Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

Лабораторная работа № 7

**«Frontend на ReactJS »**

## Отчёт по дисциплине

**«Web-программирование»**

Пенза 2020

**Выполнили с-ты гр. 16ВВ2:**

Анастасин В.Д.  
Прокуров Д.А.

**Приняли:**

к.т.н. Дубравин А.В.  
Кармышева Н.С.

**Цель работы.**

Реализовать Frontend на ReactJS.

**Задание.**

Необходимо для лабораторной работы №5 реализовать клиентское приложение с использованием технологи ReactJS.

|  |  |
| --- | --- |
| **Вариант** | **Животные** |
| 4 | Гиена, газель, геккон |

**Листинг сервера.**

**index.js**

const express = require("express");

const mongoose = require("mongoose");

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

const app = express();

mongoose.connect("mongodb://localhost/animal-db",

{

    useNewUrlParser: true,

    useUnifiedTopology: true

}).then(() => {

    console.log("Connection to BD Success!")

});

app.use('/pic', express.static('pic'));

app.use(bodyParser.json());

app.use("/api",require("./api"));

app.listen(8888, ()=>{

    console.log("Server processing...");

});

**api.js**

const express = require("express");

var fs = require("fs");

const router = express.Router();

const Animal = require("./animal");

// Тело запроса

let jsonbody = null;

// router.get("", (req, res) => {

//     var page = fs.readFileSync("startPage.html");

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

//     res.write(page);

//     res.end();

// })

router.get("/animal", (req, res)=>{

    console.log(jsonbody);

    res.send(jsonbody);

    // res.send({method: "GET"});

});

router.post("/find", (req, res)=>{

    console.log("POST is reached");

    console.log(req.body);

    var species = req.body.species;

    var value = req.body.value;

    if (value === '')

    {

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

    }

    Animal.findOne({species: species, value: value})

        .then(animal => {

            if (animal != null)

            {

                res.send({species: animal.species, value: animal.value, width: animal.width, height: animal.height, ID: animal.ID});

            }

            else

            {

                imgerror = {error: "Image not found"};

                res.send(imgerror);

            }

        })

});

router.post("/animal", (req, res)=>{

    console.log("POST is reached");

    console.log(req.body);

    var species = req.body.species;

    var width = req.body.width

    var height = req.body.height;

    var ID = req.body.ID;

    var number = req.body.number

    let finderror = null;

    if (width == "") {

        width = 1;

    }

    if (height == ""){

        height = 1;

    }

    if (ID == ""){

        finderror = {error: "ID Error", species: species, number: number };

        res.send(finderror);

    }

    Animal.findOne({species: species, number: number})

    .then(animal => {

        if (animal != null)

        {

            Animal.findOne({width: width, height: height})

            .then(animal => {

                if(animal != null)

                {

                    Animal.findOne({ID: ID})

                    .then(animal => {

                        if(animal != null)

                        {

                            jsonbody = animal;

                            res.redirect("./animal");

                        }

                        else

                        {

                            finderror = {error: "ID Error", species: species, number: number };

                            res.send(finderror);

                        }

                    });

                }

                else

                {

                    finderror = {error: "Size Error", species: species, number: number };

                    res.send(finderror);

                }

            });

        }

        else

        {

            finderror = {error: "Species Error", species: species, number: number };

            res.send(finderror);

        }

    });

});

// Технический метод для заполнения БД

// router.post("/animals", (req, res)=>{

//     animal.create(req.body)

//     .then(animal =>{

//         res.send(animal);

//     })

// });

module.exports = router;

**animal.js**

const mongoose = require("mongoose");

const Schema = mongoose.Schema;

const animalSchema = new Schema({

    species: String,

    width: Number,

    height: Number,

    ID: String,

    image: String,

    value: Number

});

const animal = mongoose.model("animal",animalSchema);

module.exports = animal;

**Листинг клиента**

**App.js**

import React, { useState } from 'react';

function App() {

  const [species, setSpecies] = useState('')

  const [value, setValue] = useState('')

  const [width, setWidth] = useState('')

  const [height, setHeight] = useState('')

  const [ID, setID] = useState('')

  const [image, setImage] = useState('')

  const [img, setImg] = useState('')

  /// Функция получения всех параметров

  function startSearch(event) {

    event.preventDefault()

    if (species.trim()) {

      var animalspecies = null;

      var animalvalue = null;

      animalspecies = species;

      animalvalue = value;

      let animanlsJSON = {

        species: animalspecies,

        value: animalvalue,

      };

      console.log(animanlsJSON);

      fetch('/api/find', {

        method: 'POST',

        headers: {

          'Content-Type': 'application/json'

