**ОТЧЕТ**

по лабораторной работе №5

по курсу «Разработка Web-приложений»

Выполнили:

студенты группы 16ВВ2

Аккуратнов И.М.

Кожаев В.Д.

Принял:

Дубравин А.В.

Пенза 2020

Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

### Название

### 5. Реализация ответа обработчиков и POST-запросы.

### Цель работы

1. Выполнить редактирование файла server.js

### 2. Выполнить редактирование файла router.js

### 3. Выполнить редактирование файла requestHandlers.js

4. Реализовать обработку POST запроса

5. Выполнить редактирование кодаhttp-сервера в соответствии с заданием.

a. На стороне клиента реализовать проверку: если в текстовом поле указано одно изтрех 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 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.

Вариант Задание

1. Животные: антилопа, акула, ара.

### Листинг

**Файл** *start.html*

<!DOCTYPE *html*>  
<html *lang*="ru">  
 <head>  
 <meta *charset*="UTF-8">  
 <title>Лаборатоная 4</title>  
 <link *rel*="icon" *href*="http://localhost:8888/favicon.ico" *type*="image/x-icon">  
 </head>  
 <body>  
 <form *action*="/upload" *method*="post">  
 <input *name*="text" *type*="text" *required pattern*="(shark|antelope|macaw)( [1-9])?$"  
 *title*="Можно ввести только shark, antelope, macaw с цифрой от 1 до 9"/>  
 <input *type*="submit" *value*="Submit" />  
 </form>  
 </body>  
</html>

**Файл** *server1.3.js*

*const* http = require("http");  
*const* URL = require("url");  
  
*function* start(route, handle) {  
 *function* onRequest(request, response) {  
 *var* postData = "";  
 *var* SiteUrl = URL.*parse*(request.url);  
 *console*.log("Request for " + SiteUrl + " received.");  
 request.setEncoding("utf8");  
 request.addListener("data", (postDataChunk) => {  
 postData += postDataChunk;  
 *console*.log(`Received POST data chunk ' ${postDataChunk}' .`);  
 });  
 request.addListener("end", () => {  
 route(handle, SiteUrl, response, postData);  
 });  
 }  
 http.*createServer*(onRequest).listen(8888);  
 *console*.log("Server has started.");  
}  
exports.start = start;

**Файл** *index.js*

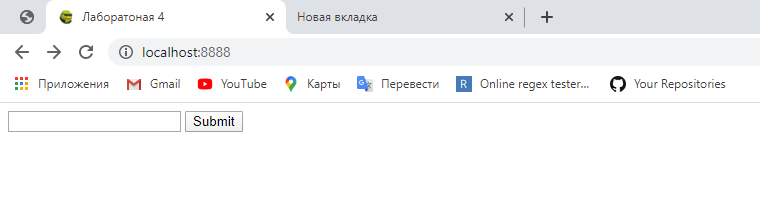
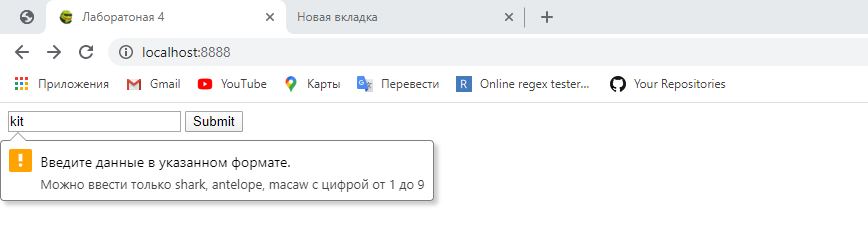
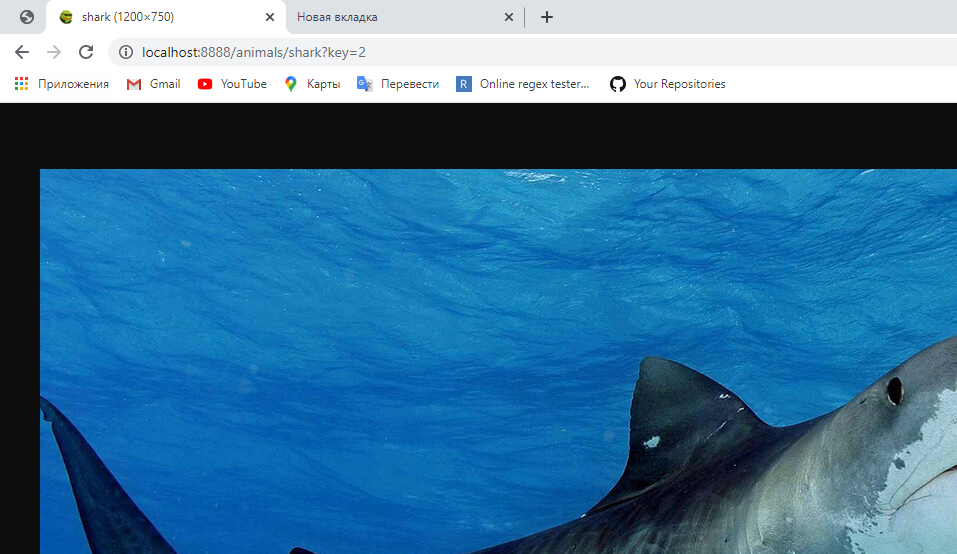
*const* path = require('path');  
*const* server = require(path.resolve(\_\_dirname, 'server1.3.js'));  
*const* router = require(path.resolve(\_\_dirname, 'router.js'));  
*const* requestHandlers = require(path.resolve(\_\_dirname, 'requestHandlers'));  
  
