|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 № 2**

Работа с веб-сервером.

|  |  |
| --- | --- |
| **Студент** Жигалкин Д.Р  **Группа** ИУ7-55Б  **Оценка (баллы) \_\_\_\_\_\_\_\_** |  |

Москва.

2020 г.

**Цель**: Ознакомиться с созданием файла с зависимостями. Изучить формат JSON, запустить веб-сервер.

***Task\_3***

### Задание 1

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

"use strict";

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

const fs = require("fs");

let n = readlineSync.question("Input n: ");

if (n <= 0)

    return;

const arr = [];

for (let j = 0; j < n; j++)

{

    let value = readlineSync.question("Input string: ");

    if (value.length % 2 == 0)

    {

        arr.push(value);

    }

}

const jsonString = JSON.stringify(arr);

const nameString = "test.txt";

fs.writeFileSync(nameString, jsonString);

console.log("Create File OK");

console.log(jsonString);

### 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.

"use strict";

const fs = require("fs");

const nameString = "test.txt";

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

const obj = JSON.parse(contentString);

console.log(obj.filter(s => s.search(/[aeiou]+$/i) != -1));

### 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 fs = require("fs");

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

let folder = readlineSync.question("Input folder: ");

//const folder = "C:/Users/zhigalkin/OneDrive/Desktop/t";

let extensions = readlineSync.question("Input extensions: ");

const arr = fs.readdirSync(folder);

function getExtension(filename)

{

    var i = filename.lastIndexOf('.');

    return (i < 0) ? '' : filename.substr(i + 1);

}

for (let i = 0; i < arr.length; i++)

 {

    const fileName = arr[i];

    if (extensions.indexOf(getExtension(fileName)) != -1)

    {

        const directory = folder + "/" + fileName

        let fileContent = fs.readFileSync(directory, "utf8");

        console.log(fileContent);

    }

}

### Задание 4

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

var fs = require('fs');

function getFiles(dir, files\_)

{

    files\_ = files\_ || [];

    var files = fs.readdirSync(dir);

    for (var i in files)

    {

        var name = dir + '/' + files[i];

        if (fs.statSync(name).isDirectory())

        {

            getFiles(name, files\_);

        }

        else

        {

            let fileContent = fs.readFileSync(name, "utf8");

            if (fileContent.length <= 10)

            {

                files\_.push(name);

            }

        }

    }

    return files\_;

};

console.log(getFiles('C:/Users/zhigalkin/OneDrive/Desktop/t'));

### 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 fs = require('fs');

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

let n = readlineSync.question("Input n: ");

if (n <= 0)

    return;

const arrayNamefiles = [];

const folder = "C:/Users/zhigalkin/OneDrive/Desktop/t";

const arr = fs.readdirSync(folder);

function InputFileName(arrayNamefiles)

{

    for (let j = 0; j < n; j++)

    {

        let nameFile = readlineSync.question("Input name file: ");

        arrayNamefiles.push(nameFile);

    }

    console.log(arrayNamefiles);

}

function JoinContent(arrayAllfiles, arrayNamefiles)

{

    var string = "";

    for (let j = 0; j < arrayAllfiles.length; j++)

    {

        var fileName = arrayAllfiles[j];

        if (arrayNamefiles.indexOf(fileName) != -1)

        {

            const directory = folder + "/" + fileName

            let fileContent = fs.readFileSync(directory, "utf8");

            string += fileContent;

        }

    }

    console.log(string);

    return string;

}

function WriteToFile(string)

{

    const nameString = "test1.txt";

    fs.writeFileSync(nameString, string);

    console.log("Create File OK");

}

InputFileName(arrayNamefiles);

var string = JoinContent(arr, arrayNamefiles);

WriteToFile(string);

### 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.

let user =

{

    firstUser:

    {

      age:

      {

        a2level:

        {

            a3level:

            {

                a4level: 100

            }

        }

      }

    },

    testLevels:

    {

      is2level:

      {

          a2level:

          {

              a3level: 100

          }

      }

    }

}

  let level = 0;

  var maxLevel = 0;

  function rec(user)

  {

    for (let key in user)

    {

      if (typeof (user[key]) == "object")

      {

        level++;

        rec(user[key]);

      }

      else

      {

          if (level > maxLevel)

          {

              maxLevel = level;

          }

          level = 0

          console.log(key.toUpperCase())

      }

    }

  }

  rec(user);

  console.log(maxLevel)

### 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.

let user =

{

    firstUser:

    {

      age:

      {

        a2level:

        {

            a3level:

            {

                a4level: 100

            }

        }

      }

    },

    testLevels:

    {

      is2level:

      {

          a2level:

          {

              a3level: 100

          }

      }

    }

}

  let level = 0;

  var maxLevel = 0;

  var maxBranch = "";

  var currentBranch = "";

  function rec(user)

  {

    for (let key in user)

    {

        currentBranch += " " + key;

      if (typeof (user[key]) == "object")

      {

        level++;

        rec(user[key]);

      }

      else

      {

          if (level > maxLevel)

          {

              maxLevel = level;

              maxBranch = currentBranch;

          }

          currentBranch = ""

          level = 0

          console.log(key.toUpperCase())

      }

    }

  }

  rec(user);

  console.log(maxLevel)

  console.log(maxBranch)

***Task\_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.

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Add information</title>

</head>

<body>

    <h1>infromation: </h1>

    <form method="GET" action="/calculate/max">

        <p>Введите эл.почту</p>

        <input name="email" spellcheck="false" autocomplete="off">

        <p>Введите фамилию</p>

        <input name="surname" spellcheck="false" autocomplete="off">

        <p>Введите телефон</p>

        <input name="phone" spellcheck="false" autocomplete="off">

        <br>

        <br>

        <input type="submit" value="Отправить запрос">

    </form>

</body>

</html>

"use strict";

const fs = require("fs");

const express = require("express");

const app = express();

const port = 5015;

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("/calculate/max", function(request, response)

{

    var answerJSON = JSON.stringify({result: "badInput"});

    const a = request.query.first;

    const b = request.query.second;

    const c = request.query.third;

    const aInt = parseInt(a);

    const bInt = parseInt(b);

    const cInt = parseInt(c);

    if (aInt == null || bInt == null || cInt == null)

    {

        response.end(answerJSON);

    }

    else

    {

        const maxElem = Math.max(aInt, bInt, cInt);

        answerJSON = JSON.stringify({result: maxElem});

        response.end(answerJSON);

    }

});

### 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.

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Find object in JSON</title>

</head>

<body>

    <h1>Find object in JSON</h1>

    <form method="GET" action="/find/object">

        <p>Введите id</p>

        <input name="id" spellcheck="false" autocomplete="off">

        <br>

        <br>

        <input type="submit" value="Отправить запрос">

    </form>

</body>

</html>

"use strict";

const fs = require("fs");

const express = require("express");

const app = express();

const port = 5015;

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("/find/object", function(request, response)

{

    const id = parseInt(request.query.id);

    if (id == null || id < 0)

    {

        response.end(JSON.stringify({result: "badInput"}));

    }

    else

    {

        var contentString = fs.readFileSync("JSON\_for\_task4.2.txt", "utf8");

        var objects = [];

        objects = JSON.parse(contentString);

        console.log(objects.length);

        if (id + 1 > objects.length)

        {

            response.end(JSON.stringify({result: "indexOutofRange"}));

        }

        else

        {

            response.end(JSON.stringify(objects[id]));

        }

    }

});

### 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.

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Get array</title>

</head>

<body>

    <form method="GET" action="/calculate/special">

        <p>Введите url</p>

        <input name="url" spellcheck="false" autocomplete="off">

        <p>Введите имя поля</p>

        <input name="name" spellcheck="false" autocomplete="off">

        <br>

        <br>

        <input type="submit" value="Отправить запрос">

    </form>

</body>

</html>

"use strict";

const fs = require("fs");

const express = require("express");

const app = express();

const port = 5015;

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("/calculate/special", function(request, response)

{

    var answerJSON = JSON.stringify({result: "badInput"});

    let first = '<!DOCTYPE html><html><head><meta charset="UTF-8"><title>Get array</title></head><body>';

    let third = '</form></body></html>'

    const url = request.query.url;

    const name = request.query.name;

    if (url == null || name == null)

    {

        response.end(answerJSON);

    }

    else

    {

        let second = '<form method="GET" action=' + '"' + url + '"' +  '>' +  '<p>Введите ' + name + '</p>';

        let sec1 = '<input name=' + name +  ' spellcheck="false" autocomplete="off">';

        let sec2 = '<input type="submit" value="Отправить запрос">';

        const jsonString = first + second + sec1 + sec2 + third;

        const nameString = "html.html";

        fs.writeFileSync(nameString, jsonString);

        answerJSON = JSON.stringify({res: 0});

        response.end(answerJSON);

    }

});

### Задание 4

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

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Get array</title>

</head>

<body>

    <h1>Get array [a, b] where (elem % c == 0)</h1>

    <form method="GET" action="/calculate/special">

        <p>Введите A</p>

        <input name="first" spellcheck="false" autocomplete="off">

        <p>Введите B</p>

        <input name="second" spellcheck="false" autocomplete="off">

        <p>Введите C</p>

        <input name="third" spellcheck="false" autocomplete="off">

        <br>

        <br>

        <input type="submit" value="Отправить запрос">

    </form>

</body>

</html>

"use strict";

const fs = require("fs");

const express = require("express");

const app = express();

const port = 5015;

app.listen(port);

console.log("My server on port " + port);

app.get("/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("/calculate/special", function(request, response)

{

    var answerJSON = JSON.stringify({result: "badInput"});

    const url = request.query.first;

    const b = request.query.second;

    const aInt = parseInt(a);

    const bInt = parseInt(b);

    const cInt = parseInt(c);

    if (aInt == null || bInt == null || cInt == null)

    {

        response.end(answerJSON);

    }

    else

    {

        const maxElem = Math.max(aInt, bInt, cInt);

        answerJSON = JSON.stringify(getAnswer(aInt, bInt, cInt));

        response.end(answerJSON);

    }

});

function getAnswer(a, b, c)

{

    var answer = [];

    for (var i = a; i <= b; i++)

    {

        if (i % c == 0)

            answer.push(i);

    }

    return answer;

}

**Вывод:** Я ознакомился с созданием файла с зависимостями, изучил формат json и запустил веб-сервер.