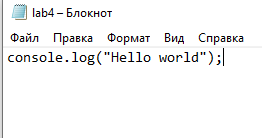
Задания по NodeJS

**Задача 0**

**Поставить NodeJs и позапускать \*.js-файлы из консоли**







**Задача 1**

**Написать скрипт, который считывает текстовый файл с диска и записывает его в обратном порядке ('абв' -> 'вба') в тот же файл.**

* Создаём файл script.js и пишем в него скрипт:

const fs = require('fs');

const filePath = 'Текст.txt';

fs.readFile(filePath, 'utf8', (err, data) =>

{

if (err)

{

console.error('Ошибка при чтении файла:', err);

return;

}

const reversedData = data.split('').reverse().join('');

fs.writeFile(filePath, reversedData, (err) => {

if (err) {

console.error('Ошибка при записи в файл:', err);

return;

}

console.log('Данные успешно инвертированы и сохранены в файле!');

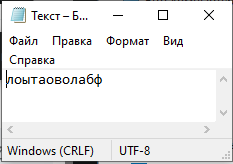
});

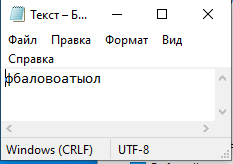
});

(модуль fs для работы с файловой системой в Nodejs)

-Запускаем js файл:  


-Результат:

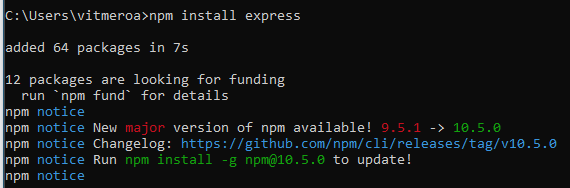




**Задача 2**

**Написать сервер, который при запросе GET / отправляет содержимое некоего файла на диске (см. предыдущую задачу). POST / добавляет в файл тело запроса (body).**

* Подключаем модуль Express.js и модуль fs для работы с файлами



* Создаём файл server.js и пишем в него скрипт:

const express = require('express');

const fs = require('fs');

const app = express();

const PORT = 3000;

app.use(express.text());

app.get('/', (req, res) => {

fs.readFile('Текст.txt', 'utf8', (err, data) => {

if (err) {

res.status(500).send('Ошибка чтения файла');

} else {

res.send(data);

}

});

});

app.post('/', (req, res) => {

fs.appendFile('Текст.txt', req.body, 'utf8', (err) => {

if (err) {

res.status(500).send('Ошибка записи в файл');

} else {

res.send('Данные успешно добавлены в файл');

}

});

});

app.listen(PORT, () => {

console.log(`Сервер запущен на порту ${PORT}`);

});

(подключаем модуль Express.js и модуль fs для работы с файлами, создаем экземпляр приложения Express, настраиваем обработку get запроса по пути /, где читается содержимое файла и отправляется клиенту в ответ на запрос. Настраиваем обработку POST запроса по корневому пути /, где получаем новые данные от клиента, добавляем их в файл и отправляем обратно сообщение о успешном добавлении.

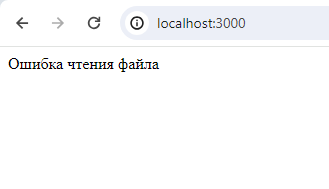
Запускаем сервер и выводим сообщение о том, что он запущен.)

-Запускаем

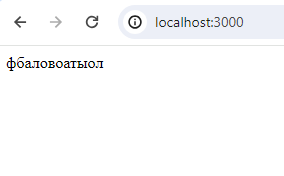


-Результат работы:

Когда текстовый файл еще не был создан:



После создания:



* Отправка POST запроса на http://localhost:3000 с данными, которые

нужно добавить в файл.



-Результат работы:

