Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

Лабораторная работа № 6

**«Реализация REST API»**

## Отчёт по дисциплине

**«Web-программирование»**

Пенза 2020

**Выполнили с-ты гр. 16ВВ1:**

Ежов И.С.  
Екименко Д.В.

**Приняли:**

к.т.н. Дубравин А.В.  
Кармышева Н.С.

**Цель работы.**

Реализовать RESTful API.

**Задание.**

Необходимо доработать лабораторную работу №5 реализовав в ней поддержку REST API и реализовать два обработчика:

1. Получает запрос с названием животного и дополнительными параметрами и подбирает по этому запросу подходящее изображение из числа имеющихся;

2. По имени и параметрам изображения возвращает картинку.

|  |  |
| --- | --- |
| 5. | Животные: дельфин, дикобраз, дрозд. |

**Листинг.**

**Hello.html**

<html>

    <head>

        <meta charset="utf-8" />

        <script>

            let array = ["verblyud", "vol", "volk"];

            async function check() {

                var animalname = null;

                var imgwidth = null;

                var imgheight = null;

                var imgcolor = null;

                var imgvalue = null;

                animalname = animal.value;

                imgwidth = width.value;

                imgheight = height.value;

                imgcolor = color.value;

                imgvalue = value.value;

                let animanlsJSON = {

                    name: animalname,

                    width: imgwidth,

                    height: imgheight,

                    color: imgcolor,

                    value: imgvalue

                };

                let response = await fetch('./api/animal', {

                method: 'POST',

                headers: {

                    'Content-Type': 'application/json'

                },

                body: JSON.stringify(animanlsJSON)

                });

                const json = await response.json();

                console.log(json);

                if (json.error != null)

                {

                    var pjson = JSON.stringify(json);

                    document.write(pjson);

                }

                else

                {

                    document.write('<img id=myImg src= "">');

                    var img = json.image;

                    document.getElementById("myImg").src = "images/"+img+".jpg";

                    delete json.\_id;

                    delete json.\_\_v;

                    var pjson = JSON.stringify(json);

                    document.write(pjson);

                }

            }

        </script>

    </head>

    <body>

        <a>Название животного</a><br>

        <input id="animal" rows="1" cols="10"></input><p></p>

        <a>Значение</a><br>

        <input id="value" rows="1" cols="10"></input><p></p>

        <a>Ширина</a><br>

        <input id="width" rows="1" cols="10"></input><p>

        <a>Высота</a><br>

        <input id="height" rows="1" cols="10"></input><p>

        <a>Цвет</a><br>

        <input id="color" rows="1" cols="10"></input><p>

        <input type="submit" value="Найти изображение" onclick="check()"/>

    </body>

</html>

**index.js**

const express = require("express");

const mongoose = require("mongoose");

const bodyParser = require("body-parser");

const app = express();

mongoose.connect("mongodb://localhost/animal-db",

{

    useNewUrlParser: true,

    useUnifiedTopology: true

});

app.use('/images',express.static('images'));

app.use(bodyParser.json());

app.use("/api", require("./api"));

app.listen(8888, () =>{

    console.log("server start")

});

**api.js**

var fs = require("fs");

const express = require("express");

const router = express.Router();

const Animal = require("./animal")

let jsonbody = null;

let imgerror = null;

let backerr = null;

router.get("", (req, res)=>{

    var body = fs.readFileSync('Hello.html');

    res.writeHead(200, {"Content-Type": "text/html"});

    res.write(body);

    res.end();

});

router.get("/animal", (req, res)=>{

    console.log(jsonbody);

    res.send(jsonbody);

});

router.post("/animal", (req, res)=>{

    console.log(req.body);

    var name = req.body.name;

    var height = req.body.height;

    var width = req.body.width;

    var color = req.body.color;

    var value = req.body.value;

    if (width == "")

    {

        width = Math.floor(Math.random() \* 2000);

    }

    if (height == "")

    {

        height = Math.floor(Math.random() \* 2000);

    }

    if (color == "")

    {

        var coloring = "abcdef0123456789";

        var color = "";

        for (var i = 0; i < 6; i++)

            color += coloring.charAt(Math.floor(Math.random() \* coloring.length));

    }

    Animal.findOne({name: name, value: value})

    .then(animal => {

        if (animal != null)

        {

            Animal.findOne({width: width, height: height})

            .then(animal => {

                if (animal != null)

                {

                    Animal.findOne({color: color})

                    .then(animal => {

                        if (animal != null)

                        {

                            jsonbody = animal;

                            res.redirect("./animal");

