**ОТЧЕТ**

по лабораторной работе №6

по курсу «Web-программирование»

на тему «. Реализация REST API»

Выполнили:

Студенты группы 16ВВ1

Соловьев А.А.

Чиркин К.Д.

Принял:

к.т.н., доцент Дубравин А.В.

Пенза 2020

Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

### Лабораторное 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;

В ответ на запрос сервер должен сформировать ответ, в котором перечислить переданные параметры, а недостающие заполнить случайными значениями (случайные значения должны выбираться из разумного диапазона), а также должен добавить поле с именем файла, соответствующего заданным параметрам.

### Ход работы.

#### Реализация обработчиков.

В данной работе начинаем использовать минималистичный и гибкий веб-фреймворк Express для приложений Node.js, предоставляющий обширный набор функций для мобильных и веб-приложений.

Для работы с ним необходимо выполнить его установку. Для этого необходимо открыть консоль в корне проекта и выполнить:

$ npm install express --save

После этого мы можем подключить его в свой проект путем добавления в файл:

const express = require('express');

#### Реализация обработчиков.

В данной работе начинаем использовать минималистичный и гибкий веб-фреймворк Express для приложений Node.js, предоставляющий обширный набор функций для мобильных и веб-приложений.

Выносим API в отдельный файл api.js и пишем в него обработчики. Файл принимает следующий вид:

const express = require('express');

const fs = require("fs");

const router = express.Router();

const colors = ['#FF0000', '#00FF00', '#FF8000']

const sizes = ['400x600', '800x600', '1280x960']

function getRandomInt(min, max) {

    return Math.floor(Math.random() \* (max - min + 1)) + min;

}

router.get('/', (req, res) => {

    let body = fs.readFileSync('index.html');

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

    res.write(body);

    res.end();

})

router.post('/getPicture', (req, res) => {

    let splitSizes;

    if (req.body.sizeSelector == 'Выберите размер') {

        splitSizes = sizes[getRandomInt(0, 2)].split('x');

    }

    else {

        splitSizes = req.body.sizeSelector.split('x');

    }

    let params = {

        name: req.body.text.trim().split(' '),

        color: req.body.colorSelector,

        size: {

            width: splitSizes[0],

            height: splitSizes[1],

        },

        image: 'default',

    }

    if (params.name[0] != 'kaban' &

        params.name[0] != 'kalmar' &

        params.name[0] != 'kakadu') {

        res.send({ name: 'not found' });

        return;

    }

    if (typeof params.name[1] === 'undefined') {

        params.name[1] = getRandomInt(0, 9);

    }

    if (params.color == 'Выберите цвет') {

        params.color = colors[getRandomInt(0, 2)];

    }

    params.image = 'images/' + params.name[0] + '/' + params.name[0] + '\_' + params.name[1] + '.jpg';

    res.send(params);

})

router.get('/picture/:name/:id/:color/:size', (req, res) => {

    let splitSizes;

    if (req.params.size == '400x600' || '800x600' || '1280x960') {

        splitSizes = req.params.size.split('x');

    }

    else { splitSizes = sizes[getRandomInt(0, 2)].split('x'); }

    let imageParams = {

        name: req.params.name,

        color: req.params.color,

        size: {

            width: splitSizes[0],

            height: splitSizes[1],

        },

        id: req.params.id

    }

    if (Number.isInteger(imageParams.id) || Number(imageParams.id) > 9) {

        imageParams.id = getRandomInt(0, 9);

    }

    if (imageParams.name != 'kaban' &&

        imageParams.name != 'kalmar' &&

        imageParams.name != 'kakadu') {

        res.send('not found');

        return;

    }

    if (imageParams.color != 'red' &&

        imageParams.color != 'green' &&

        imageParams.color != 'orange') {

        imageParams.color = colors[getRandomInt(0, 2)];

    }

    res.send({

        page:

            '<div style="background: ' + imageParams.color + ';">' +

            '<image width="' + imageParams.size.width +

            ' height="' + imageParams.size.height +

            '" src="http://localhost:8888/' + imageParams.name + '/' + imageParams.name + '\_' + imageParams.id + '.jpg">' +

