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. 16ВВ2:**

Анастасин В.Д.  
Прокуров Д.А.

**Приняли:**

к.т.н. Дубравин А.В.  
Кармышева Н.С.

**Цель работы.**

Реализовать REST 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.

|  |  |
| --- | --- |
| **Вариант** | **Животные** |
| 4 | Гиена, газель, геккон |

**Листинг.**

**startPage.html**

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

    </head>

    <body>

        <a>Введите данные для запроса:</a>

        <br>

        <b>Вид</b>

        <br>

        <input id="species" rows="1" cols="10"></input>

        <p></p>

        <b>Ширина</b>

        <br>

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

        <p></p>

        <b>Высота</b>

        <br>

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

        <p></p>

        <b>Цвет</b>

        <br>

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

        <p></p>

        <b>Номер</b>

        <br>

        <input id="number" rows="1" cols="10"></input>

        <p></p>

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

    </body>

    <script>

        async function find() {

            console.log("Find started");

            var imgspecies = null;

            var imgwidth = null;

            var imgheight = null;

            var imgcolor = null;

            var imgnumber = null;

            var imgimage = null;

            imgspecies = document.getElementById("species").value;

            imgwidth = document.getElementById("width").value;

            imgheight = document.getElementById("height").value;

            imgcolor = document.getElementById("color").value;

            imgnumber = document.getElementById("number").value;

            imgimage = "pic/" + String(imgspecies) + String(imgnumber) + ".jpg";

            let animalsJSON = {

                species: imgspecies,

                width: imgwidth,

                height: imgheight,

                color: imgcolor,

                image: imgimage,

                value: imgnumber

            };

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

                method: 'POST',

                headers :{

                    'Content-Type': 'application/json'

                },

                body: JSON.stringify(animalsJSON)

            });

            const json = await responce.json();

            console.log(json);

            // Ошибка

            if (json.error != null)

            {

                console.log(json.error);

                var pjson = JSON.stringify(json);

                document.write(pjson);

            }

            else

            {

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

                console.log(json.image);

                document.getElementById("reqImg").src = json.image;

                delete json.\_id;

                delete json.\_\_v;

                var pjson = JSON.stringify(json);

                document.write(pjson);

            }

        }

    </script>

</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

}).then(() => {

    console.log("Connection to BD Success!")

});

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

app.use(bodyParser.json());

app.use("/api",require("./api"));

app.listen(8888, ()=>{

    console.log("Server processing...");

});

**api.js**

const express = require("express");

var fs = require("fs");

const router = express.Router();

const Animal = require("./animal");

// Тело запроса

let jsonbody = null;

router.get("", (req, res) => {

    var page = fs.readFileSync("startPage.html");

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

    res.write(page);

    res.end();

})

router.get("/animal", (req, res)=>{

    console.log(jsonbody);

    res.send(jsonbody);

    // res.send({method: "GET"});

});

router.post("/animal", (req, res)=>{

    console.log("POST is reached");

    console.log(req.body);

    var species = req.body.species;

    var width = req.body.width

    var height = req.body.height;

    var color = req.body.color;

    var number = req.body.number

    let finderror = null;

    if (width == "") {

        width = 1;

    }

    if (height == ""){

        height = 1;

    }

    if (color == ""){

        color = "white";

    }

    Animal.findOne({species: species, number: number})

    .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

                        {

                            finderror = {error: "Color Error", species: species, number: number };

                            res.send(finderror);

                        }

                    });

                }

                else

                {

                    finderror = {error: "Size Error", species: species, number: number };

                    res.send(finderror);

                }

            });

        }

        else

        {

            finderror = {error: "Species Error", species: species, number: number };

            res.send(finderror);

        }

    });

});

// Технический метод для заполнения БД

// router.post("/animals", (req, res)=>{

//     animal.create(req.body)

//     .then(animal =>{

//         res.send(animal);

//     })

