**Потоки**

Посилання

<https://metanit.com/web/nodejs>

Потік є абстракція, що представляє собою послідовність байтів. Серед усіх потоків виділяють:

* потоки для запису;
* потоки для зчитування.

Зауважимо, що при створенні найпростішого сервера об’єкти request, response також є потоками:

* request - потік для читання;
* response - потік для запису.

|  |
| --- |
| const http = require("http");    http.createServer(function(**request, response**){    }).listen(3000); |

**Створення власних потоків для читання і запису**

Для створення потоків можна використати модуль fs і відповідні методи:

* **fs.createWriteStream(“ім’я файлу”) –** створюємо потік для запису (якщо файлу немає, то його буде створено)**;**
* **fs.createReadStream(“ім’я файлу”, “кодування”) –** створюємо потік для читання.

При зчитуванні усі дані розбиваються на частинки (chunks). При зчитуванні кожної частинки генерується подія **data**. За допомогою метода **on()** здійснюється аналіз події, де можна виконати операції над зчитаною частинкою даних.

|  |  |
| --- | --- |
| 1  2  3 | readableStream.on("data", function(chunk){      console.log(chunk);  }); |

|  |
| --- |
| **const fs = require(“fs”);**  //- - - - - - - - - - - - - - Запис - - - - - - - - - - - - - -  let writeableStream = **fs.createWriteStream**("hello.txt");  writeableStream.**write**("Привет мир!");  writeableStream.**write**("Продолжение записи \n");  writeableStream.**end**("Завершение записи");  //- - - - - - - - - - - - - - Зчитування - - - - - - - - - - - - - -  let readableStream = **fs.createReadStream**("hello.txt", "utf8");    readableStream.on("data", function(chunk){      console.log(chunk);  }); |

З’єднання потоків вводу і виводу. Pipe

За допомогою pipe є можливість з’єднати між собою потоки вводу і виводу. При цьому кожного разу частинки даних при зчитуванні одразу передаються у потік для запису. Так, наприклад, без використання pipe копіювання необхідно здійснити поступово записуючи наступну порцію зчитаних даних у потік виводу (у нову копію файлу).

|  |  |
| --- | --- |
| Без використання pipe | З використанням pipe |
| const fs = require("fs");   //- - - - Створюємо потік для читання - - - -  let readableStream = fs.createReadStream("hello.txt", "utf8");    //- - - Створюємо потік для запису - - - - - - -  let writeableStream = fs.createWriteStream("some.txt");   //- - - При надходженні наступних зчитаних даних записуємо у файл - - -  readableStream.on("data", function(chunk){      writeableStream.write(chunk);  }); | const fs = require("fs");   //- - - - Створюємо потік для читання - - - -  let readableStream = fs.createReadStream("hello.txt", "utf8");  //- - - Створюємо потік для запису - - - - - - -  let writeableStream = fs.createWriteStream("some2.txt");    //- - - З’єднуємо потік для читання і потік для запису - - - -  readableStream**.pipe**(writeableStream); |

Приклад з архівацією файлів

const fs = require("fs");

const zlib = require("zlib"); //Модуль для створення архіву Gzip

 //- - - - Створюємо потік для читання - - - -

let readableStream = fs.createReadStream("hello.txt", "utf8");

//- - - Створюємо потік для запису - - - - - - -

let writeableStream = fs.createWriteStream("hello.txt.gz");

//- - - Створюємо об’єкт **gzip** - - - - - -

let gzip = zlib.createGzip();

//- - - З’єднуємо : потік\_для\_читання 🡪 стиснення\_даних 🡪 запис\_у\_файл - - - - - -

readableStream.**pipe**(gzip).**pipe**(writeableStream);

**Потоки і надсилання клієнту файлів**

|  |  |
| --- | --- |
| Без використання pipe | Передача з використанням pipe |
| onst http = require("http");  const fs = require("fs");    http.createServer(function(request, response){        console.log(`Запрошенный адрес: ${request.url}`);      if(request.url.startsWith("/public/")){            // получаем путь после слеша          var filePath = request.url.substr(1);  **fs.readFile**(filePath, function(error, data){                if(error){                  response.statusCode = 404;                  response.end("Resourse not found!");              }              else{                  response.setHeader("Content-Type", "text/html");  **response.end(data);**              }          })      }      else{          response.end("Файл недоступний");      }  }).listen(3000); | const http = require("http");  const fs = require("fs");    http.createServer(function(request, response){        if(request.url=="/some.doc"){        response.writeHead(200, {"Content-Type":"text/html"});        fs.createReadStream("some.html").**pipe**(response);      }      else{          response.end("Файл недоступний");      }    }).listen(3000) |