**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.

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

const express = require('express');

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) => {

    res.send('<html>' +

        '<head>' +

        '<meta http-equiv="Content-Type" content="text/html; ' +

        'charset=UTF-8" />' +

        '</head>' +

        '<body>' +

        '<form action="/getPicture" method="post">' +

        '<select name="colorSelector">' +

        '<option hidden>Выберите цвет</option>' +

        '<option>Красный</option>' +

        '<option>Зеленый</option>' +

        '<option>Оранжевый</option>' +

        '</select>' +

        '<select name="sizeSelector">' +

        '<option hidden>Выберите размер</option>' +

        '<option>400х600</option>' +

        '<option>800х600</option>' +

        '<option>1280х960</option>' +

        '</select>' +

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

        '<input type="submit" value="Submit text" />' +

        '</form>' +

        '</body>' +

        '</html>');

})

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('х'); }

    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' | 'kalmar' | 'kakadu') {

        res.send('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) || 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('<html>' +

        '<head>' +

        '<meta http-equiv="Content-Type" content="text/html; ' +

        'charset=UTF-8" />' +

        '</head>' +

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

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

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

        '" src="/' + imageParams.name + '/' + imageParams.name + '\_' + imageParams.id + '.jpg">' +

        '</body>' +

        '</html>');

})

module.exports = router;

Переписываем index.js до следующего состояния:

const express = require('express')

const app = express()

const port = 8888

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

app.use(express.urlencoded());

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

app.listen(port, () => {

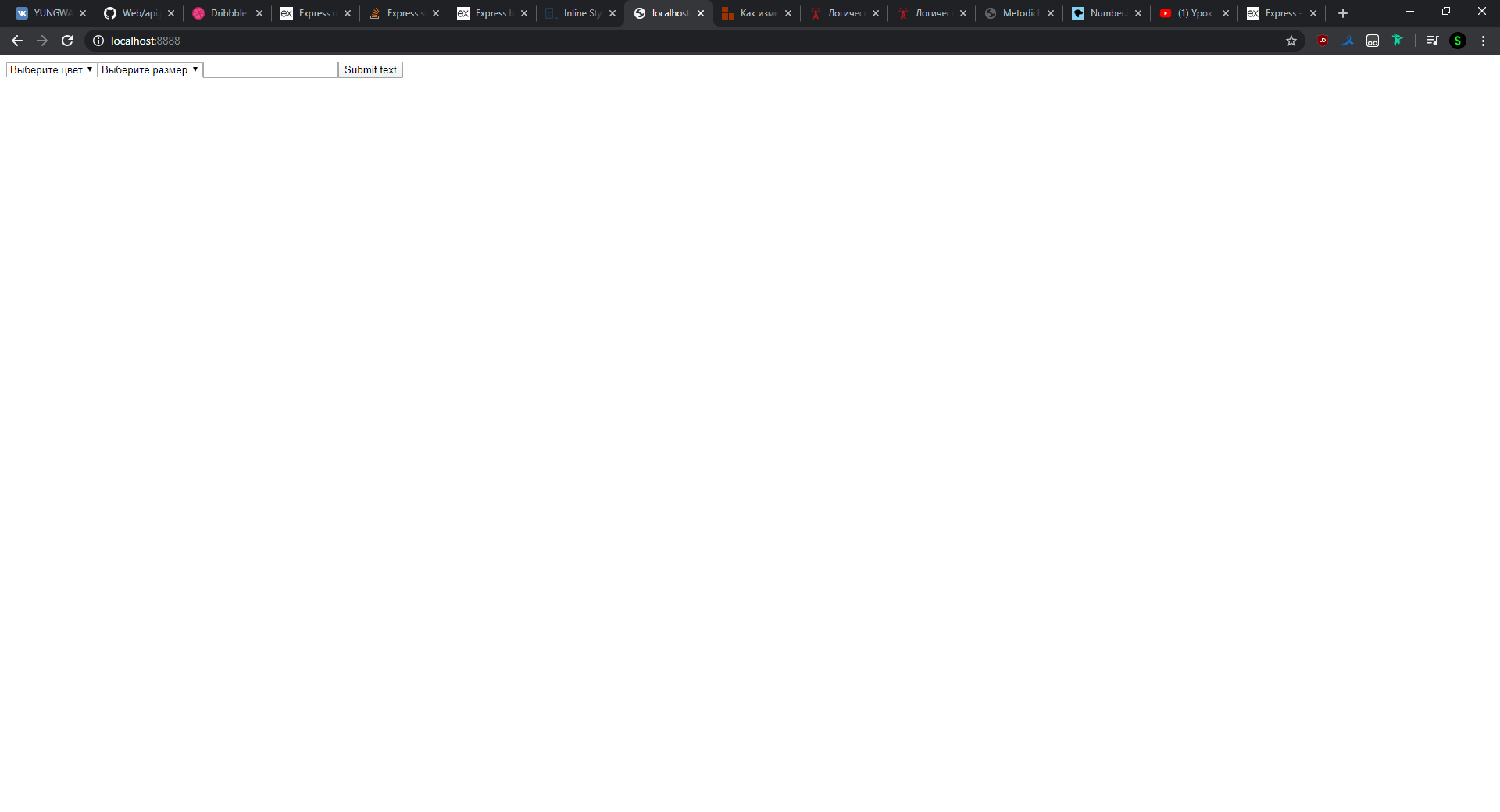
    console.log(`Example app listening at http://localhost:${port}`);