// });

module.exports = router;

**animal.js**

const mongoose = require("mongoose");

const Schema = mongoose.Schema;

const animalSchema = new Schema({

    species: String,

    width: Number,

    height: Number,

    color: String,

    image: String,

    value: Number

});

const animal = mongoose.model("animal",animalSchema);

module.exports = animal;

**Результат работы.**

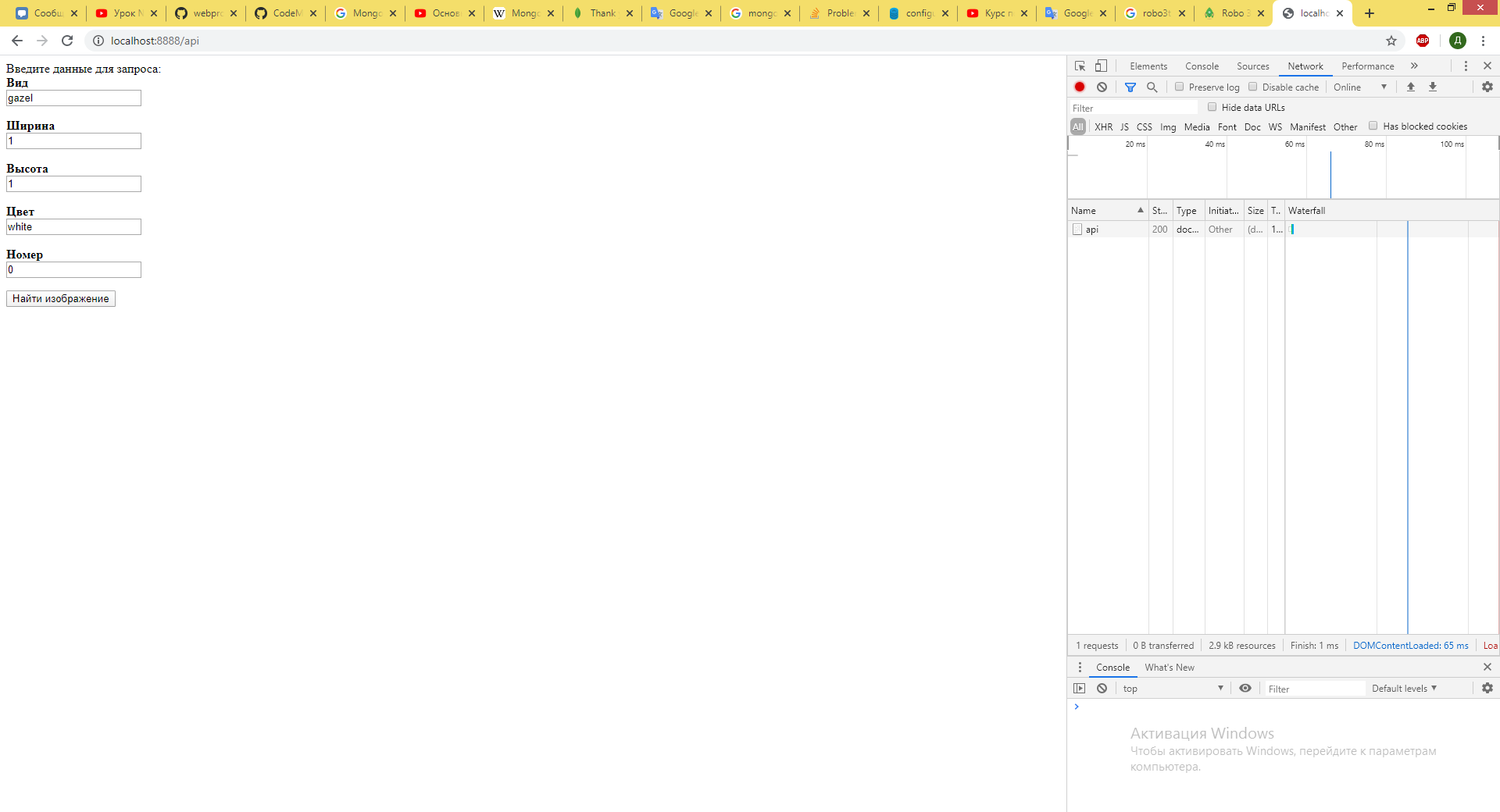


Рисунок 1 – API

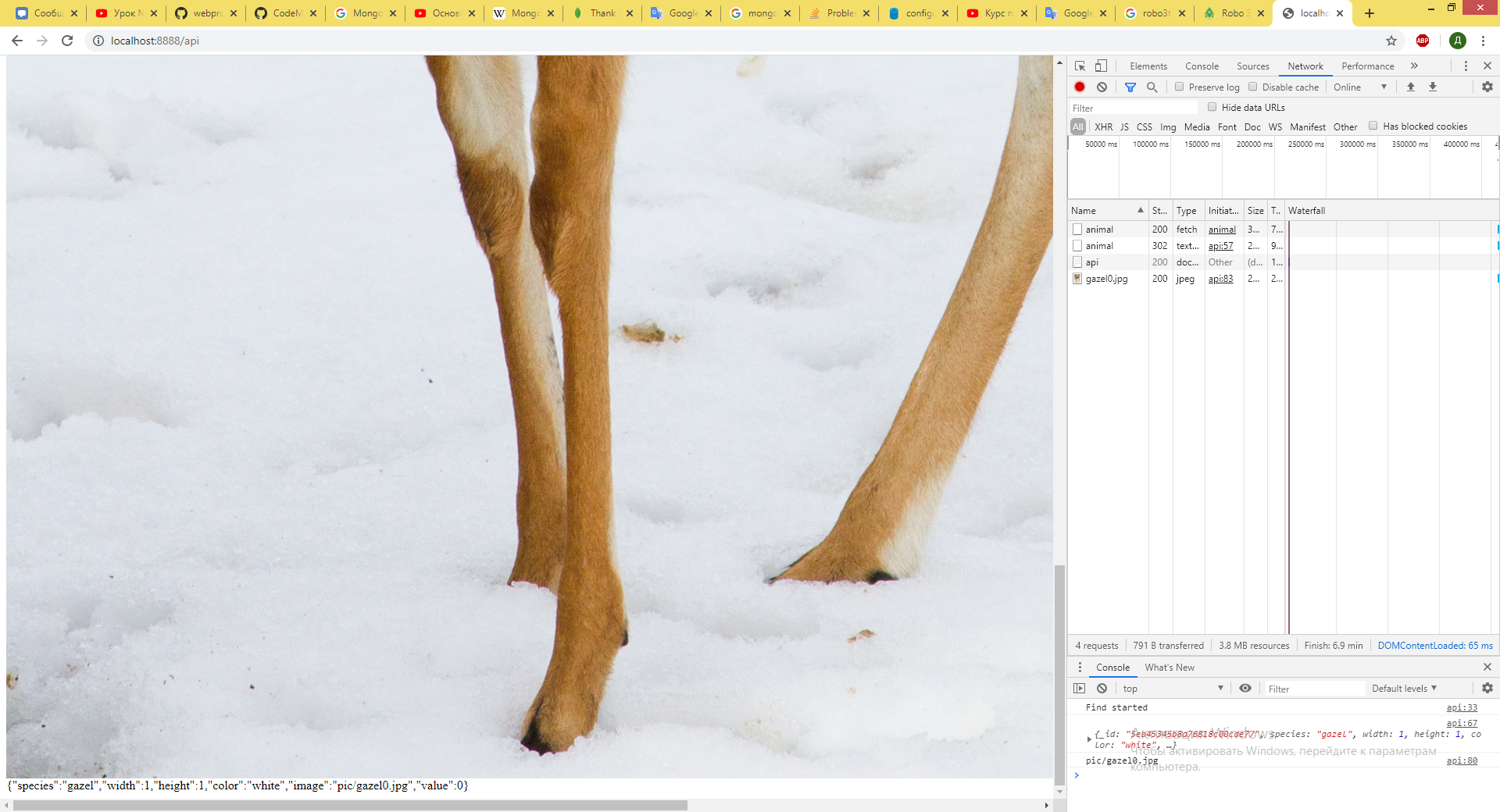


Рисунок 2 - Результат работы