                        }

                        else

                        {

                            imgerror = {error: "Color error", name: name, value: value, width: width, height: height, color: color};

                            res.send(imgerror);

                        }

                    });

                }

                else

                {

                    imgerror = {error: "Size error", name: name, value: value, width: width, height: height};

                    res.send(imgerror);

                }

            });

        }

        else

        {

            imgerror = {error: "Name error", name: name, value: value};

            res.send(imgerror);

        }

    });

});

module.exports = router;

**animal.js**

const mongoose = require("mongoose");

const Schema = mongoose.Schema;

const AnimalSchema =  new Schema({

    name: String,

    width: Number,

    height: Number,

    color: String,

    image: String ,

    value: Number

});

const Animal = mongoose.model("animal", AnimalSchema);

module.exports = Animal;

**Результат работы.**

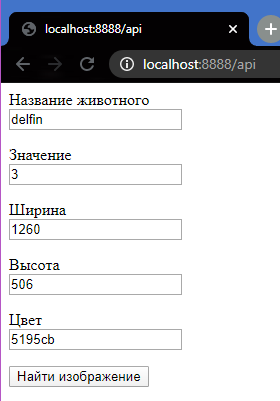


Рисунок 1 – Ввод правильных параметров

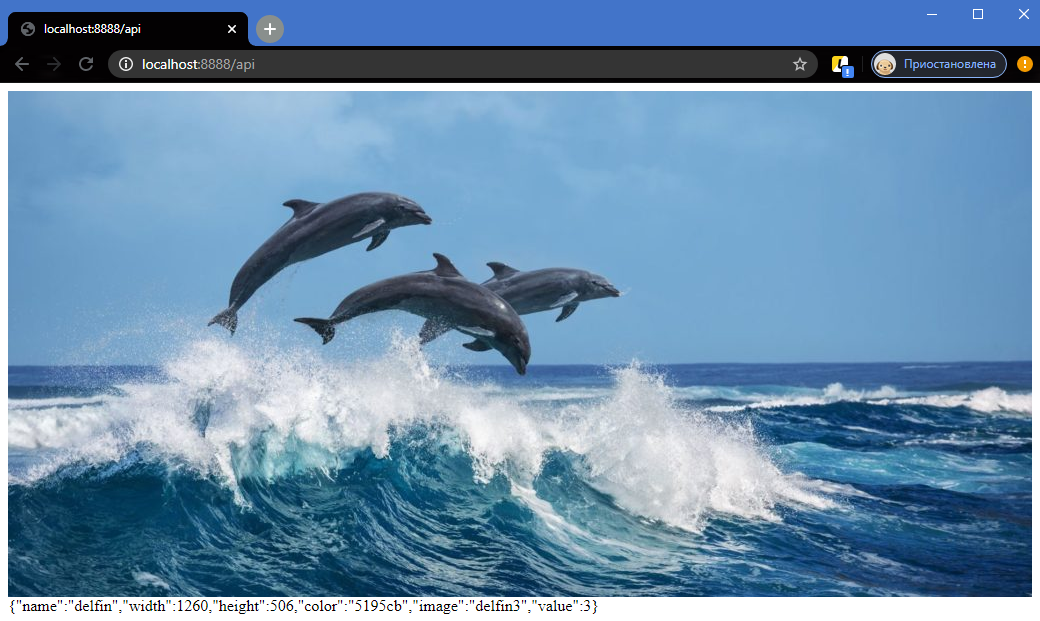


Рисунок 2 – Результат работы

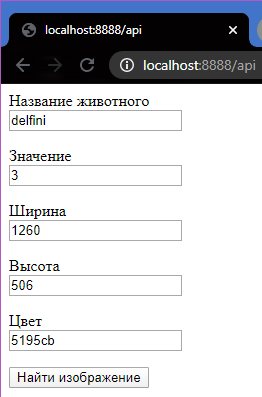


Рисунок 3 – Ввод с ошибкой в имени

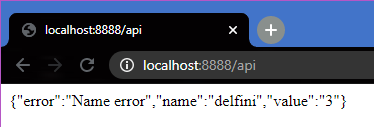


Рисунок 4 – Ошибка имени

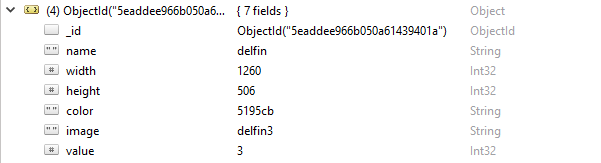


Рисунок 5 – Запись в БД

**Вывод.**

Научились реализовывать RESTful API.