МИНОБРНАУКИ РОССИИ

–––––––——————————–––––––

Санкт-Петербургский государственный  
электротехнический университет «ЛЭТИ»

————————————————————

**Web-программирование**

**Лабораторная № 2**

**Тема: СОЗДАНИЕ ВЕБ ПРОЕКТА С ИСПОЛЬЗОВАНИЕМ NODEJS**

Выполнил: студент группы 3311 Шарпинский Денис

Проверил: Калмыков Михаил Александрович

Санкт-Петербург

Издательство СПбГЭТУ «ЛЭТИ»

2025

Цель работы: Установка среды разработки Web-сервера.

В ходе предыдущей работы был установлен веб-сервер nginx, который работает как прокси для порта 8000. Теперь необходимо установить среду разработки. В предыдущей работе был сделан выбор в пользу nodejs.

Если ранее nodejs не была установлена, то это можно сделать с официального сайта. После установки она будет доступна в терминале. В рабочей директории необходимо выполнить команду npm init. Для работы нам потребуются следующие библиотеки:

1. express
2. cookie-parser
3. cors
4. typescript
5. uuid
6. nodemon
7. jest
8. supertest
9. ts-jest
10. ts-node

А также типы для работы библиотек:

1. @types/cookie-parser
2. @types/cors
3. @types/express
4. @types/jest
5. @types/node
6. @types/supertest

Всё это можно установить, используя команду npm install. После установки необходимо создать tsconfig.json для работы typescript.

{

  "compilerOptions": {

    "target": "es6",

    "module": "commonjs",

    "outDir": "./dist",

    "strict": true,

    "esModuleInterop": true,

    "skipLibCheck": true,

    "forceConsistentCasingInFileNames": true

  },

  "include": [

    "src/\*\*/\*"

  ]

}

После создания конфига в папке src нужно создать server.ts. В этом файле и будет находиться сервер. Затем для работы nodemon (эта библиотека обеспечит удобство при работе с ts сервером, давая live-reload при изменении файла сервера) необходимо создать его конфиг

{

  "watch": ["src"],

  "ext": "ts",

  "exec": "ts-node src/server.ts"

}

В самом server.ts необходимо импортировать библиотеку express, создать приложение, обозначить порт, создать базовый роут и запустить прослушивание подключение по порту.

Для этого напишем:

import express from 'express';

const app = express();

const port = 8000;

app.get(“/”, (req, res) => {

res.status(200).send(“Hello World”);

});

app.listen(port, () => {  
 console.log(`Сервер запущен на http://localhost:${port}`);

});

Перед запуском необходимо убедиться, что в package.json есть необходимые скрипты:

"scripts": {

    "test": "jest",

    "dev": "nodemon",

    "build:server": "tsc",

    "build:client": "tsc -p public/ts/tsconfig.json",

    "build": "npm run build:server && npm run build:client",

    "watch:server": "tsc --watch",

    "watch:client": "tsc -p public/ts/tsconfig.json --watch"

  },

Для запуска в рабочей директории напишем npm run dev. Проверить работу сервера можно, открыв в браузере localhost:8000.

Также теперь можно увидеть изменения по адресу <https://localhost:12345> . Поскольку на 8000 порту запущен сервер, можно будет увидеть сообщение hello world.

Итого после установки зависимостей будет получена следующая структура папок:

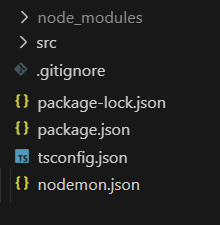


Рис 2.1

Выводы:

В ходе выполнения работы была установлена среда разработки nodejs и настроена рабочая директория проекта. Также был создан и запущен сервер приложения.