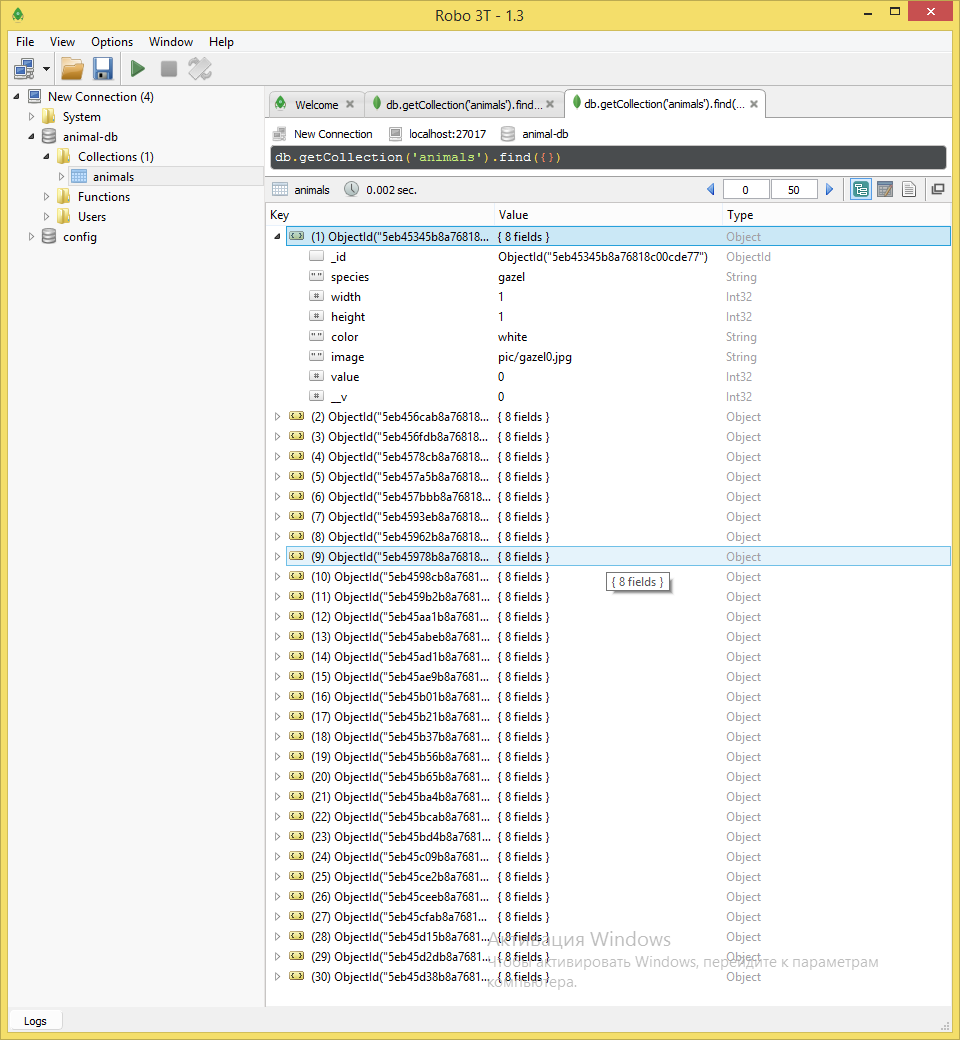


Рисунок 3 - База данных и соответствующая запись

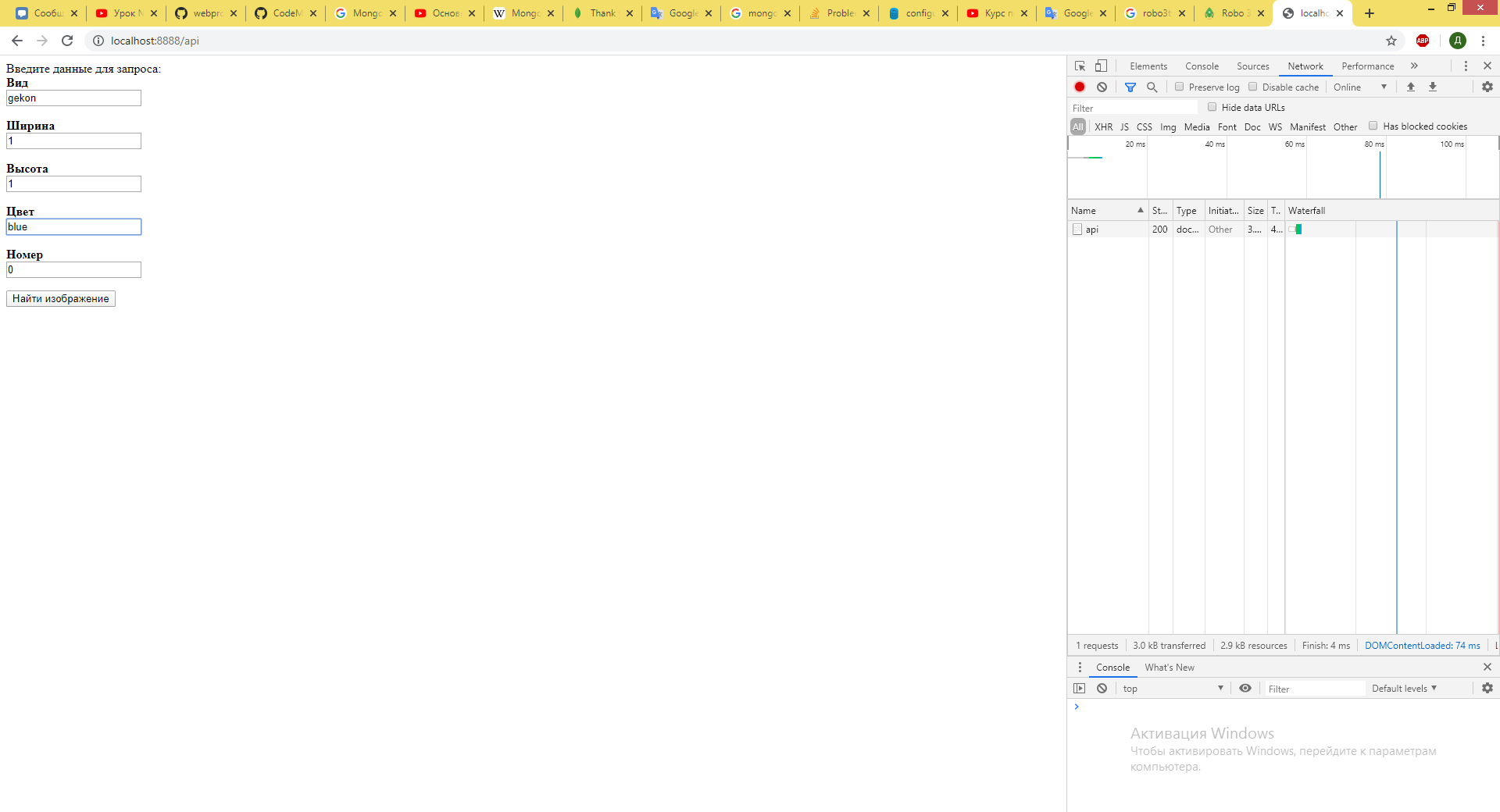


Рисунок 4 - ввод названия с ошибкой (правильно gekkon)

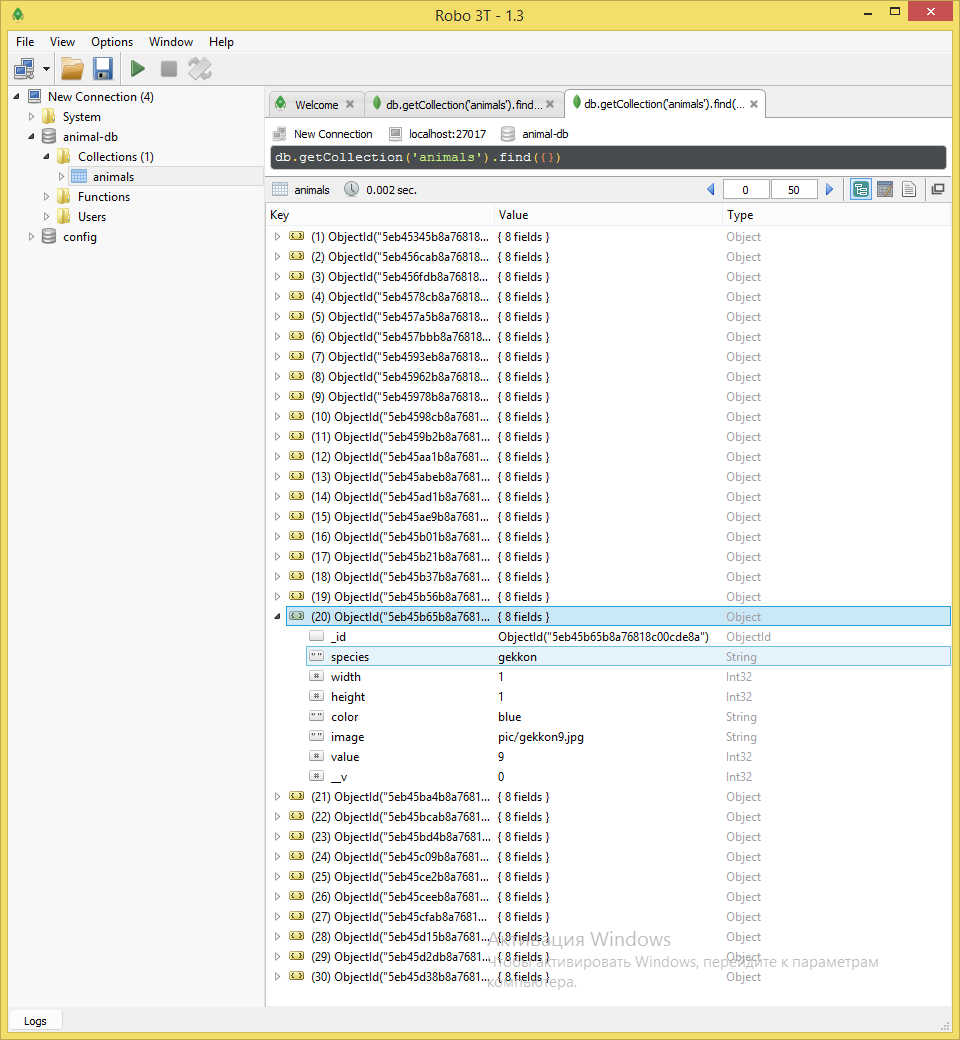


Рисунок 5 - правильная запись

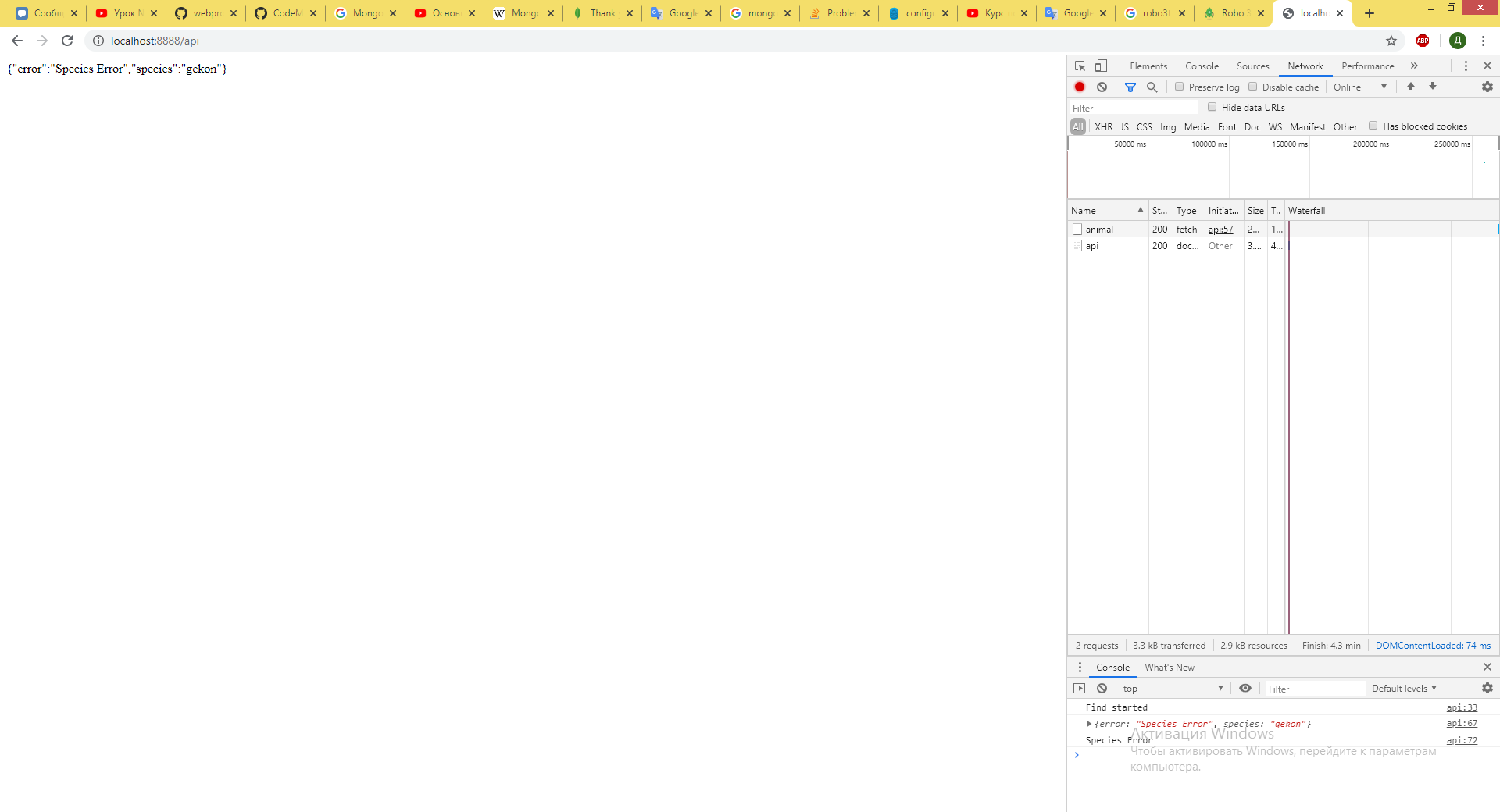


Рисунок 6 - результат работы

**Вывод.**

В ходе выполнения лабораторной работы научились реализовывать REST API.