        },

        body: JSON.stringify(animanlsJSON)

        })

        .then(response => {

          if(!response.ok) {

            console.log(response);

            throw new Error(response.statusText)

          }

          return response;

        })

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

            .then(result => {

              console.log(result)

              if (result.error !== "Image not found")

              {

                setValue(result.value)

                setWidth(result.width)

                setHeight(result.height)

                setID(result.ID)

                setImage('')

              }

              else

              setImage('Nothing in DB')

            });

    }

  }

  /// Функция получения картинки

  function sendReq(event) {

    event.preventDefault()

    if (species.trim()) {

      var animalspecies = null;

      var animalvalue = null;

      var imgwidth = null;

      var imgheight = null;

      var imgID = null;

      animalspecies = species;

      animalvalue = value;

      imgwidth = width;

      imgheight = height;

      imgID = ID;

      let animalsJSON = {

        species: animalspecies,

        value: animalvalue,

        width: imgwidth,

        height: imgheight,

        ID: imgID

      };

      console.log(animalsJSON)

      fetch('/api/animal', {

        method: 'POST',

        headers: {

          'Content-Type': 'application/json'

        },

        body: JSON.stringify(animalsJSON)

        })

        .then(response => {

          if(!response.ok) {

            throw new Error(response.statusText)

          }

          return response;

        })

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

            .then(result => {

              console.log(result)

              if (result.image != null)

              {

                setImage(result.image)

                setImg(result.image)

              }

              else

              setImage('Image not found')

            });

    }

  }

  return (

    <div className="wrapper">

      <label>

          <a>Вид: </a>

          <a style={{marginLeft: 150}}>Значение: </a>

          <br></br>

          <input

            value={species}

            onChange={event => setSpecies(event.target.value)}

          />

          <input

            style={{marginLeft: 15}}

            value={value}

            onChange={event => setValue(event.target.value)}

          /><p></p>

          <a>Ширина: </a>

          <a style={{marginLeft: 117}}>Высота: </a>

          <br></br>

          <input

            value={width}

            onChange={event => setWidth(event.target.value)}

          ></input>

          <input

            style={{marginLeft: 15}}

            value={height}

            onChange={event => setHeight(event.target.value)}

          ></input><p></p>

          <a>ID:</a>

          <br></br>

          <input

            value={ID}

            onChange={event => setID(event.target.value)}

          ></input><p></p>

          <a>Изображение:</a>

          <br></br>

          <input

            value={image}

            onChange={event => setImage(event.target.value)}

          ></input><p></p>

          <input

            type="submit"

            value="Поиск"

            onClick={sendReq}

          />

          <p></p>

          <input

            type="submit"

            value="Найти недостающие параметры"

            onClick={startSearch}

          /><p></p>

        <img

          src={img}

          onChange={event => setImg(event.target.value)}

        />

      </label>

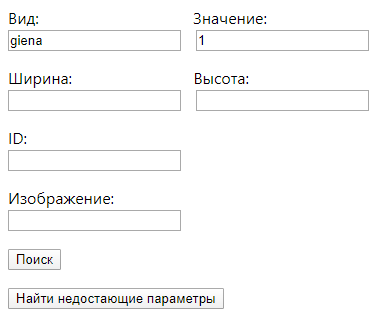
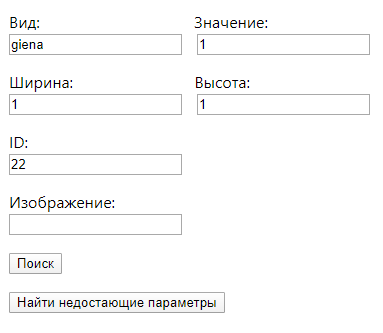
    </div>

  );

}

export default App;

