Министерство образования и науки Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЁТ**

по лабораторной работе №5

по курсу "Разработка web-приложений"  
на тему "Реализация ответа обработчиков и POST-запросы "

Выполнили:

Студенты группы 16ВВ2

Воронков Е.В.

Духанов И.В.

Проверили:

Дубравин А.В.  
Карамышева Н.С.

Пенза 2020

**Цель работы**

Изучить основы обработки запросов клиента. Реализовать ответы обработчиков на POST-запросы клиентов. Вариант - 2.

**Задание:**

1. На стороне клиента реализовать проверку: если в текстовом поле указано одно из трех животных в следующем формате: «antilopa» или «antilopa 7»(число от 0 до 9), выполнить формирование и отправку на сервер POST-запроса, в противном случае вывести сообщение о некорректно указанных данных.

Вариант 2: бобер, бизон, бегемот.

1. На стороне сервера реализовать три обработчика, соответствующих каждому из указанных в варианте животных.
2. В обработчике, в случае если в URL запросе отсутствует параметр с указанием номера запрашиваемого изображения животного, выполнить генерацию случайного числа в диапазоне от 0 до 9 и выполнить отправку клиенту соответствующего изображения. Если в URL запросе присутствует параметр с указанием номера запрашиваемого изображения животного – выполнить отправку клиенту соответствующего изображения.

**Выполнение:**

1. На стороне клиента реализовали проверку: если в текстовом поле указано одно из трех животных выполнили формирование и отправку на сервер POST-запроса, в противном случае выводится сообщение о некорректно указанных данных.
2. На стороне сервера реализовали три обработчика, соответствующих каждому из указанных в варианте животных.
3. В обработчике, в случае, если в URL запросе отсутствует параметр с указанием номера запрашиваемого изображения животного, выполнили генерацию случайного числа в диапазоне от 0 до 9 и выполнить отправку клиенту соответствующего изображения. Если в URL запросе присутствует параметр с указанием номера запрашиваемого изображения животного – выполнили отправку клиенту соответствующего изображения.

Листинг файла index.js:

var server = require("./server");

var router = require("./router");

var requestHandlers = require("./requestHandlers");

var handle = {}

handle["/"] = requestHandlers.start;

handle["/start"] = requestHandlers.start;

handle["/favicon.ico"] = requestHandlers.favicon;

handle["/upload"] = requestHandlers.upload;

server.start(router.route, handle);

Листинг файла router.js:

function route(handle, pathname, response, postData) {

console.log("About to route a request for " + pathname);

if (typeof handle[pathname] === 'function') {

handle[pathname](response, postData);

}

else {

console.log("No request handler found for " + pathname);

response.writeHead(404, {"Content-Type": "text/plain"});

response.write("404 Not found");

response.end();

}

}

exports.route = route

Листинг файла requestHandler.js:

const fs = require("fs");

const querystring = require("querystring");

function start(response, postData) {

console.log("Request handler 'start' was called.");

var body = '<html>'+

'<head>'+

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

'charset=UTF-8" />'+

'</head>'+

'<body>'+

'Лабораторная работа 5. Реализация ответа обработчиков и POST-запросы'+

'<br /><br />'+

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

'<input type="text" id="textId" name="UI" type="text" placeholder="" size="40"></input>'+

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

'</form>'+

'</body>'+

'</html>';

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

response.write(body);

response.end();

}

function upload(response, postData) {

console.log("Request handler 'upload' was called.");

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

var req = postData.split('=', 2)[1];

var arr = req.split('+', 2);

var name = arr[0];

var id = arr[1];

if(!id || id.length === 1){

if(!id){

id = Math.floor(Math.random()\*9);

}

if('beaver' === name){

beaver(response, id);

}

else if('bison' === name){

bison(response, id);

}

else if('hippo' === name){

hippo(response, id);

}

else{

formatError(response);

}

}

else{

formatError(response);

}

response.end();

}

function favicon(response) {

console.log("Request handler 'favicon' was called.");

response.writeHead(404);

response.end();

}

function beaver(response, id)

{

response.write(fs.readFileSync('images/beaver/'+id+'.jpg'));

}

function bison(response, id)

{

response.write(fs.readFileSync('images/bison/'+id+'.jpg'));

}

function hippo(response, id)

{

response.write(fs.readFileSync('images/hippo/'+id+'.jpg'));

}

function formatError(response)

{

var body = '<html>'+

'<head>'+

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

'charset=UTF-8" />'+

'</head>'+

'<body>'+

'Ошибка: неверный формат запроса'+

'<br /><br />'+

'Формат запроса: животное или животное номер'+

'<br /><br />'+

'Животные: beaver (бобер), bison (бизон), hippo (бегемот)'+

'<br /><br />'+

'Номер: цифры от 0 до 9'+

'<br /><br />'+

'</body>'+

'</html>';

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

response.write(body);

response.end();

}

exports.start = start;

exports.upload = upload;

exports.favicon = favicon;

Листинг файла server.js:

var http = require("http");

var url = require("url");

function start(route, handle) {

function onRequest(request, response) {

var postData = "";

var pathname = url.parse(request.url).pathname;

console.log("Request for " + pathname + " received.");

request.setEncoding("utf8");

request.addListener("data", function(postDataChunk) {

postData += postDataChunk;

console.log("Received POST data chunk '"+

postDataChunk + "'.");

});

request.addListener("end", function() {

route(handle, pathname, response, postData);

});

}

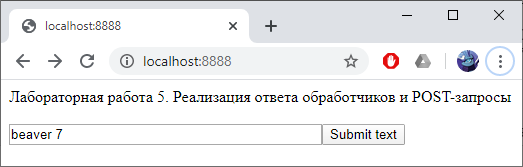
http.createServer(onRequest).listen(8888);

console.log("Server has started.");

