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 імʼя 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 :

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

1. За допомогою модуля http запуск веб сервера:

Зображення, що містить знімок екрана, текст, Шрифт

Автоматично згенерований опис

1. Автоматичний перезапуск після зміни програми :

Зображення, що містить текст, Шрифт, знімок екрана, ряд

Автоматично згенерований опис

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



1. Комітимо результат :

Зображення, що містить текст, Шрифт, знімок екрана

Автоматично згенерований опис

Зображення, що містить текст, Шрифт, знімок екрана, число

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

**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**