



**Министерство науки и высшего образования Российской
Федерации
Федеральное государственное бюджетное образовательное
учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 3-4

Тема: Знакомство с форматом JSON и запуском сервера

Студент Унтилова А.О.

Группа ИУ7-56

Оценка (баллы) _____

Преподаватель Попов А.Ю.

Москва.
2020 г.

Цель: Ознакомиться с форматом JSON, научиться получать объекты из строки JSON, работать с файлами, запускать сервер, получать HTML страницы и работать с формами.

Лабораторная работа №3.

Задание 1.

С клавиатуры считывается число N. Далее считывается N строк. Необходимо создать массив и сохранять в него строки только с четной длиной. Получившийся массив необходимо преобразовать в строку JSON и сохранить в файл.

```
"use strict";

const readlineSync = require('readline-sync');
const fs = require("fs");

const n = readlineSync.question("Input N: ");

if (n <= 0){
    console.log("N должно быть > 0!");
    return;
}

let mas_str = [];

for (let i = 0; i < n; i++){
    let str = readlineSync.question("Input string: ");
    if (str.length % 2 == 0){
        mas_str.push(str);
    }
}

const result = JSON.stringify(mas_str);

const nameString = "z1.txt";

fs.writeFileSync(nameString, result);

console.log("Create File OK");
```

```
Input N: 3
Input string: aa
Input string: sss
Input string: vvvv
Create File OK
```

```
z1.txt
1 ["aa", "vvvv"]
```

Задание 2

Необходимо считать содержимое файла, в котором хранится массив строк в формате JSON. Нужно вывести только те строки на экран, в которых содержатся только гласные буквы.

```
const fs = require("fs");
```

```

const readlineSync = require('readline-sync');

const nameString = "z2.txt";

const contentString = fs.readFileSync(nameString, "utf8");

console.log(contentString);

const obj = JSON.parse(contentString);

let mas_vowel = "a A e E i I o O u U y Y";

console.log("Result: ");
for (let i = 0; i < obj.length; i++){
    let flag = 0;
    let s = obj[i];
    for (let j = 0; j < s.length && flag == 0; j++){
        if (!mas_vowel.includes(s[j])) {
            flag = 1;
        }
    }
    if(!flag){
        console.log(obj[i]);
    }
}

```

```

["aaaa","oo","llll","pppp","eeee"]
Result:
aaaa
oo
eeee

```

Задание 3

С клавиатуры считывается строка - название расширения файлов. Далее считывается строка - адрес папки. Необходимо перебрать все файлы в папке и вывести содержимое файлов, у которых расширение совпадает с введенным расширением.

```

"use strict";

const readlineSync = require('readline-sync');
const fs = require("fs");

const name_folder = readlineSync.question("Input name of folder: ");
const type_files = readlineSync.question("Input type of files: ");

const arr = fs.readdirSync(name_folder);    // ПОЧИТАТЬ ПРО ФУНКЦИЮ!!

console.log("Length: " + arr.length);

```

```

for(let i = 0; i < arr.length; i++) {
  const fileName = arr[i];
  let type = fileName.split(".").pop();    // ПОЧИТАТЬ!
  if (type == type_files){
    const contentString = fs.readFileSync(fileName, "utf8");
    console.log(contentString);
  }
}

```

```

Input name of folder: ./
Input type of files: txt
Length: 36
[{"name":"Anna","age":20,"group":"iu7-56"}, {"name":"Kate","age":18,"group":"mt12-21"}, {"name":"Polina","age":20,"group":"rk4-43"}, {"name":"Tanya","age":21,"group":"iu3-23"}, {"name":"Rita","age":23,"group":"rl6-21"}, {"name":"Vera","age":19,"group":"sm4-16"}]
I am text from 51!
I am text from 52!
Hello!

