

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Практическая работа 1: изучение основ работы
с менеджером зависимостей prn

Выполнил: Золотухин А.Н. К33392

Проверил: Добряков Д. И.

Санкт-Петербург

2024 г.

Задание.

Изучить основы работы с менеджером зависимостей npm.

Ход работы.

1) Установка node.js – для этого достаточно перейти на <https://nodejs.org/en/download> и скачать от туда LTS версию:



Downloads
Latest LTS Version: 20.11.1 (includes npm 10.2.4)

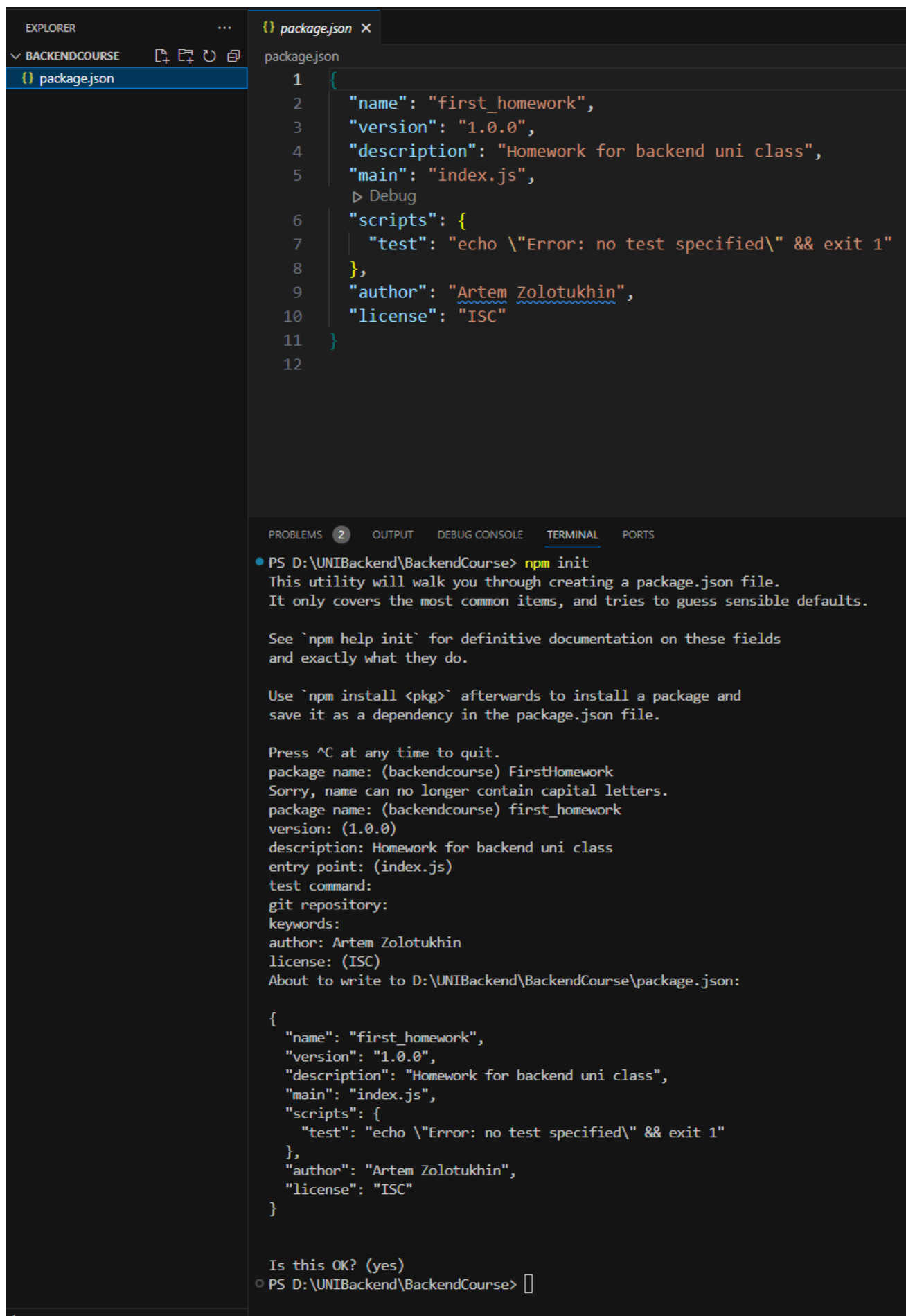
Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS Recommended For Most Users	Current Latest Features	
 Windows Installer node-v20.11.1-x64.msi	 macOS Installer node-v20.11.1.pkg	 Source Code node-v20.11.1.tar.gz

Windows Installer (.msi)	32-bit	64-bit	ARM64
Windows Binary (.zip)	32-bit	64-bit	ARM64
macOS Installer (.pkg)	64-bit / ARM64		
macOS Binary (.tar.gz)	64-bit	ARM64	
Linux Binaries (x64)	64-bit		
Linux Binaries (ARM)	ARMv7	ARMv8	
Source Code	node-v20.11.1.tar.gz		

