работы я научился работать с npm и устанавливать различные версии node js, а также работать с express настраивать сборку esbuild, писать кастомные npm команды.