})

Вычищаем все лишние файлы, которые использовались в предыдущей работе.

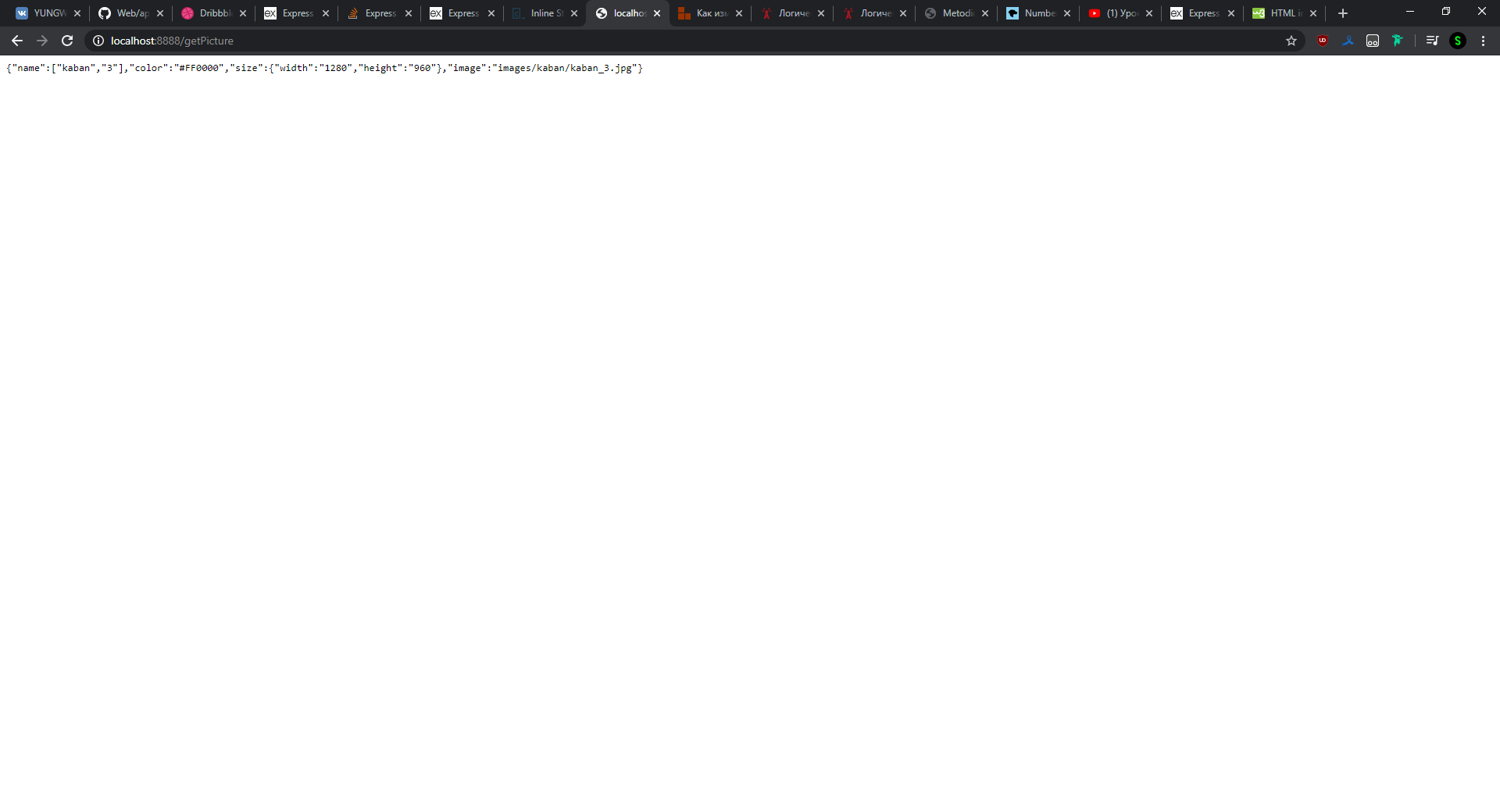
#### Тестирование работы.