            '</div>'

    });

})

module.exports = router;

Переписываем index.js до следующего состояния:

const express = require('express')

const app = express()

const port = 8888

const cors = require('cors');

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

app.use(cors());

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

app.use(bodyParser.json());

app.use(express.urlencoded());

app.use(require('./api'))

app.listen(port, () => {

    console.log(`Example app listening at http://localhost:${port}`);

})

Редактируем index.html для работы с API:

<html>

<head>

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

</head>

<body>

    <div>

        <form>

            <select id="colorSelector">'

                <option value="Выберите цвет" hidden>Выберите цвет</option>

                <option value="red">Красный</option>

                <option value="green">Зеленый</option>

                <option value="orange">Оранжевый</option>

            </select>

            <select id="sizeSelector">

                <option value="Выберите размер" hidden>Выберите размер</option>

                <option value="400x600">400x600</option>

                <option value="800x600">800x600</option>

                <option value="1280x960">1280x960</option>

            </select>

            <input pattern="[a-zA-Z]{1,8}\s?\d?" id="text" required>

            <input id="clickButton" type="submit" value="Получить картинку" onclick="RecieveAnswer(event)" />

        </form>

    </div>

</body>

<script>

    clickButton.addEventListener('keypress', RecieveAnswer);

    let buttonIsPressed = false;

    function RecieveAnswer(event) {

        event.preventDefault();

        let reqBody = {

            colorSelector: colorSelector.value,

            text: text.value,

            sizeSelector: sizeSelector.value,

        }

        fetch('http://localhost:8888/getPicture', {

            method: 'POST',

            headers: {

                'Content-Type': 'application/json'

            },

            body: JSON.stringify(reqBody)

        })

            .then(response => response.json())

            .then(result => {

                console.log(result);

                if (result.name === 'not found') {

                    text.value = result.name;

                    sizeSelector.value = 'Выберите размер';

                    colorSelector.value = 'Выберите цвет';

                    return;

                }

                text.value = result.name.join(' ');

                if (result.color === '#FF0000') { colorSelector.value = 'red' }

                if (result.color === '#00FF00') { colorSelector.value = 'green' }

                if (result.color === '#FF8000') { colorSelector.value = 'orange' }

                if (result.size.width === '400') { sizeSelector.value = '400x600' }

                if (result.size.width === '800') { sizeSelector.value = '800x600' }

                if (result.size.width === '1280') { sizeSelector.value = '1280x960' }

                if (!buttonIsPressed) {

                    let previewButton = document.createElement('button');

                    let buttonContent = document.createTextNode('Отобразить картинку');

                    previewButton.setAttribute('onclick', 'showPicture(event)')

                    previewButton.appendChild(buttonContent);

                    document.body.appendChild(previewButton);

                    let pictureWrapper = document.createElement('div');

                    pictureWrapper.setAttribute('id', 'pictureWrapper');

                    document.body.appendChild(pictureWrapper);

                    buttonIsPressed = true;

                }

            });

    }

    function showPicture(event) {

        console.log('pressed');

        fetch('http://localhost:8888/picture/' +

            text.value.trim().split(' ')[0] + '/' +

            text.value.trim().split(' ')[1] + '/' +

            colorSelector.value + '/' +

            sizeSelector.value, {

            method: 'GET',

        })

            .then(response => response.json())

            .then(result => {

                console.log(result);

                pictureWrapper.innerHTML = result.page;

            });