2) Инициализация npm-пакета – для этого нужно ввести команду npm init:

При инициализации проекта, если нет нужных пакетов, нас попросят их установить, а после можно сразу переходить к инициализации проекта:



3) Установка express – для этого перейдем на страницу установки express (<https://expressjs.com/en/starter/installing.html>) и пройдемся по всем шагам:

Installing

Assuming you've already installed [Node.js](#), create a directory to hold your application, and make that your working directory.

```
$ mkdir myapp  
$ cd myapp
```

Use the `npm init` command to create a `package.json` file for your application. For more information on how `package.json` works, see [Specifics of npm's package.json handling](#).

```
$ npm init
```

This command prompts you for a number of things, such as the name and version of your application. For now, you can simply hit RETURN to accept the defaults for most of them, with the following exception:

```
entry point: (index.js)
```

Enter `app.js`, or whatever you want the name of the main file to be. If you want it to be `index.js`, hit RETURN to accept the suggested default file name.

Now install Express in the `myapp` directory and save it in the dependencies list. For example:

```
$ npm install express
```

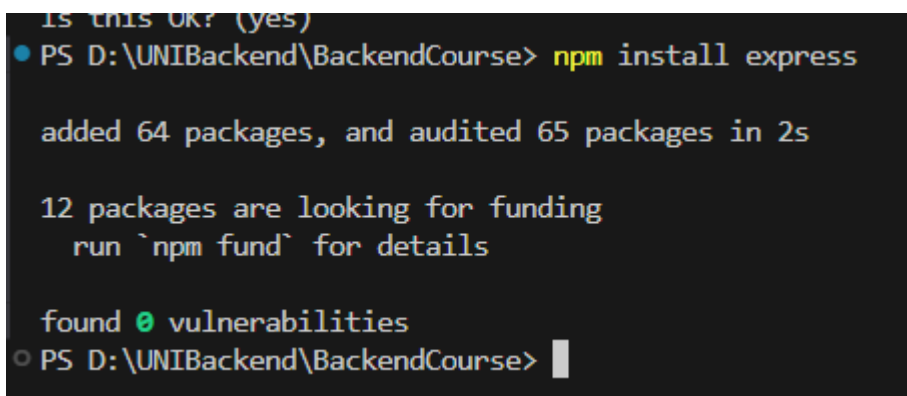
To install Express temporarily and not add it to the dependencies list:

```
$ npm install express --no-save
```

By default with version npm 5.0+ npm install adds the module to the dependencies list in the `package.json` file; with earlier versions of npm, you must specify the `--save` option explicitly. Then, afterwards, running `npm install` in the app directory will automatically install modules in the dependencies list.

Next: [Hello World](#)

Поскольку `npm init` мы уже прогнали, можно сразу перейти к установке `express`-а используя команду `npm install express`:



```
Is this OK? (yes)  
● PS D:\UNIBackend\BackendCourse> npm install express  
  
added 64 packages, and audited 65 packages in 2s  
  
12 packages are looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities  
● PS D:\UNIBackend\BackendCourse> 
```

При этом - у меня создался файл `package-lock.json` — в нем описываются точные версии пакетов, используемых в проекте.

4) Удаление `express`/попытка его обновить — для удаления пакетов используется команда `npm uninstall *package_name*`:

```
PS D:\UNIBackend\BackendCourse> npm uninstall express

removed 64 packages, and audited 1 package in 386ms

found 0 vulnerabilities
PS D:\UNIBackend\BackendCourse>
```

А для обновления версии пакета используется команда `npm update *package name*`:

```
PS D:\UNIBackend\BackendCourse> npm update express

up to date, audited 65 packages in 674ms

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS D:\UNIBackend\BackendCourse>
```

5) Написание кастомной команды для npm, чтобы можно удалить express командой “`npm run me`”:

Для этого создадим скрипт в `package.json`:

```
package.json X
package.json
 2  "name": "first_homework",
 3  "version": "1.0.0",
 4  "description": "Homework for backend uni class",
 5  "main": "index.js",
 6  "scripts": {
 7    "rme": "npm uninstall express"
 8  },
 9  "author": "Artem Zolotukhin",
10  "license": "ISC"
11 }
12
```

И используем эту команду через `npm run *script name*`:

```
{} package.json X
package.json
1  {
2    "name": "first_homework",
3    "version": "1.0.0",
4    "description": "Homework for backend uni class",
5    "main": "index.js",
6    "scripts": {
7      "rme": "npm uninstall express"
8    },
9    "author": "Artem Zolotukhin",
10   "license": "ISC"
11 }
12

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS D:\UNIBackend\BackendCourse> npm run rme

> first_homework@1.0.0 rme
> npm uninstall express

removed 64 packages, and audited 1 package in 334ms

found 0 vulnerabilities
○ PS D:\UNIBackend\BackendCourse> 
```

Выводы.

В ходе данной работы изучены основы работы с npm.