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



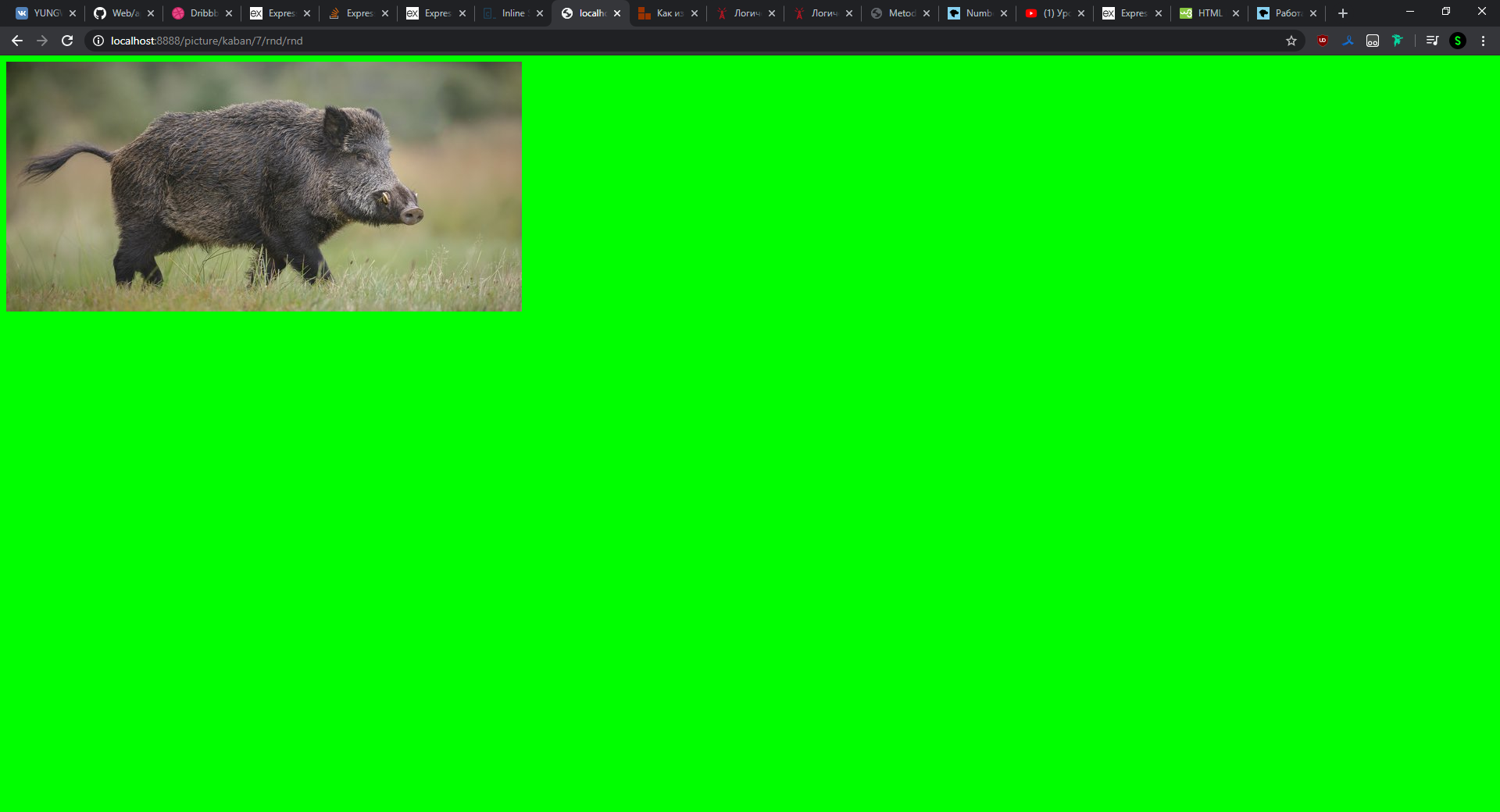
**Рисунок 1. Форма для первого обработчика.**

Необходимо заполнить только поле с именем животного, остальное выберется автоматически в разумных пределах. После нажатия на “submit text” получаем ответ в виде json объекта.