```

Задание 4

Дана вложенная структура файлов и папок. Все файлы имеют расширение "txt". Необходимо рекурсивно перебрать вложенную структуру и вывести имена файлов, у которых содержимое не превышает по длине 10 символов.

```

const readlineSync = require('readline-sync');
const fs = require("fs");

const name_folder = "./test_4";
const type_files = "txt";

function read_folder(name_folder)
{
  const arr = fs.readdirSync(name_folder);    // ПОЧИТАТЬ ПРО ФУНКЦИЮ!!!!!!!!!!!!!!!!!!!!
  ?????????????????
  for(let i = 0; i < arr.length; i++) {
    const fileName = arr[i];
    let flag = fileName.indexOf('.txt')    // поиск подстроки в строке; не найдено = -
1
    if (flag > 0){
      if (fs.existsSync(name_folder + "/" + fileName)) {
        const content = fs.readFileSync(name_folder + "/" + fileName, "utf8");
        if (content.length <= 10){
          console.log(name_folder + "/" + fileName);
        }
      }
      else {
        console.log("Файл " + fileName + " не найден.");
      }
    }
  }
}

```

```

    }
    else{
        read_folder(name_folder + "/" + fileName);
    }
}
}
read_folder(name_folder);

```

```

./test_4/new_1/yesyes11.txt
./test_4/new_2/new_3/yes3.txt
./test_4/new_2/yes2.txt
./test_4/yes1.txt
./test_4/yes11.txt

```

Задание 5

С клавиатуры считывается число N. Далее считывается N строк - имена текстовых файлов. Необходимо склеить всё содержимое введенных файлов в одну большую строку и сохранить в новый файл.

```

const readlineSync = require('readline-sync');
const fs = require("fs");

function write_in_file(contentString){
    const fs = require("fs");
    const fileName = "result_5.txt";
    fs.writeFileSync(fileName, contentString);
    console.log("Запись в файл прошла успешно!");
}

function input_f_names(){
    console.log("Введите количество файлов: ")
    const N = readlineSync.questionInt(); // проблемы с русским текстом в приглашении !!
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    let f_names = [];
    let fileName;
    for (let i = 0; i < N; i++){
        console.log("Введите имя текстового файла: ")
        fileName = readlineSync.question();
        f_names[i] = fileName;
    }
    return f_names;
}

function read_files(f_names){
    let f_string_all = "";
    for (let i = 0; i < f_names.length; i++){
        if (fs.existsSync(f_names[i])) {
            let file_string = fs.readFileSync(f_names[i], "utf8");
            f_string_all += file_string;
        }
    }
}

```

```

        f_string_all += "\n";
    } else {
        console.log("Файл " + f_names[i] + " не найден.");
    }
}
return f_string_all;
}

let f_names = input_f_names(); // файлы test_51.txt, test_52.txt
let f_string = read_files(f_names);
write_in_file(f_string);

```

```

Введите количество файлов:
2
Введите имя текстового файла:
test_51.txt
Введите имя текстового файла:
test_52.txt
Запись в файл прошла успешно!

```

Задание 6

Написать код, который позволяет определить максимальный возможный уровень вложенности друг в друга полей в объекте, чтобы данный объект можно было преобразовать в строку формата JSON. Ответом является целое число.

```

function get_max_depth(global, nested, depth) {
    if (!nested) {
        nested = global;
    }
    nested.prop = {};
    JSON.stringify(global);
    console.log(global, depth);
    try{
        return get_max_depth(global, nested.prop, depth + 1);
    }
    catch(RangeError){
        return depth;
    }
}

let res = get_max_depth({}, null, 0);
console.log("Answer = " + res);

```

```

{ prop: { prop: { prop: [Object] } } } 859
{ prop: { prop: { prop: [Object] } } } 860
{ prop: { prop: { prop: [Object] } } } 861
{ prop: { prop: { prop: [Object] } } } 862
{ prop: { prop: { prop: [Object] } } } 863
{ prop: { prop: { prop: [Object] } } } 864
Answer = 864