**Результат работы.**

 Рисунок 1 – Получение параметров с сервера

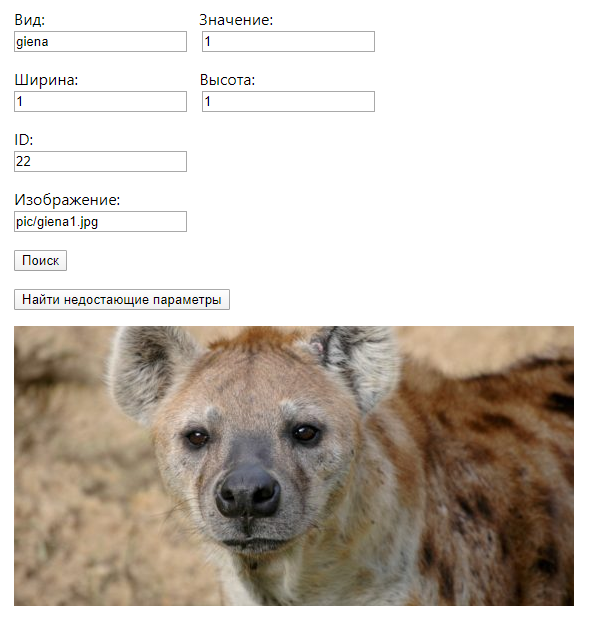


Рисунок 2 – Поиск изображения

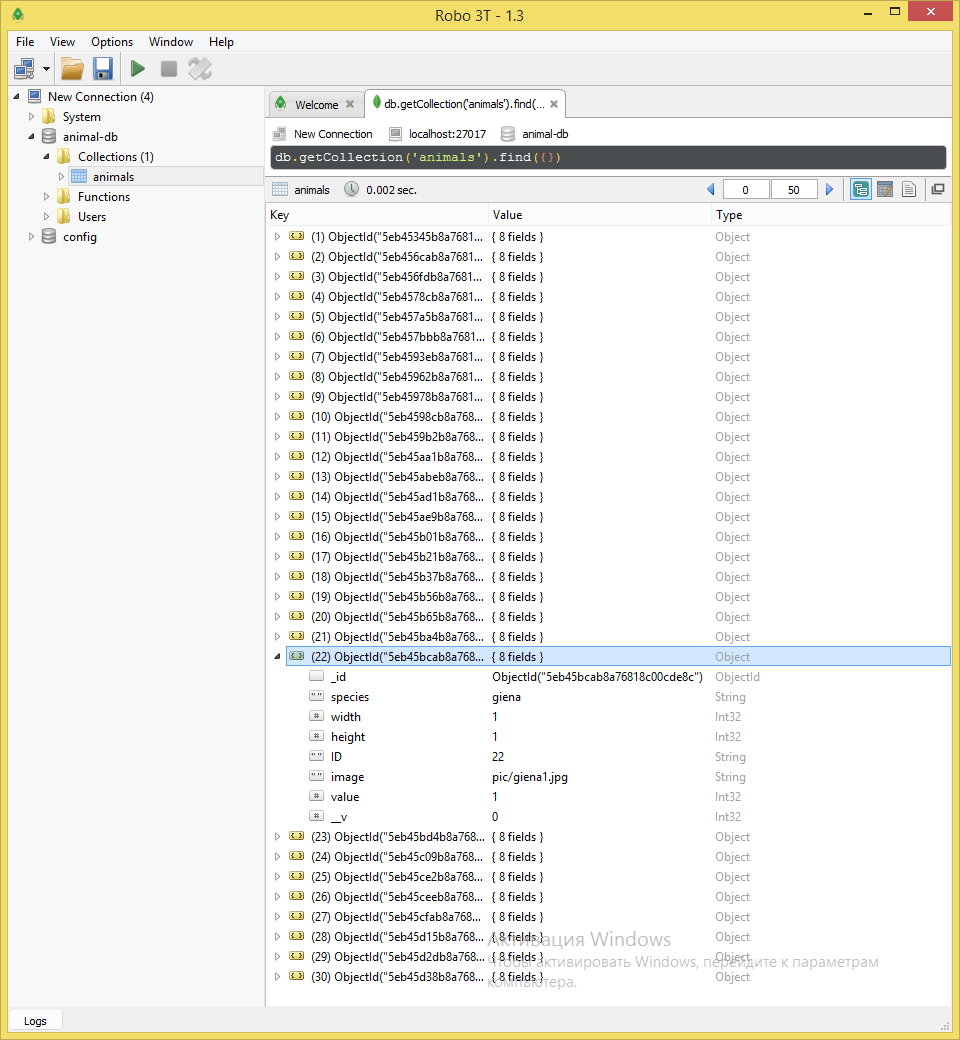


Рисунок 3 – Запись в БД

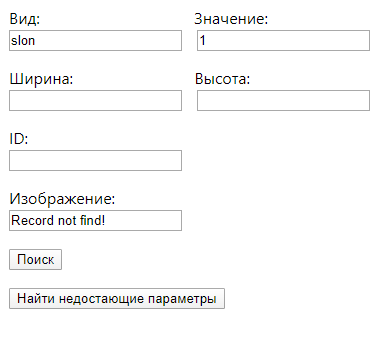


Рисунок 4 – Результат ввода неверных данных

**Вывод.**

В ходе выполнения лабораторной работы мы научились реализовывать Frontend на ReactJS.