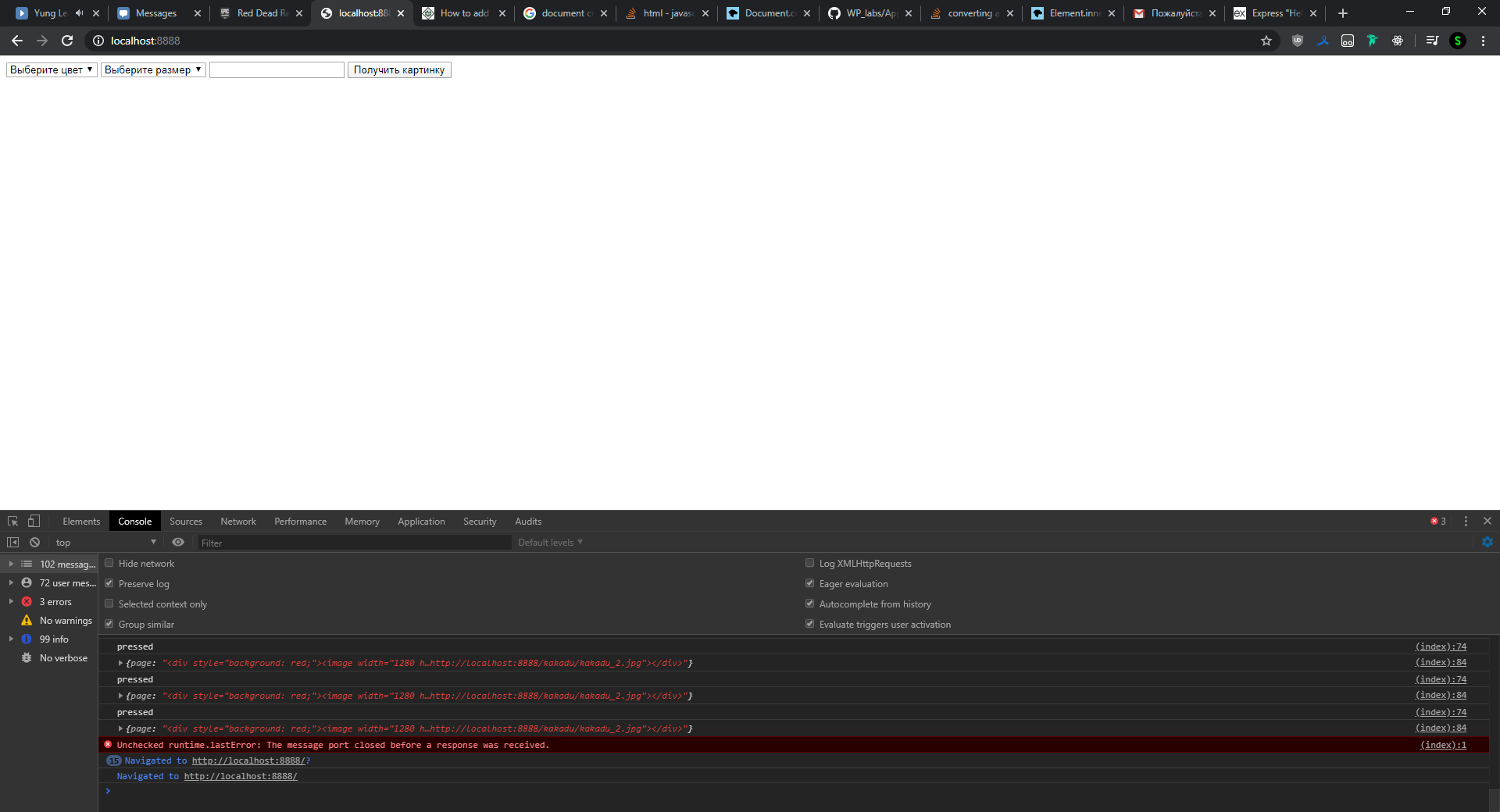
    }

</script>

</html>