```

Задание 7

Из файла считывается строка в формате JSON. В этой строке информация об объекте, в котором находится большое количество вложенных друг в друга полей. Объект представляет из себя дерево. Необходимо рекурсивно обработать дерево и найти максимальную вложенность в дереве. Необходимо вывести на экран ветку с максимальной вложенностью.

```
const fs = require("fs");

function find_max(arr){
    let m = arr[0];
    for (let i = 0; i < arr.length; i++){
        if (arr[i] > m){
            m = arr[i]
        }
    }
    return m;
}

function getProp(o, h, mas_h, way, max_h) {
    h += 1;
    for(let prop in o) {
        if(typeof(o[prop]) === 'object') { // проверяет равенство без приведения типов

            way.push(o.field);
            getProp(o[prop], h, mas_h, way, max_h);
        } else {
            way.push(o.field);
            way = way.slice(0, way.length - h - 1);
            if (max_h > 0 && max_h == h){
                way.push(o.field);
                console.log(way);
            }
            else if (max_h < 0){
                mas_h.push(h);
            }
        }
    }
    return (mas_h);
}

function proc(obj) {
    let h = 0;
    let max_h = -1;
    let mas_h = [];
    let way = [];
    mas_h = getProp(obj, h, mas_h, way, max_h);
    max_h = find_max(mas_h);
    console.log("Max height = " + max_h);
    h = 0;
    way = []
    getProp(obj, h, mas_h, way, max_h);
}
```

```
let stringJSON = fs.readFileSync("test_7.txt", "utf8");

let global = JSON.parse(stringJSON);

proc(global);
```

```
Max height = 7
[
  'a', 'c', 'k',
  'm', 't', 'o',
  'z'
]
```

Лабораторная работа №4.

Задание 1

Запустить сервер. Реализовать на сервере функцию для сравнения трёх чисел и выдачи наибольшего из них. Реализовать страницу с формой ввода для отправки запроса на сервер.

```
const fs = require("fs");

const express = require("express");

const app = express();
const port = 2000;
app.listen(port);
console.log("My server on port " + port);

app.get("/me/page", function(request, response) {
  const nameString = request.query.p;
  if (fs.existsSync(nameString)) {
    const contentString = fs.readFileSync(nameString, "utf8");
    response.end(contentString);
  } else {
    const contentString = fs.readFileSync("bad.html", "utf8");
    response.end(contentString);
  }
});

app.get("/compare", function(request, response) {
  const a = request.query.a;
  const b = request.query.b;
  const c = request.query.c;
  const aInt = parseInt(a);
  const bInt = parseInt(b);
  const cInt = parseInt(c);
  const sInt = Math.max(aInt, bInt, cInt);
  const answerJSON = JSON.stringify({Max: sInt});
  response.end(answerJSON);
});
```


Задача 1

```
{ "Max": 17 }
```

Введите А

Введите В

Введите С

Задание 2

Запустить сервер. На стороне сервера должен храниться файл, внутри которого находится JSON строка. В этой JSON строке хранится информация о массиве объектов. Реализовать на сервере функцию, которая принимает индекс и выдает содержимое ячейки массива по данному индексу. Реализовать страницу с формой ввода для отправки запроса на сервер.

```
const filename = "2_arr.txt"

const fs = require("fs");

const express = require("express");

const app = express();
const port = 3006;
app.listen(port);
console.log("My server on port " + port);

app.get("/me/page", function(request, response) {
  const nameString = request.query.p;
  if (fs.existsSync(nameString)) {
    const contentString = fs.readFileSync(nameString, "utf8");
    response.end(contentString);
  } else {
    const contentString = fs.readFileSync("bad.html", "utf8");
    response.end(contentString);
  }
});

app.get("/get/by/index", function(request, response) {
  const index = request.query.index;
  let stringJSON = fs.readFileSync(filename, "utf8");
  let arr = JSON.parse(stringJSON);
  let answerJSON;
  if (index >= 0 && index < arr.length){
    answerJSON = JSON.stringify(arr[index], null, 4);
    console.log(arr[index]);
    console.log(answerJSON);
  }
});
```

```
    } else {  
        answerJSON = JSON.stringify("Index error.");  
    }  
    response.end(answerJSON);  
});
```