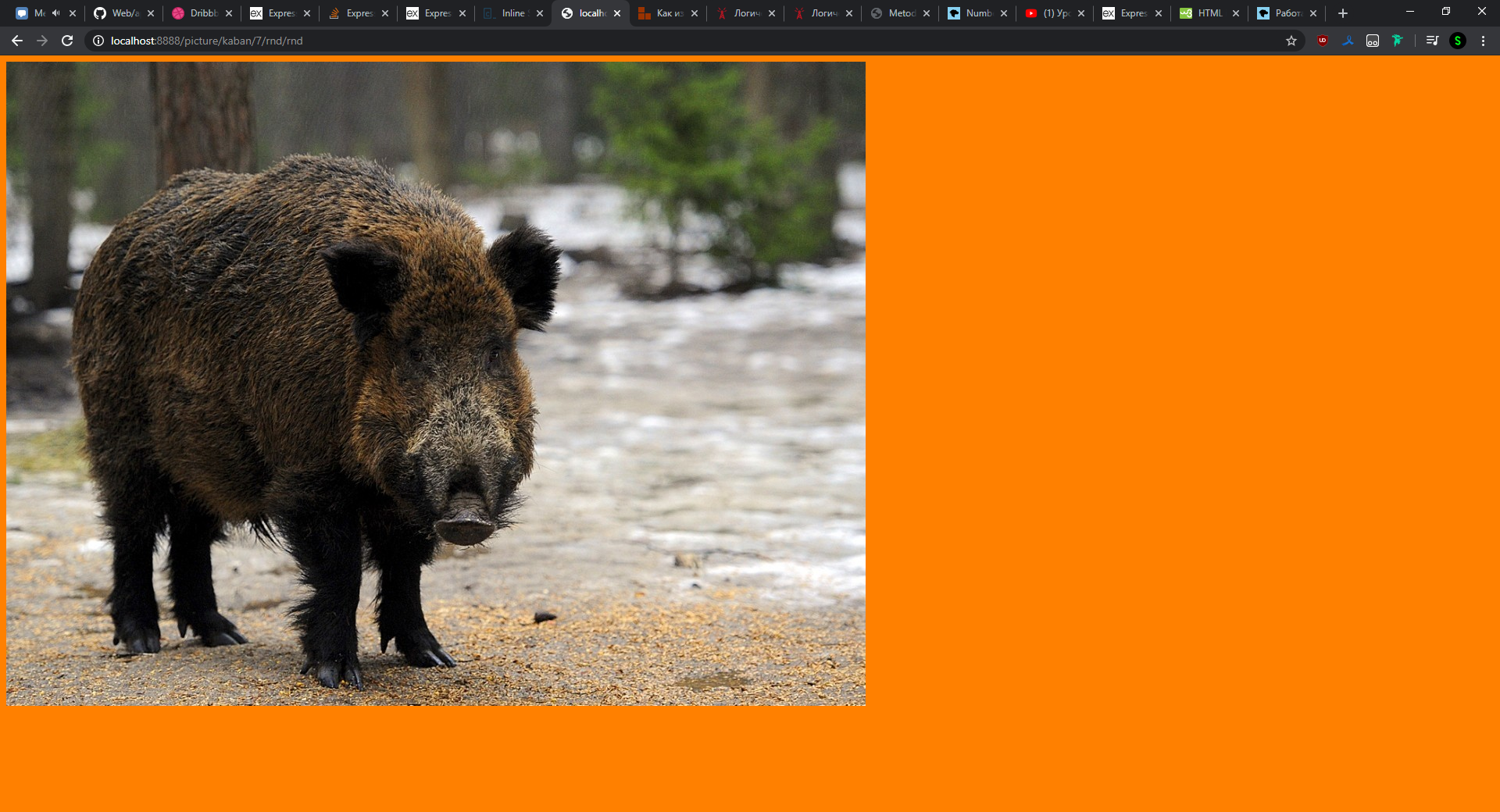


**Рисунок 2. Ответ сервера.**

Также мы можем получить изображение в нужном виде следующим запросом /picture/:name/:id/:color/:size, при этом id, color, size при несоответствии заполнятся случайными значениями.



**Рисунок 3. 1 вариант ответа сервера.**



**Рисунок 4. 2 вариант ответа сервера.**

### Выводы

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