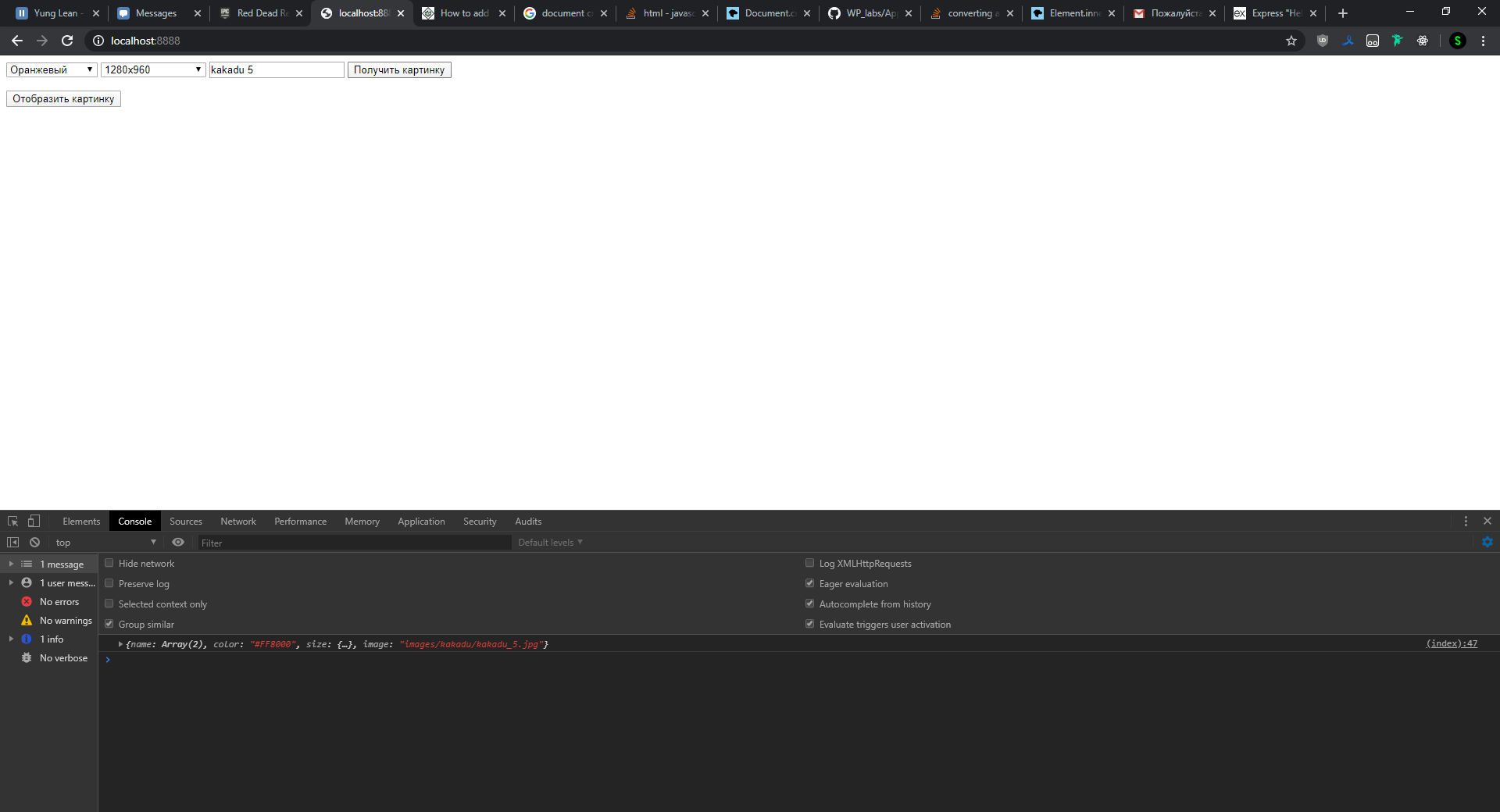
#### Тестирование работы.

Запускаем сервер, открываем браузер по адресу <http://localhost:8888/> и видим следующую форму:

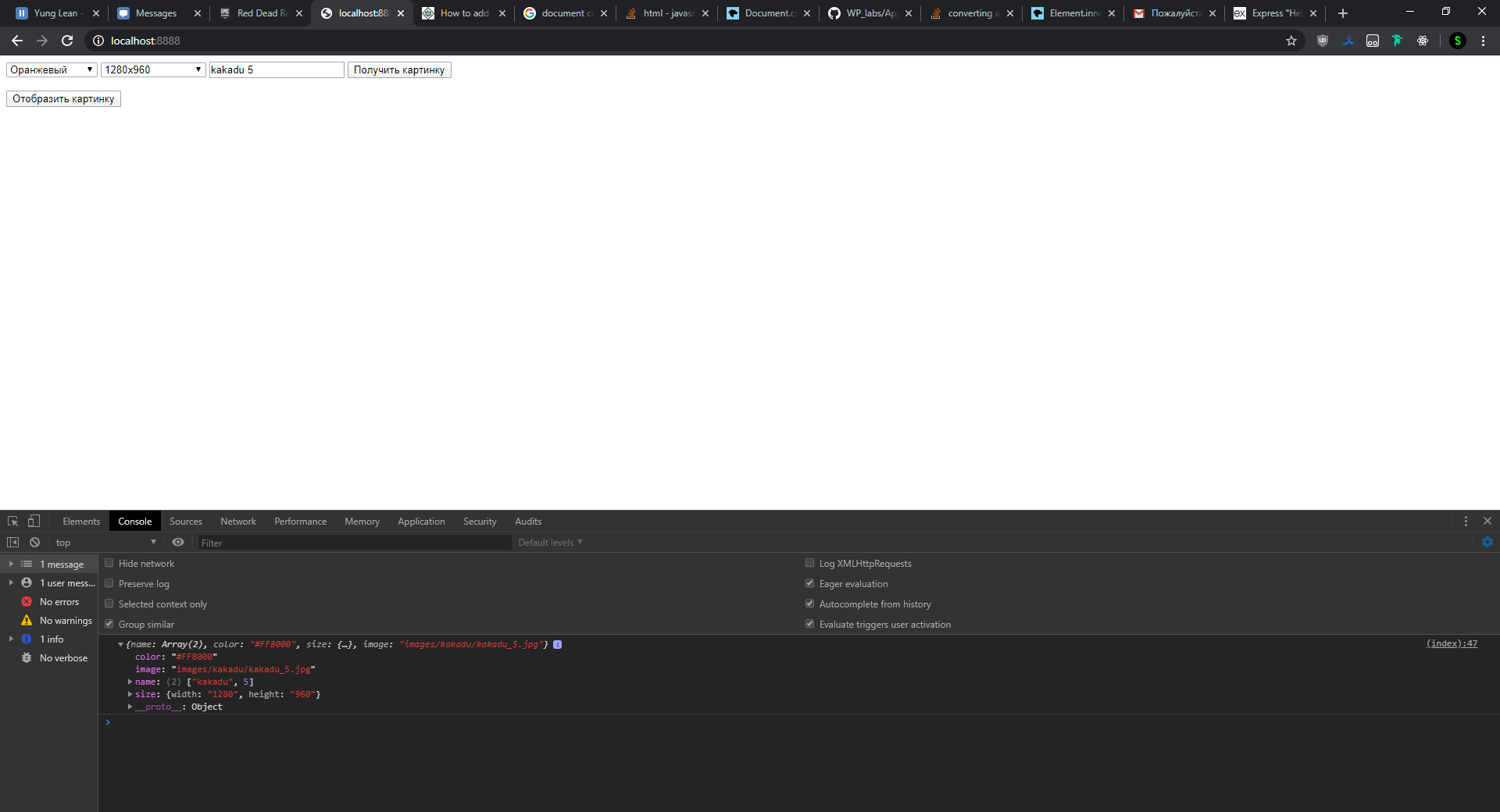


**Рисунок 1. Интерфейс.**

Необходимо заполнить только поле с именем животного, остальные элементы формы заполнятся автоматически после нажатия на кнопку «Получить картинку». Также появится кнопка «Отобразить картинку»