Задача 2

Введите индекс

Отправить запрос

```
{  
  "name": "Polina",  
  "age": 20,  
  "group": "rk4-43"  
}
```

Задание 3

Написать программу, которая на вход получает массив названий полей и адрес запроса (куда отправлять). Программа должна генерировать HTML разметку страницы, в которую встроена форма для отправки запроса.

```
"use strict";  
const readlineSync = require('readline-sync');  
  
const fs = require("fs");  
// readlineSync.setDefaultOptions({encoding: 'utf8'});  
const FILE_ERROR = -1;  
  
function read_file(filename){  
    const fs = require("fs");  
  
    if (fs.existsSync(filename)) {  
        const contentString = fs.readFileSync(filename, "utf8");  
        return contentString;  
    } else {  
        console.log("Файл " + filename + " не найден.");  
        return FILE_ERROR;  
    }  
}  
  
function write_file(contentString, filename){  
    const fs = require("fs");  
  
    fs.writeFileSync(filename, contentString);  
  
    console.log("Файл создан");  
}  
  
console.log("Введите количество полей: ")  
const N = readlineSync.questionInt();
```

```

let fields = [];
for (let i = 0; i < N; i++){
    console.log("Введите поле: ");
    let str = readlineSync.question();
    fields.push(str);
}
console.log("Введите адрес запроса: ");
const address = readlineSync.question();
let result = "";
let begin = read_file("3_begin.html");

if (begin == FILE_ERROR){
    console.log("Файл не может быть сгенерирован.");
    return;
}
result += begin + "\n";
result += "    <form method=\"GET\" action=\"\" + address + "\">" + "\n";

for (let i = 0; i < fields.length; i++){
    result += "        <p>Введите " + fields[i] + "</p>" + "\n";
    result += "        <input name=\"\" + fields[i] + "\" spellcheck=\"false\" autocomplete=\"off\">\n";
}

let end = read_file("3_end.html");
if (end == FILE_ERROR){
    console.log("Файл не может быть сгенерирован.");
    return;
}
result += end;
write_file(result, "3_result.html");

```

Задание 3

Введите field_1

Введите field_2

Отправить запрос

Задание 4

Запустить сервер. Реализовать на сервере функцию, которая принимает на вход числа А, В и С. Функция должна выдавать массив целых чисел на отрезке от А до В, которые делятся на С нацело.

```
const fs = require("fs");
```

```
const express = require("express");

const app = express();
const port = 2000;
app.listen(port);
console.log("My server on port " + port);

app.get("/me/page", function(request, response) {
  const nameString = request.query.p;
  if (fs.existsSync(nameString)) {
    const contentString = fs.readFileSync(nameString, "utf8");
    response.end(contentString);
  } else {
    const contentString = fs.readFileSync("bad.html", "utf8");
    response.end(contentString);
  }
});

app.get("/get/array", function(request, response) {
  const a = request.query.a;
  const b = request.query.b;
  const c = request.query.c;
  const aInt = parseInt(a);
  const bInt = parseInt(b);
  const cInt = parseInt(c);
  let arr = [];
  for (let i = aInt; i <= bInt; i++){
    if (i % cInt == 0)
      arr.push(i);
  }
  const answerJSON = JSON.stringify(arr);
  response.end(answerJSON);
});
```

Задача 4

[3,6,9,12,15,18]

Введите A

Введите B

Введите C

Вывод: Во время выполнения лабораторной работы я ознакомилась с форматом JSON, научилась получать объекты из строки JSON, работать с файлами, запускать сервер, получать HTML страницы и работать с формами.