

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бек-энд разработка

Отчет

Практическая работа №1

“Основы работы с npm, node.js”

Выполнил:  
Стукалов Артем  
К33392

Проверил:  
Добряков Д. И.

Санкт-Петербург

2024 г.

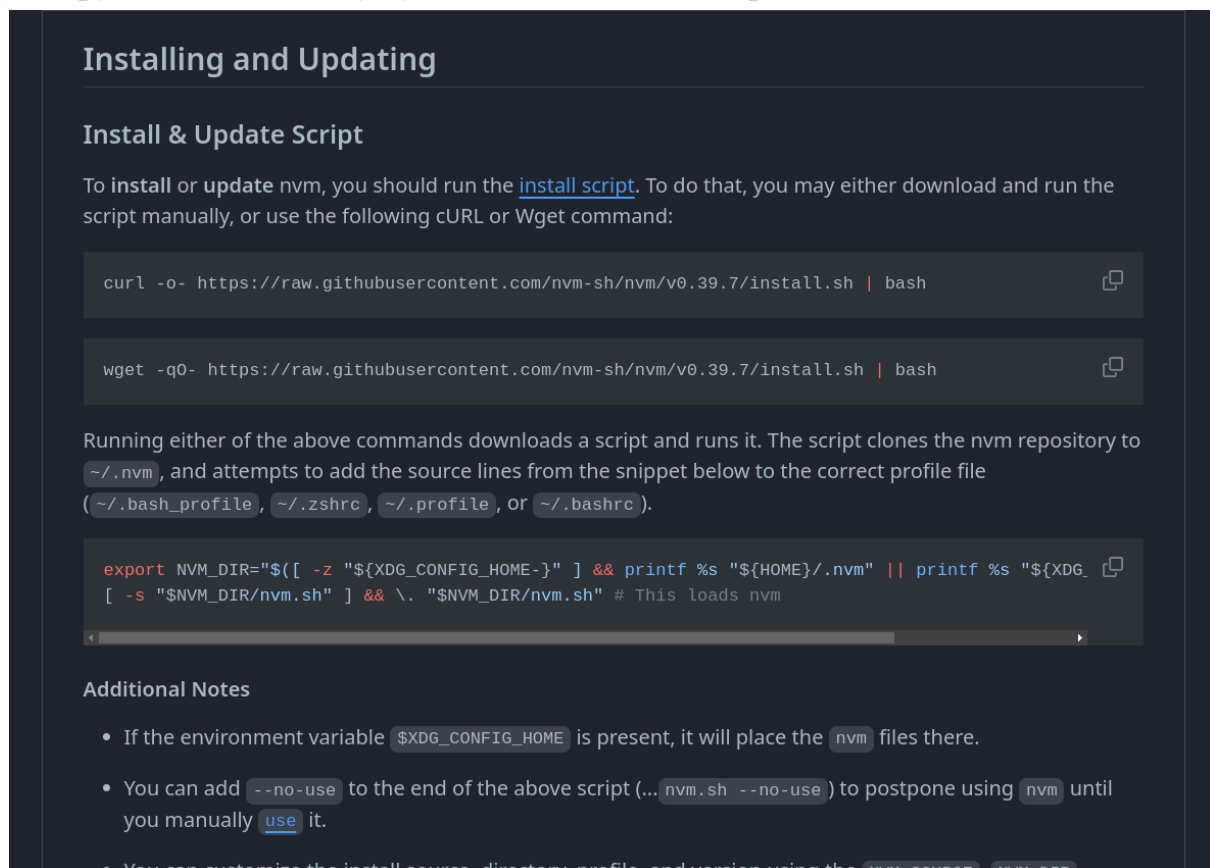
## Задание:

1. Установка npm (10.2.4), node.js (18.19.1 / 20.11.1)
2. Инициализация npm-пакета
3. Установка express
4. Удаление express/попытка его обновить
5. Написание кастомной команды для npm, чтобы можно удалить express командой “npm run rme”

## Установка node и npm

Для установки node и npm воспользуемся менеджером версий node - nvm. Это позволит нам в дальнейшем без особых проблем обновляться на более актуальные версии и изменять их по выполнению одной команды.

Перейдем на страницу проекта на github и выполним установку по инструкциям, используя установочный bash скрипт.



The screenshot shows the 'Installing and Updating' section of the nvm GitHub repository. It includes the 'Install & Update Script' section with instructions on how to install or update nvm using curl or wget. It also shows the commands to run the script and the resulting output, which includes the installation of nvm and the addition of source lines to the user's profile file. The 'Additional Notes' section provides further details on environment variables and options.

### Installing and Updating

#### Install & Update Script

To install or update nvm, you should run the [install script](#). To do that, you may either download and run the script manually, or use the following cURL or Wget command:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash
```

```
wget -qO- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash
```

Running either of the above commands downloads a script and runs it. The script clones the nvm repository to `~/.nvm`, and attempts to add the source lines from the snippet below to the correct profile file (`~/.bash_profile`, `~/.zshrc`, `~/.profile`, or `~/.bashrc`).