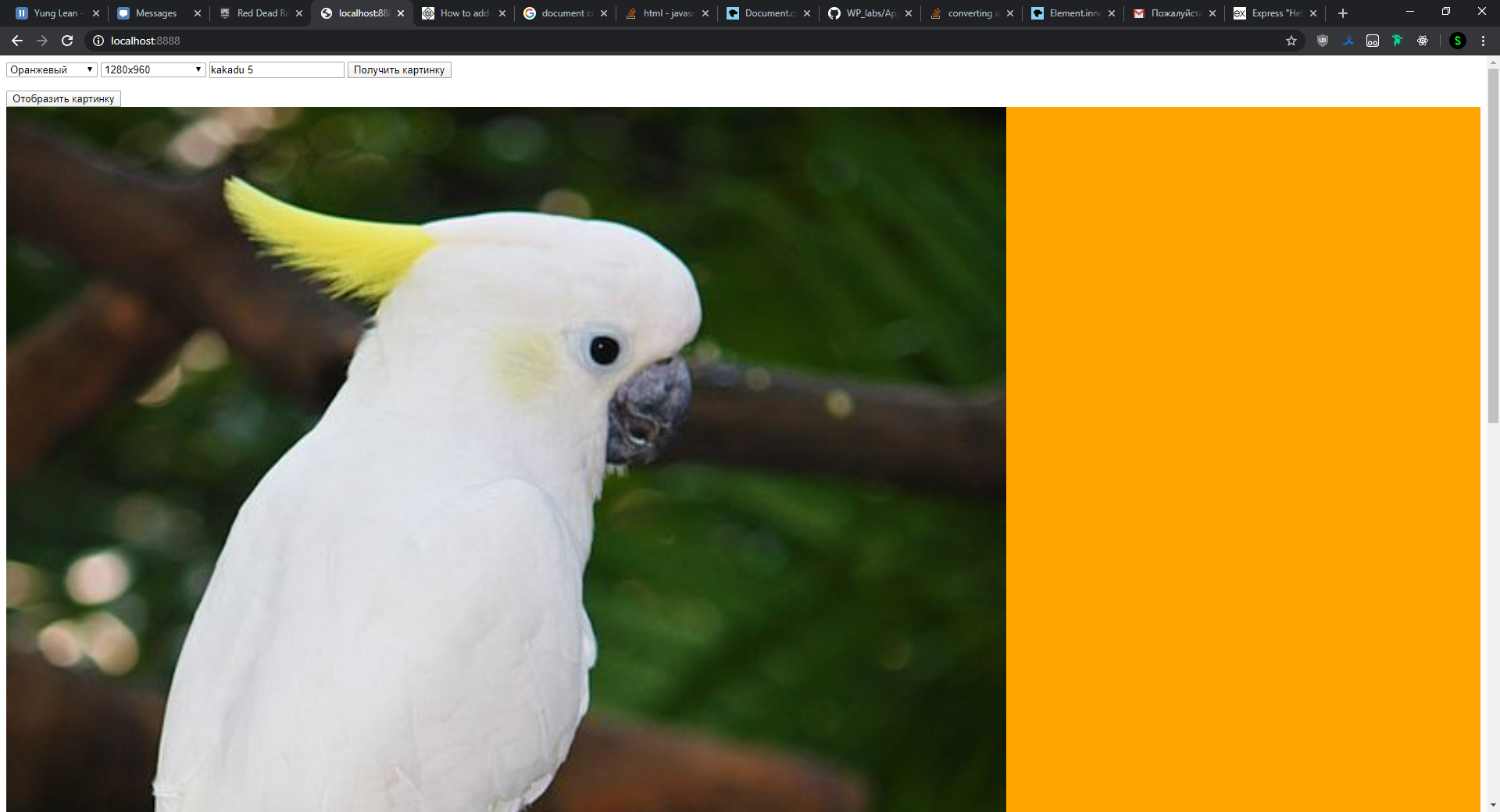


**Рисунок 2. Заполненная форма.**

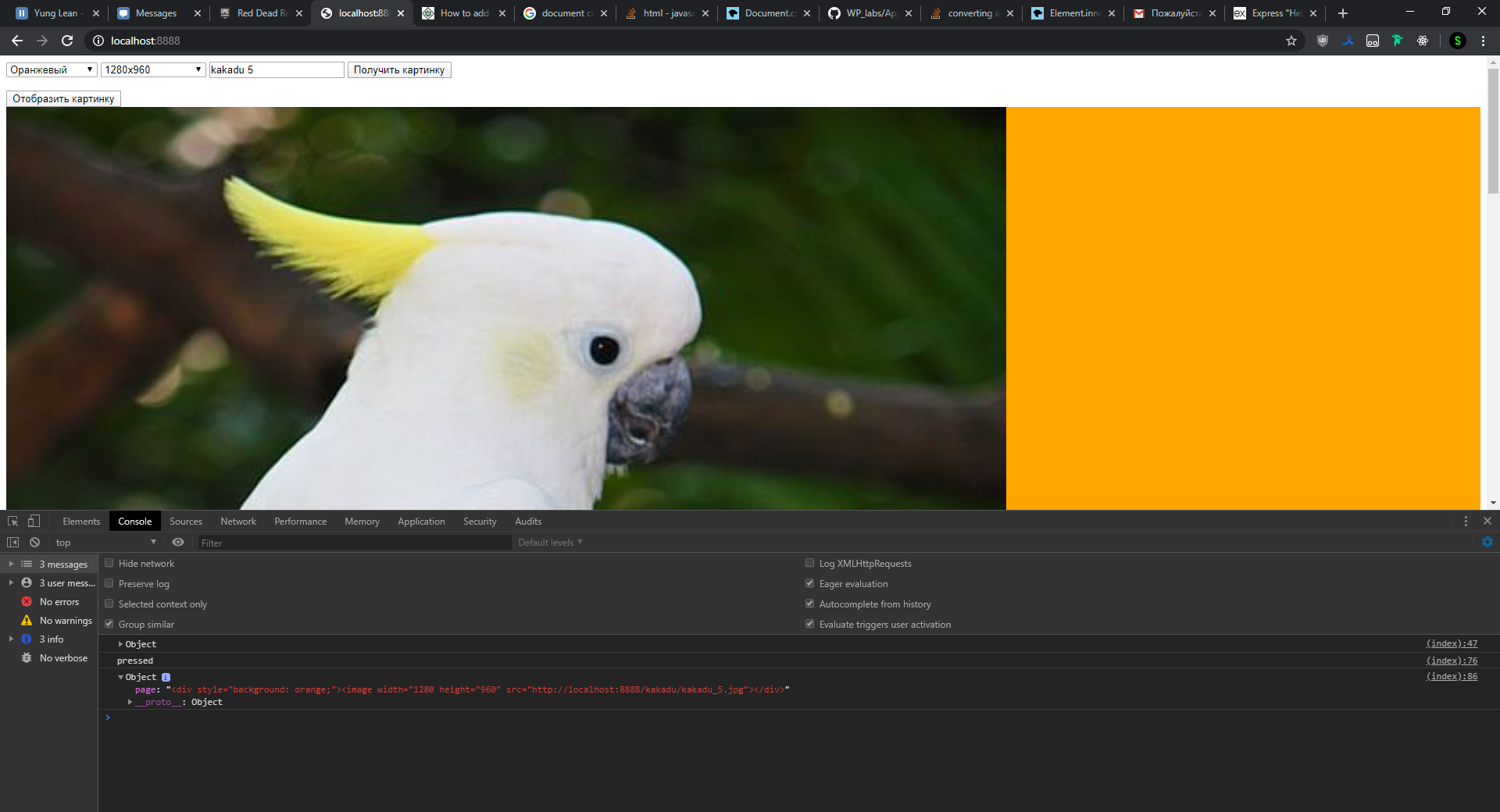


**Рисунок 3. Пришедший ответ, выведенный в консоль.**

После этого нажимаем «Отобразить картинку» и получаем результат ниже формы.



**Рисунок 4. Состояние интерфейса после нажатия.**



**Рисунок 5. Результат из консоли.**

### Выводы

В ходе данной лабораторной работы получили практические навыки реализации REST API.