МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Факультет информационных систем и технологий

Кафедра «Информационные системы»

Дисциплина «Интернет программирование»

**Отчёт по лабораторной работе №1**

**Создание серверной части веб-приложения**

Выполнил студент

гр. ПИбд-21:

Сергеев Н.И.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Научный руководитель:

Филиппов А.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ульяновск

2022

**Задание**

1. Создать spring boot приложение на сайте start.spring.io со следующими параметрами:
   * система сборки – gradle,
   * версия JDK – 17,
   * зависимости – Web.
2. Добавить REST-контроллер, в котором содержится не менее 4 методом для выполнения каких-либо действий над данными, например, реализация операций нахождения результатов суммы, разности, умножения и деления. Данные для выполнения операций должны быть указаны в виде параметров (RequestParam) HTTP-запроса. Результат выполнения операции должен отображаться в браузере.

**Сервер**

Файл LabworkApplication – метод main + контроллер, обрабатывающий http-запросы.

Методы:

* Sum – сложение числа;
* Ras – разность чисел;
* Pros – произведение чисел;
* Del – деление чисел;

package ip.labwork;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

@SpringBootApplication

@RestController

public class LabworkApplication {

public static void main(String[] args) {

SpringApplication.run(LabworkApplication.class, args);

}

@GetMapping("/hello")

public String hello(@RequestParam(value = "name", defaultValue = "World") String name) {

return String.format("Hello %s!", name);

}

@GetMapping("/sum")

public String Sum(@RequestParam(defaultValue = "0") double first,

@RequestParam(defaultValue = "0") double second) {

return Double.toString(first + second);

}

@GetMapping("/minus")

public String Ras(@RequestParam(defaultValue = "0") double first,

@RequestParam(defaultValue = "0") double second) {

return Double.toString(first - second);

}

@GetMapping("/multi")

public String Pros(@RequestParam(defaultValue = "1") double first,

@RequestParam(defaultValue = "1") double second) {

return Double.toString(first \* second);

}

@GetMapping("/div")

public String Del(@RequestParam(defaultValue = "1") double first,

@RequestParam(defaultValue = "1") double second) {

if (second == 0) {

return null;

}

return Double.toString(first / second);

}

}

Файл WebConfiguration – файл-конфигурация для приложения.

package ip.labwork;

import org.springframework.context.annotation.Configuration;

import org.springframework.web.servlet.config.annotation.CorsRegistry;

import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

@Configuration

class WebConfiguration implements WebMvcConfigurer {

@Override

public void addCorsMappings(CorsRegistry registry){

registry.addMapping("/\*\*").allowedMethods("\*");

}

}

**Клиент**

Файл index.html

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<script src="node\_modules/bootstrap/dist/js/bootstrap.min.js"></script>

<link href="node\_modules/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet"/>

<link href="node\_modules/@fortawesome/fontawesome-free/css/all.min.css" rel="stylesheet"/>

<title>Document</title>

</head>

<body>

<form id="frm-items" class="row g-3">

<div class="d-flex flex-column w-25 my-5 mx-3">

Введите первое число

<input id="first"></input>

Введите второе число

<input id="second"></input>

Выберите операцию

<div>

<button type="button" class="btn btn-primary" id="buttonPlus">+</button>

<button type="button" class="btn btn-primary" id="buttonMinus">-</button>

<button type="button" class="btn btn-primary" id="buttonMulti">\*</button>

<button type="button" class="btn btn-primary" id="buttonDiv">/</button>

</div>

Результат

<input id="res"></input>

</div>

</form>

<script src="/script.js"></script>

</body>

</html>

Файл script.js

let calculateButton = document.getElementById("calculate");

let numberOneInput = document.getElementById("first");

let numberTwoInput = document.getElementById("second");

let resultInput = document.getElementById("res");

buttonPlus.onclick = function(event) {

event.preventDefault();

let num\_1 = numberOneInput.value;

let num\_2 = numberTwoInput.value;

fetch(`http://localhost:8080/sum?first=${num\_1}&second=${num\_2}`)

.then(response => response.text())

.then(res => {

const ru = new Intl.NumberFormat("ru").format(res);

resultInput.value = ru;

});

}

buttonMinus.onclick = function(event) {

event.preventDefault();

let num\_1 = numberOneInput.value;

let num\_2 = numberTwoInput.value;

fetch(`http://localhost:8080/minus?first=${num\_1}&second=${num\_2}`)

.then(response => response.text())

.then(res => {

const ru = new Intl.NumberFormat("ru").format(res);

resultInput.value = ru;

});

}

buttonMulti.onclick = function(event) {

event.preventDefault();

let num\_1 = numberOneInput.value;

let num\_2 = numberTwoInput.value;

fetch(`http://localhost:8080/multi?first=${num\_1}&second=${num\_2}`)

.then(response => response.text())

.then(res => {

const ru = new Intl.NumberFormat("ru").format(res);

resultInput.value = ru;

});

}

buttonDiv.onclick = function(event) {

event.preventDefault();

let num\_1 = numberOneInput.value;

let num\_2 = numberTwoInput.value;

fetch(`http://localhost:8080/div?first=${num\_1}&second=${num\_2}`)

.then(response => response.text())

.then(res =>

{

const ru = new Intl.NumberFormat("ru").format(res);

resultInput.value = ru;

});

}