```
export NVM_DIR="$([ -z "${XDG_CONFIG_HOME-}" ] && printf %s "${HOME}/.nvm" || printf %s "${XDG_
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
```

#### Additional Notes

- If the environment variable `$XDG_CONFIG_HOME` is present, it will place the `nvm` files there.
- You can add `--no-use` to the end of the above script (`... nvm.sh --no-use`) to postpone using `nvm` until you manually [use](#) it.
- You can customize the install source, directory, profile, and version using the `NVM_SOURCE`, `NVM_DIR`, `NVM_VERSION`, and `NVM_ARCH` environment variables.

Далее через команду “nvm list-remote” отобразим список доступных версий. Установим последнюю lts версию используя “nvm install v20.11.1”.

```
v19.6.1
v19.7.0
v19.8.0
v19.8.1
v19.9.0
v20.0.0
v20.1.0
v20.2.0
v20.3.0
v20.3.1
v20.4.0
v20.5.0
v20.5.1
v20.6.0
v20.6.1
v20.7.0
v20.8.0
v20.8.1
v20.9.0
v20.10.0
v20.11.0
-> v20.11.1 (Latest LTS: Iron)
v21.0.0
v21.1.0
v21.2.0
v21.3.0
v21.4.0
v21.5.0
v21.6.0
v21.6.1
v21.6.2
astukalov:~$ nvm install v20.11.1
^C
astukalov:~$ node --version
v20.11.1
astukalov:~$ npm --version
10.2.4
astukalov:~$
```

## Инициализация npm-пакета

Просто выполним команду “npm init” и по каждому пункту нажмем enter, чтобы создать базовый npm репозиторий.

```
astukalov:~$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See 'npm help init' for definitive documentation on these fields
and exactly what they do.

Use 'npm install <pkg>' afterwards to install a package and
save it as a dependency in the package.json file.

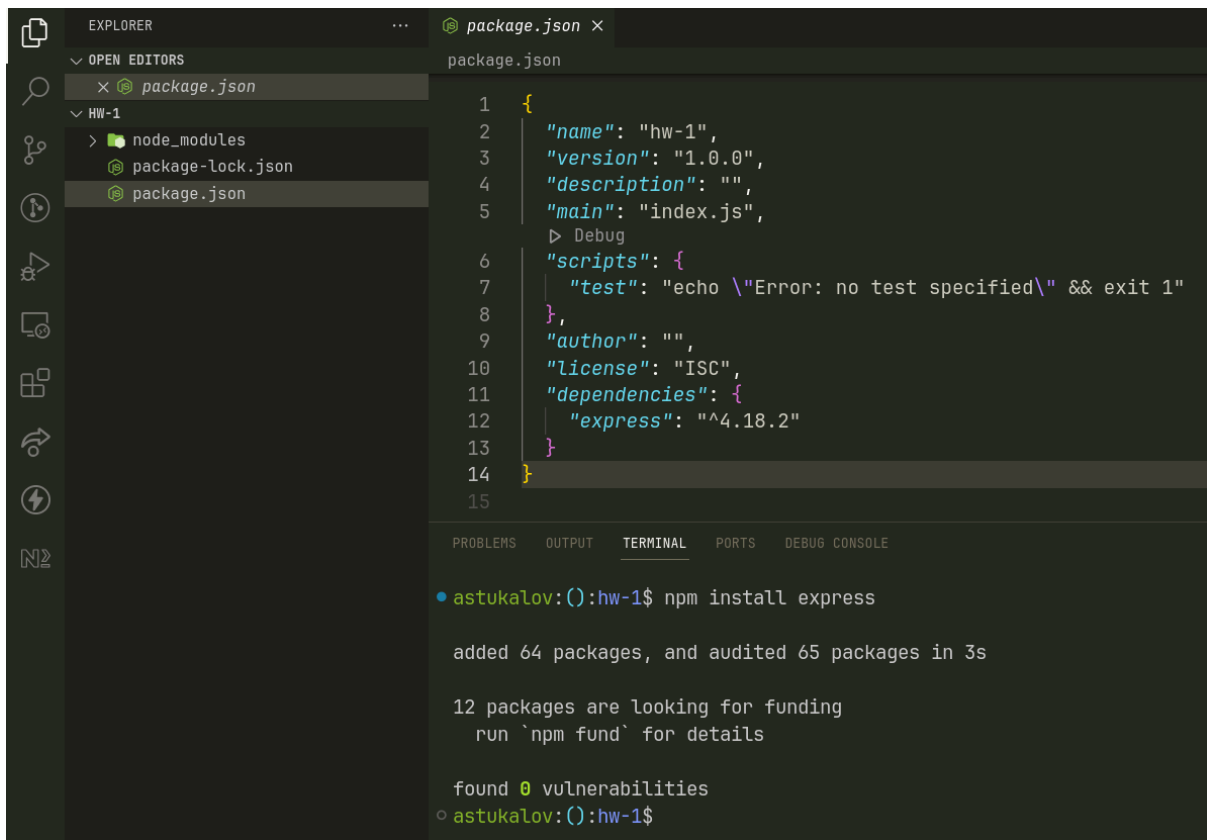
Press ^C at any time to quit.
package name: (hw-1)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/astukalov/Documents/ITMO/backend-ts/hw-1/package.json:

{
  "name": "hw-1",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes)
astukalov:~$ ls -la
total 4
drwxr-xr-x. 1 astukalov astukalov 24 Feb 23 01:41 .
drwxr-xr-x. 1 astukalov astukalov  8 Feb 23 01:21 ..
-rw-r--r--. 1 astukalov astukalov 200 Feb 23 01:41 package.json
astukalov:~$
```

## Установка express

Выполним установку через команду “npm install <package\_name>”.



The screenshot shows the VS Code interface. The Explorer sidebar on the left shows a project named 'HW-1' with files 'package-lock.json' and 'package.json'. The main editor displays the 'package.json' file with the following content:

```
1 {
2   "name": "hw-1",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1"
8   },
9   "author": "",
10  "license": "ISC",
11  "dependencies": {
12    "express": "^4.18.2"
13  }
14 }
```

Below the editor, the TERMINAL panel shows the command 'npm install express' and its output:

```
• astukaLov():hw-1$ npm install express
added 64 packages, and audited 65 packages in 3s
12 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
○ astukaLov():hw-1$
```

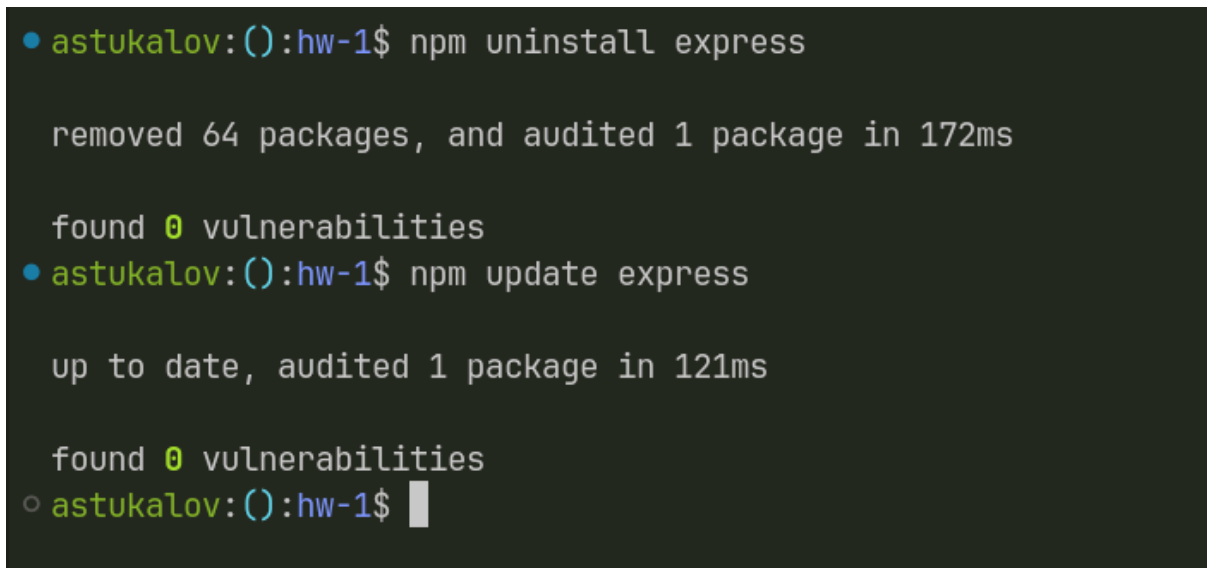
## Удаление express/попытка его обновить

Удаление можно выполнить через команду -

“npm uninstall <package\_name>”.

Обновление можно выполнить через команду -

“npm update <package\_name>”.



The screenshot shows a terminal window with the following commands and output:

```
• astukaLov():hw-1$ npm uninstall express
removed 64 packages, and audited 1 package in 172ms
found 0 vulnerabilities
• astukaLov():hw-1$ npm update express
up to date, audited 1 package in 121ms
found 0 vulnerabilities
○ astukaLov():hw-1$
```

## Команда rme для удаления express

Реализуем через добавление кастомной script команды.

```
1  {
2    "name": "hw-1",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "rme": "npm uninstall express"
8    },
9    "author": "",
10   "license": "ISC"
11 }
12
```

PROBLEMS   OUTPUT   TERMINAL   PORTS   DEBUG CONSOLE

```
• astukalov:():hw-1$ npm run rme

> hw-1@1.0.0 rme
> npm uninstall express

removed 64 packages, and audited 1 package in 134ms

found 0 vulnerabilities
○ astukalov:():hw-1$
```

## Выводы

В ходе работы были изучены основы работы с npm.