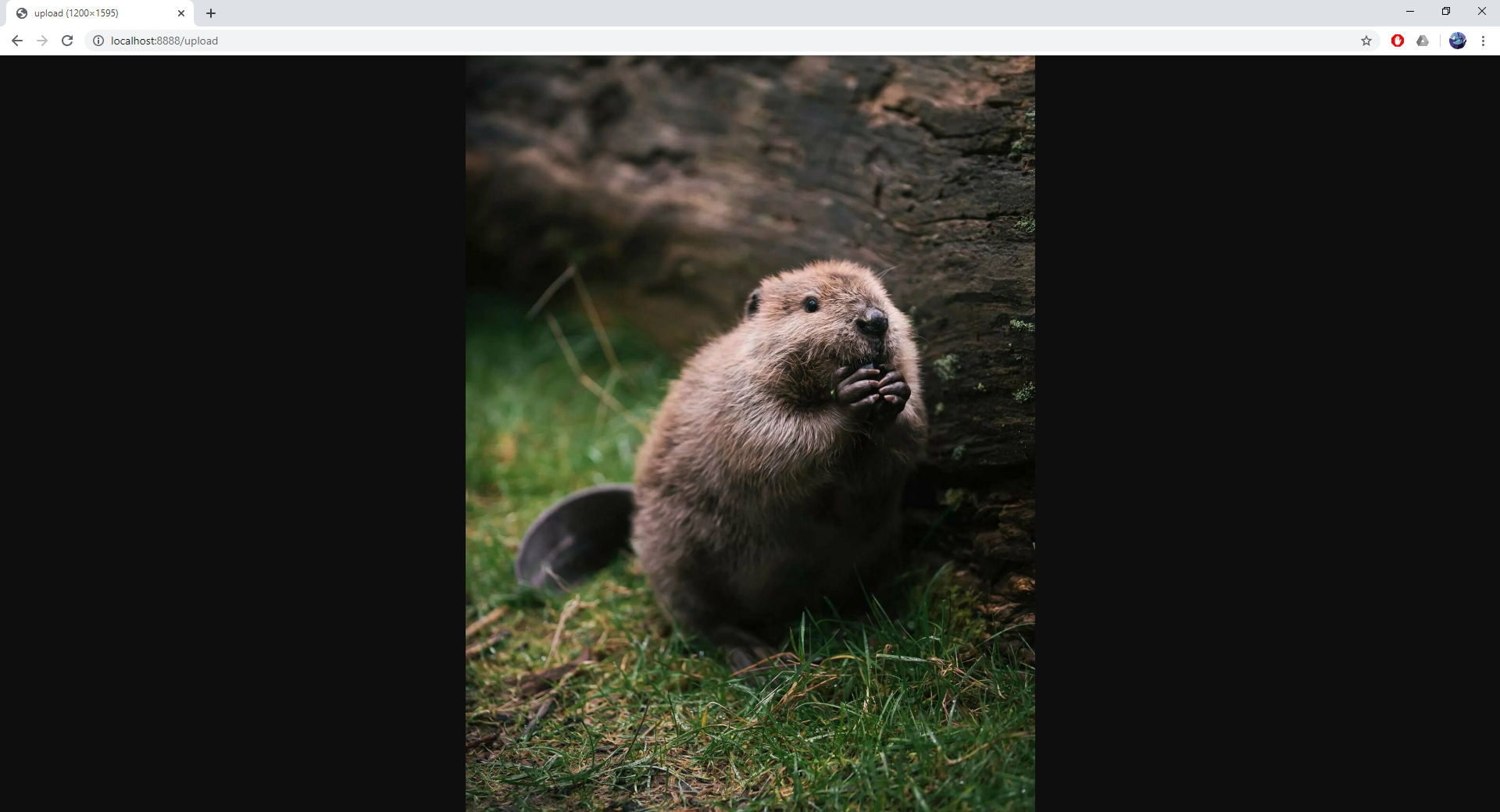
}

exports.start = start;

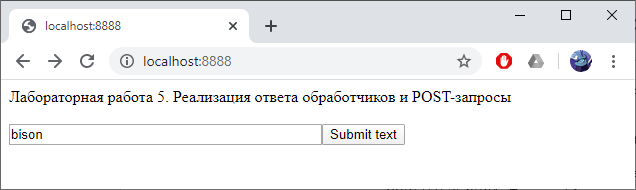
Результат работы программы показан на рисунках:



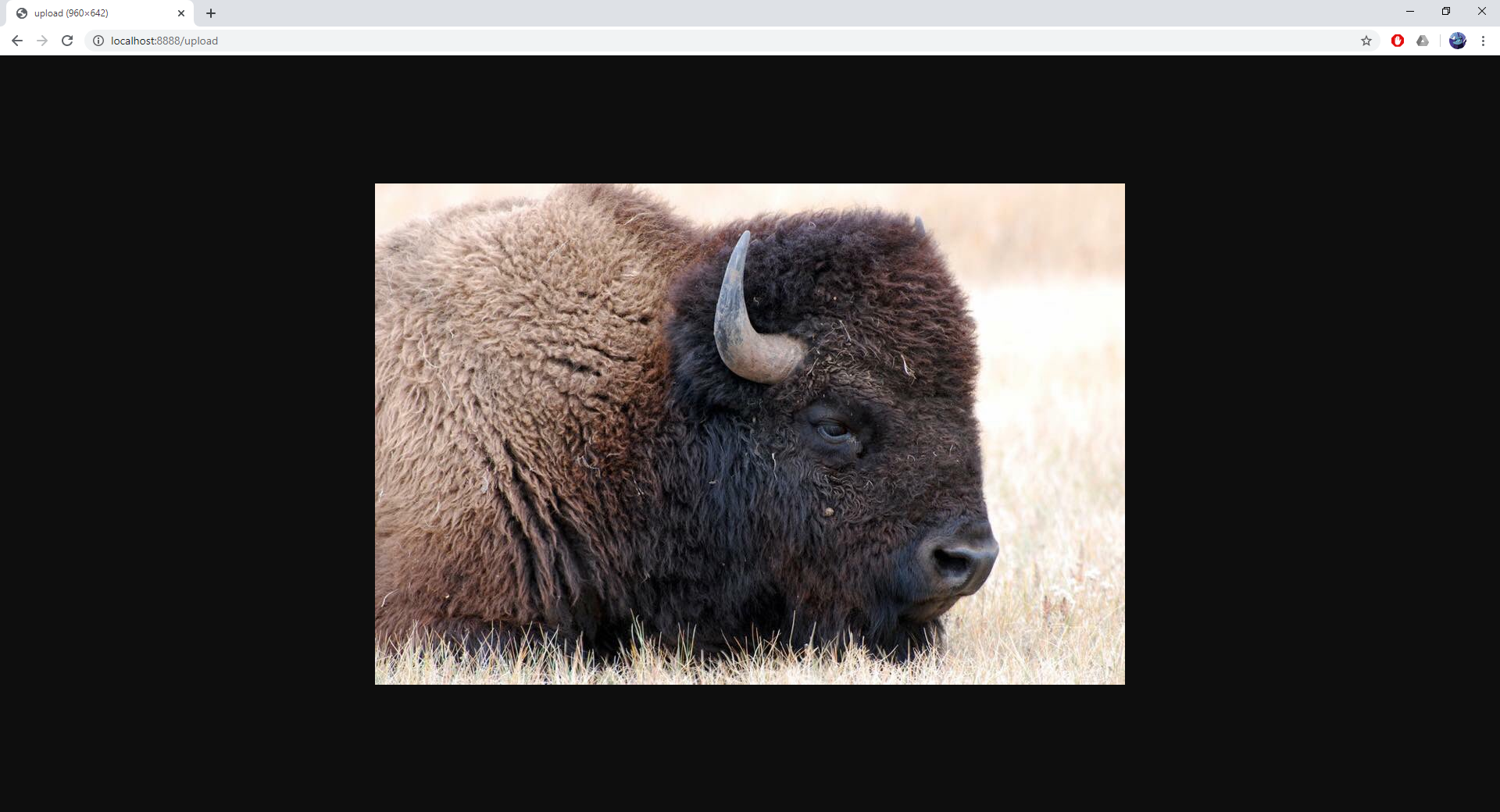
**Рисунок 1 – Запрос**



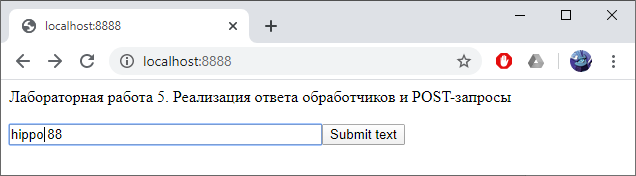
**Рисунок 2 – Ответ**



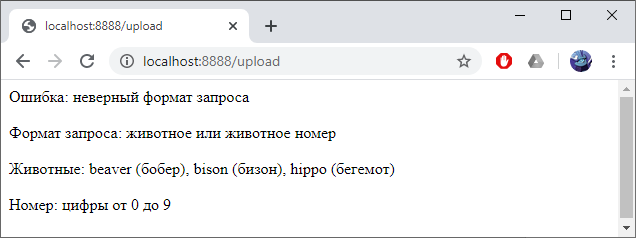
**Рисунок 3 – Запрос**



**Рисунок 4 – Ответ**



**Рисунок 5 – Неверный апрос**



**Рисунок 6 - Ответ**

**Вывод**В ходе выполнения лабораторной работы мы изучили основы обработки запросов клиента. Реализовали ответы обработчиков на POST-запросы клиентов.