  
*var* handle = {};  
handle["/"] = requestHandlers.start;  
handle["/start"] = requestHandlers.start;  
handle["/animals"] = requestHandlers.start;  
handle["/upload"] = requestHandlers.upload;  
handle["/favicon.ico"] = requestHandlers.favicon;  
handle["/animals/antelope"] = requestHandlers.animalsAntelope;  
handle["/animals/shark"] = requestHandlers.animalsShark;  
handle["/animals/macaw"] = requestHandlers.animalsMacaw;  
  
server.start(router.route, handle);

**Файл** *router.js*

*const* querystring = require("querystring");  
  
*function* route(handle, SiteUrl, response, postData) {  
 *console*.log("About to route a request for " + SiteUrl.pathname);  
 *if* (*typeof* handle[SiteUrl.pathname] === 'function') {  
 handle[SiteUrl.pathname](response, postData, querystring.*parse*(SiteUrl.query).key);  
 } *else* {  
 *console*.log("No request handler found for " + SiteUrl.pathname);  
 response.writeHead(404, {"Content-Type": "text/plain"});  
 response.write("404 Not found");  
 response.end();  
 }  
}  
exports.route = route;

**Файл** *requestHandlers.js*

*const* exec = require("child\_process").*exec*;  
*const* path = require('path');  
*const* fs = require("fs");  
*const* querystring = require("querystring");  
  
*function* start(response, postData, Key) {  
 *console*.log("Request handler 'start' was called.");  
 *const* page = fs.*readFileSync*(path.resolve(\_\_dirname, 'Start.html'));  
 response.writeHead(200, { 'Content-Type': 'text/html' });  
 response.write(page);  
 response.end();  
}  
  
*function* upload(response, postData, Key) {  
 *console*.log("Request handler 'animals' was called.");  
 *let* animalAndKey = querystring.*parse*(postData).text.split(' '); *//animalAndKey[0]- название животного  
 //animalAndKey[1] - Номер изображения животного  
 if* (animalAndKey.length === 1) animalAndKey[1] = (Math.floor(Math.random() \* (9 - 1 + 1)) + 1).toString();  
 *let* newUrl = `http://localhost:8888/animals/${animalAndKey[0]}?key=${animalAndKey[1]}`;  
 response.writeHead(200, {"Content-Type": "text/html"});  
 response.write(`<script> document.location.href = "${newUrl}"; </script>`);  
 response.end();  
}  
  
*function* favicon(response, postData, Key){  
 *console*.log("Request handler 'favicon' was called.");  
 *let* icon = fs.*readFileSync*(path.resolve(path.dirname(\_\_dirname), 'IMG/favicon.ico'));  
 response.writeHead(200, {'Content-Type': 'image/x-icon'});  
 response.write(icon);  
 response.end();  
}  
  
*function* animals(response,postData) {  
 *console*.log("Request handler 'animals' was called.");  
 *var* body = '<html lang="en">'+  
 '<head>'+  
 '<meta http-equiv="Content-Type" content="text/html; '+  
 'charset=UTF-8" />'+  
 '<title> Лабораторная 5</title>' +  
 '</head>'+  
 '<script> document.location.href = "http://localhost:8888/animals"; </script>'+  
 '<body>'+  
 '<form action="/upload" method="post">'+  
 '<input id="text" type="text" />' +  
 '<input type="submit" value="Submit" />' +  
 '</form>'+  
 '</body>'+  
 '</html>';  
 response.writeHead(200, {"Content-Type": "text/html"});  
 response.write(body);  
 response.end();  
}  
  
*function* animalsAntelope(response, postData, Key){  
  
 *console*.log("Request handler 'animalsAntelope' was called.");  
 *let* image = fs.*readFileSync*(path.resolve(path.dirname(\_\_dirname), `IMG/Antelopes/Antelope-${Key}.jpg`));  
 response.writeHead(200, {'Content-Type': 'image/jpeg'});  
 response.write(image);  
 response.end();  
}  
  
*function* animalsShark(response, postData, Key){  
 *console*.log("Request handler 'animalsShark' was called.");  
 *let* image = fs.*readFileSync*(path.resolve(path.dirname(\_\_dirname), `IMG/Sharks/Shark-${Key}.jpg`));  
 response.writeHead(200, {'Content-Type': 'image/jpeg'});  
 response.write(image);  
 response.end();  
}  
  
*function* animalsMacaw(response, postData, Key){  
 *console*.log("Request handler 'animalsMacaw' was called.");  
 *let* image = fs.*readFileSync*(path.resolve(path.dirname(\_\_dirname), `IMG/Macaws/Macaw-${Key}.jpg`));  
 response.writeHead(200, {'Content-Type': 'image/jpeg'});  
 response.write(image);  
 response.end();  
}  
  
exports.start = start;  
exports.upload = upload;  
exports.favicon = favicon;  
exports.animals = animals;  
exports.animalsAntelope = animalsAntelope;  
exports.animalsShark = animalsShark;  
exports.animalsMacaw = animalsMacaw;

1. Окно поиска
2. При ошибке
3. Как показывает картинки

### Выводы

### В ходе выполнения лабораторной работы реализовали интерфейс для реализации обработки URL запросов, добавили поиск картинок по запросу.