**ОТЧЕТ**

по лабораторной работе №5

по курсу «Web-программирование»

на тему «Реализация ответа обработчиков и POST-запросы»

Выполнили:

студенты группы 16ВВ1

Кривошеев С.В.

Щетинин К.А.

Принял:

к.т.н., доцент Дубравин А.В.

к.т.н., доцент Карамышева Н.С.

Пенза 2020

Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

### Название

Реализация ответа обработчиков и POST-запросы.

### Цель работы

Реализовать ответ обработчиков и POST-запросы.

### Лабораторное задание

a. 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.

b. На стороне сервера реализовать три обработчика, соответствующих каждому из указанных в варианте животных.

c. В обработчике, в случае, если в URLзапросе 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 URLзапросе присутствует параметр с указанием номера запрашиваемого изображения животного – выполнить отправку клиенту соответствующего изображения.

|  |  |
| --- | --- |
| Вариант | Задание |
| 4 | Животные: гиена, газель, геккон. |

### Порядок выполнения работы

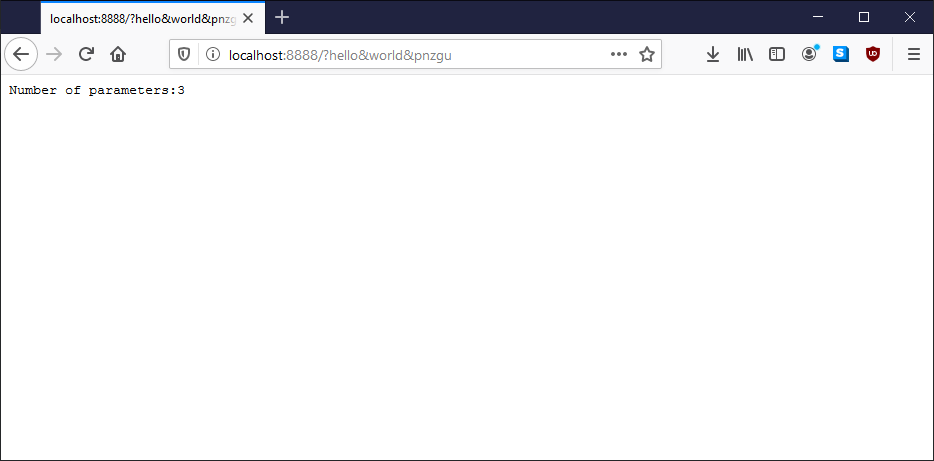


Рисунок 1 – вывод сообщения клиенту о кол-во параметров

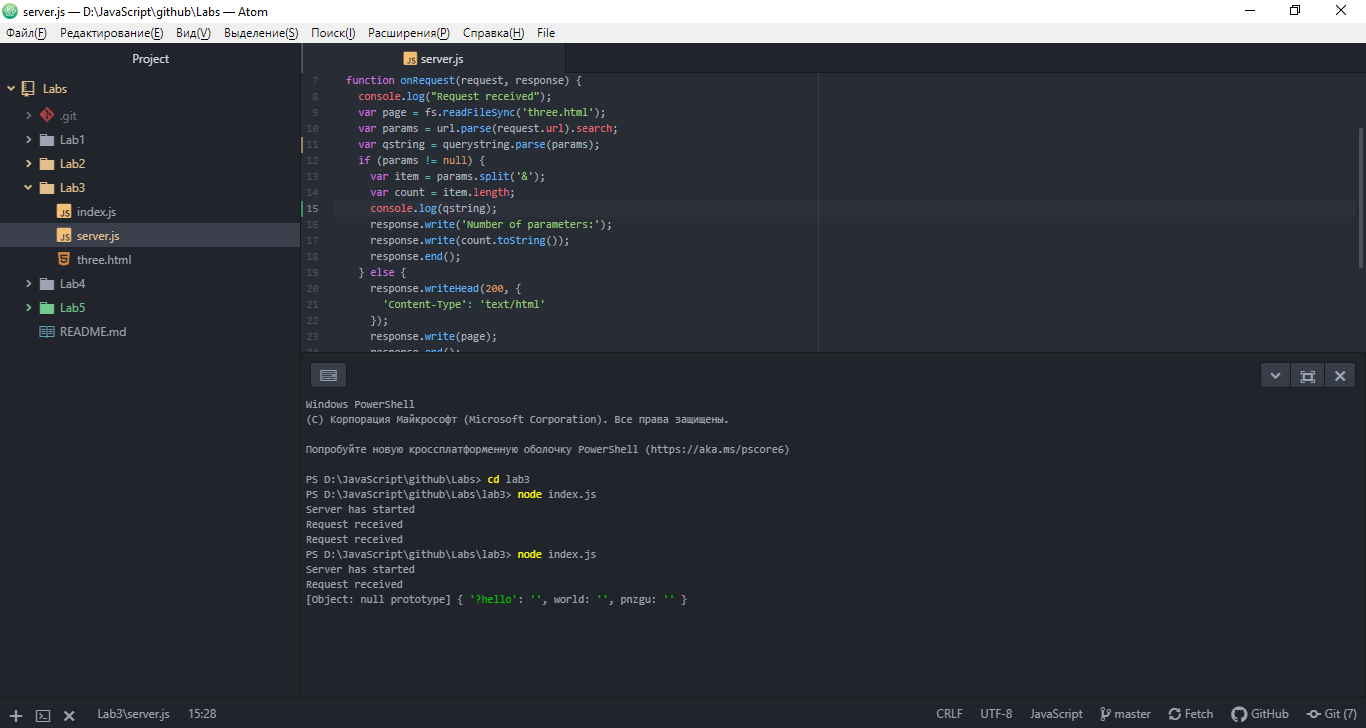


Рисунок 2 - вывод параметров в консоль

### Листинг

#### Three.html

<html>

<head>

<meta charset="utf-8">

<script>

function count() {

if (input.value.length != 0) {

var len = input.value.split("[\\s,.:!?-]+").length;

var answer = "Количество слов в строке: " + len;

} else {

var answer = "NO";

}

alert(answer);

}

</script>

</head>

<body>

<input id="input" type="text" placeholder="Введите текст">

<a onclick="count()">Подсчитать</a>

</<body>

</html>

#### Server.JS

var fs = require("fs");

var http = require("http");

var url = require("url");

var querystring = require("querystring");

function start() {

function onRequest(request, response) {

console.log("Request received");

var page = fs.readFileSync('three.html');

var params = url.parse(request.url).search;

var qstring = querystring.parse(params);

if (params != null) {

var item = params.split('&');

var count = item.length;

console.log(qstring);

response.write('Number of parameters:');

response.write(count.toString());

response.end();

} else {

response.writeHead(200, {

'Content-Type': 'text/html'

});

response.write(page);

response.end();

}

}

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

console.log("Server has started");

}

exports.start = start;

#### Index.JS

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

server.start();

### Выводы

В ходе выполнения лабораторной работы выполнили рефакторинг кода ранее созданного http-сервера,в каталоге проекта создали файл index.js и реализовали обработку входящих http-запросов.