Федеральное государственное бюджетное образовательное учреждение   
высшего образования

|  |  |
| --- | --- |
| Gerb-BMSTU_01 | «Московский государственный технический университет  им. Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана) |

ФАКУЛЬТЕТ – Информатика и управления

КАФЕДРА – Информационные системы и телекоммуникации

РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе на тему

«Разработка механизма передачи графа GEF4 с сервера в веб-браузер. Клиентская часть»

Студент группы ИУ3-72 (подпись) 06.02.2017 Д.О.Дементьев

Руководитель курсовой работы (подпись) 06.02.2017 А. М. Иванов

Москва, 2017

Код разработанной программы Client.java:

package com.mydomain.teavm\_firstapp;

import org.json.simple.JSONArray;

import org.json.simple.JSONObject;

import org.json.simple.JSONValue;

import org.teavm.jso.ajax.XMLHttpRequest;

import org.teavm.jso.canvas.CanvasRenderingContext2D;

import org.teavm.jso.dom.html.HTMLButtonElement;

import org.teavm.jso.dom.html.HTMLCanvasElement;

import org.teavm.jso.dom.html.HTMLDocument;

import org.teavm.jso.dom.html.HTMLElement;

public class Client {

private static HTMLDocument document = HTMLDocument.current();

private static HTMLButtonElement helloButton = document.getElementById("hello-button").cast();

private static HTMLButtonElement rectButton = document.getElementById("rect-button").cast();

private static HTMLElement responsePanel = document.getElementById("response-panel");

private static HTMLElement responsePanel2 = document.getElementById("response-panel2");

private static HTMLElement thinkingPanel = document.getElementById("thinking-panel");

private static HTMLCanvasElement sampleCanvas = document.getElementById("myCanvas").cast();

private static HTMLCanvasElement sampleCanvas2 = document.getElementById("myCanvas2").cast();

private Client() {

}

public static void main(String[] args) {

helloButton.listenClick(evt -> drawLine());

rectButton.listenClick(evt -> drawRect());

}

private static void drawLine() {

helloButton.setDisabled(true);

thinkingPanel.getStyle().setProperty("display", "");

XMLHttpRequest xhr = XMLHttpRequest.create();

xhr.onComplete(() -> {

Object obj = JSONValue.parse(xhr.getResponseText());

JSONArray array = (JSONArray)obj;

for (int i = 0; i < array.size(); i++)

{

JSONObject jsonobj = (JSONObject)array.get(i);

responsePanel.withText(jsonobj.toJSONString());

responsePanel.appendChild(sampleCanvas);

double x1coord = Double.parseDouble(jsonobj.get("x1coord").toString());

double x2coord = Double.parseDouble(jsonobj.get("x2coord").toString());

double y1coord = Double.parseDouble(jsonobj.get("y1coord").toString());

double y2coord = Double.parseDouble(jsonobj.get("y2coord").toString());

CanvasRenderingContext2D context = (CanvasRenderingContext2D)sampleCanvas.getContext("2d");

context.moveTo(x1coord, y1coord);

context.lineTo(x2coord, y2coord);

context.stroke();

}

helloButton.setDisabled(false);

thinkingPanel.getStyle().setProperty("display", "none");

});

xhr.open("GET", "hello");

xhr.send();

}

private static void drawRect() {

rectButton.setDisabled(true);

thinkingPanel.getStyle().setProperty("display", "");

XMLHttpRequest xhr = XMLHttpRequest.create();

xhr.onComplete(() -> {

Object obj = JSONValue.parse(xhr.getResponseText());

JSONArray array = (JSONArray)obj;

JSONObject jsonobj = (JSONObject)array.get(1);

responsePanel2.withText(jsonobj.toJSONString());

responsePanel2.appendChild(sampleCanvas2);

double x1coord = Double.parseDouble(jsonobj.get("x1coord").toString());

double x2coord = Double.parseDouble(jsonobj.get("x2coord").toString());

double y1coord = Double.parseDouble(jsonobj.get("y1coord").toString());

double y2coord = Double.parseDouble(jsonobj.get("y2coord").toString());

CanvasRenderingContext2D context2 = (CanvasRenderingContext2D)sampleCanvas2.getContext("2d");

context2.fillRect(x1coord, y1coord, x2coord, y2coord);

rectButton.setDisabled(false);

thinkingPanel.getStyle().setProperty("display", "none");

});

xhr.open("GET", "hello");

xhr.send();

}

}

Index.html:

<!DOCTYPE html>

<html>

<head>

<title>Hello web application</title>

<meta http-equiv="Content-Type" content="text/html;charset=utf-8">

<script type="text/javascript" charset="utf-8" src="teavm/runtime.js"></script>

<script type="text/javascript" charset="utf-8" src="teavm/classes.js"></script>

</head>

<body onload="main()">

<div><button id="hello-button">Line</button></div>

<div><button id="rect-button">Rectangle</button></div>

<div id="hello-panel"></div>

<div id="thinking-panel" style="display:none"><i>Server is thinking...</i></div>

<div id="response-panel">

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;"></canvas>

</div>

<div id="response-panel2">

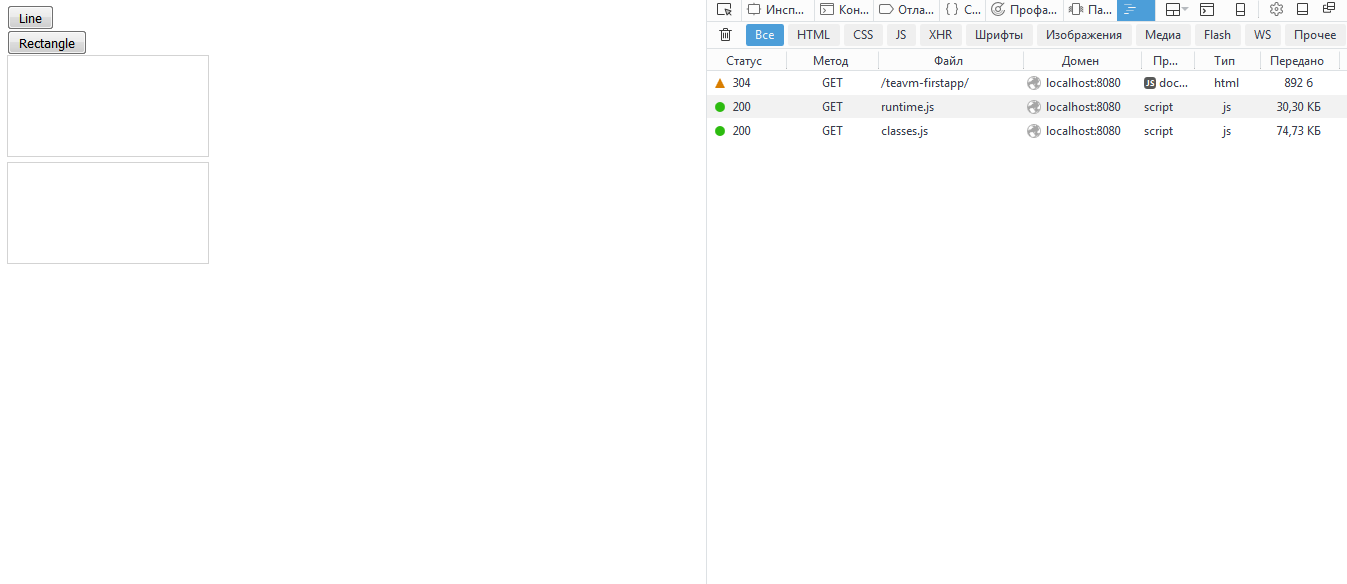
<canvas id="myCanvas2" width="200" height="100" style="border:1px solid #d3d3d3;"></canvas>

</div>

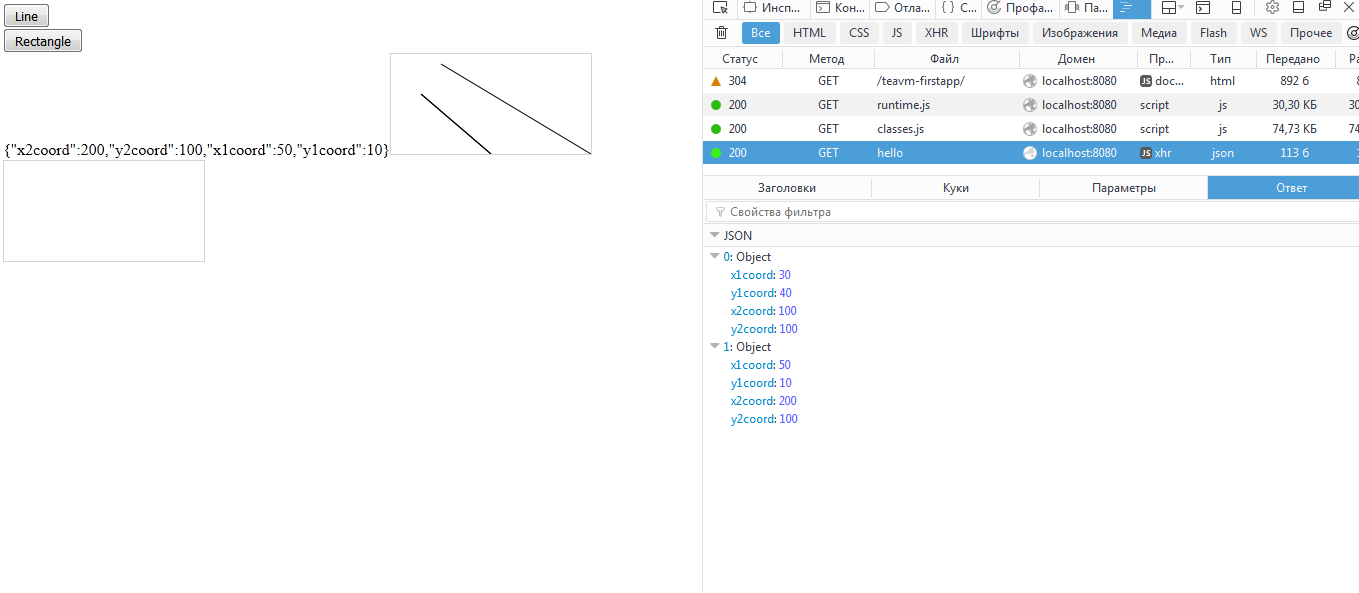
</body>

</html>

Итоговая стартовая веб-страница:



Итоговая страница после нажатия первой кнопки:



Итоговая страница после нажатия второй кнопки:

