Міністерство освіти і науки України Львівський національний університет імені Івана Франка

Факультет електроніки та комп’ютерних технологій

Звіт

про виконання лабораторної роботи № 4

з курсу “Веб-програмування на стороні сервера ”

«Створення примітивного кешуючого проксі-сервера

на базі модуля node:http»

Виконав

Студент групи ФЕІ-25

Хоптій Дмитро

Перевірив:

Чмихало О. С.

Львів-2024

Мета роботи:

- Опанувати вбудований модуль node:http
- На практиці перевірити знання про протокол http, структуру запиту та відповіді
- Навчитися застосовувати відомі патерни асинхронного програмування
- Застосувати програму curl та браузер для тестування проксі сервера

Хід роботи

Підготовка до роботи :

1. Створити новий репозиторій з назвою bc2024-4
2. Встановлюю ім’я користувача та електронну пошту для коміту
3. Створюю файл package.json у кореневій теці репозиторію. Створюю головний файл програми server.js. Встановлюю за допомогою команд

```
npm install commander superagent
npm install --save-dev nodemon
```

пакети Commander.js і superagent локально. Встановлюю пакет nodemon локально. Комічу файли package.json та головний файл програми.

Частина 1 - параметри командного рядка та Веб сервер

1. Налаштування параметрів командного рядка :

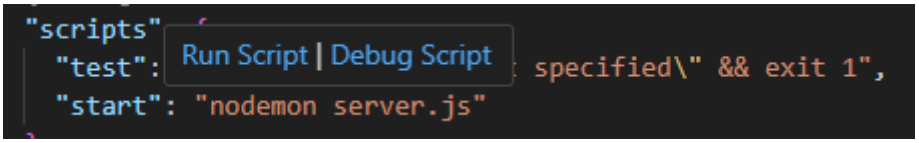
```
program
  .requiredOption('-h, --host <type>', 'server host') // Обов'язковий параметр для хоста (адреси сервера)
  .requiredOption('-p, --port <type>', 'server port') // Обов'язковий параметр для порту (на якому працюватиме сервер)
  .requiredOption('-c, --cache <type>', 'cache directory') // Обов'язковий параметр для шляху до кешу
  .parse(process.argv); // Читання параметрів із командного рядка

// Отримання значення параметрів
const { host, port, cache } = program.opts(); // Збереження значення командних параметрів
```

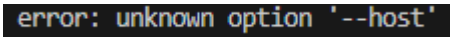
2. За допомогою модуля http запуск веб сервера:

```
// Створення HTTP-сервер
const server = http.createServer((req, res) => {
  res.statusCode = 200; // Встановлення код відповіді 200 що є успіхом
  res.setHeader('Content-Type', 'text/plain'); // Встановлення заголовку відповіді як тексту
  res.end('Proxy server is running\n'); // Відповідь клієнту
});
```

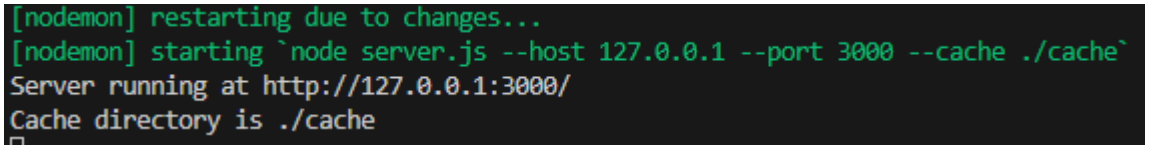
3. Автоматичний перезапуск після зміни програми :

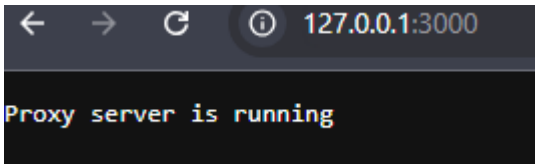


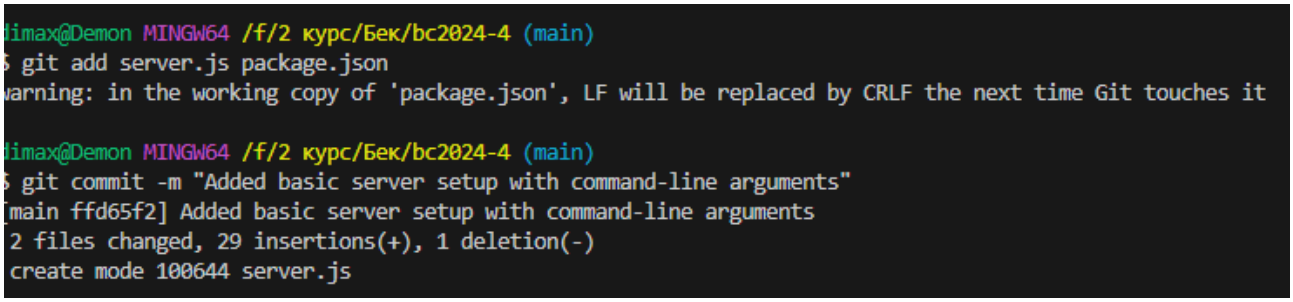
4. Якщо не заданий один з обов’язкових параметрів :



5. Комітимо результат :







Частина 2 - читання, запис та видалення файлів з кешу

Використання асинхронних викликів fs.promise.readFile та

fs.promise.writeFile для читання та запису файлів :

Кожен з http запитів приймає назву http коду, для якого ми демонструємо картинку,

яку передають у шляху URL (наприклад, /200 означає картинку для коду 200)

● Після реалізації одного з методів, зробіть коміт зі змінами перез тим, як починати

працювати над наступним. Протестуйте роботу запиту перед тим як робити коміт.

● Сервер має відповідати на запити з наступними методи HTTP:

○ GET - отримати картинку з кешу на диску, яка відповідає заданому коду

HTTP. Картинка має бути у тілі відповіді.

○ PUT - записати картинку, яка відповідає заданому коду HTTP, у кеш на диск

або замінити існуючу. Картинка яку зберігають має міститися у тілі запиту.

○ DELETE - видалити картинку, яка відповідає заданому коду HTTP, з кешу

● Якщо використано будь який інший метод, то сервер має повернути відповідь зі

статусним кодом 405 (Method not allowed)

● Якщо картинку не знайдено у кеші, сервер має повернути відповідь з кодом 404

(Not Found)

● Для кожного успішного запиту відповідь має містити код 200 (OK), крім методу PUT,

який має повертати код 201 (Created)

● Відповідь яка містить картинку, має мати хедер Content-Type зі значенням

image/jpeg

Частина 3 - реалізація запиту на сервер http.cat

● Додайте перевірку, яка буде надсилати запит на отримання картинки з сайту

https://http.cat у випадку, коли кеш не має такої картинки.

○ Для запиту підключіть і використайте модуль superagent

○ Для отримання картинки, передайте стаусний код HTTP у шляху URL

https://http.cat

○ Якщо запит завершився помилкою, то проксі-сервер має повернути

відповідь з кодом 404 (Not Found)

○ Якщо запит був успішний, збережіть картинку у кеш, так щоб наступного разу

проксі-сервер брав картинку з кешу без запиту на сервер

● Протестуйте роботу проксі сервера на відповідність вимогам. Закомітьте результат

роботи.

Висновок