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 программного обеспечения.

Разработка OSGi микро-сервиса графического языка для графических подсказок AutoComplete

Студент группы ИУ3-72 (подпись) 19.02.2017 А.И. Полторак

Руководитель курсовой работы (подпись) 22.12.2016 А.М. Иванов

Москва, 2016

В рамках курсовой работы был изучен обмен данными о подсказках между веб- страницей и веб-сервером и выяснено, что последовательность инициализации панелей жестко задана на языке JavaScript в файле app.min.js

За основу была взята система https://www.draw.io/ из github (https://github.com/agentlab/draw.io/tree/master). В файл app.min.js были внесены изменения. В результате в системе появились подсказки Autocomplete при наведении на фигуры на холсте (рисунок 2-3). В зависимости от типа фигуры, передается json объект с соответствующим ей списком подсказок. На сайте https://localhost:8443/mxgraph\_com/ была добавлена новая кнопка Add Custom Shapes Pane, добавляющая новую панель с фигурами.

В приложении 1 представлен код TooltipService.java, который формирует json объект, на основе «типа фигуры» переданного в запросе от клиента

В приложении 2 представлен код app.min.js, в котором отправляется запрос на сервер, и интерпретируется полученный json объект и отображается подсказки на страницу с помощью js.

Рисунок 1. Вид проекта в среде разработки Eclipse.

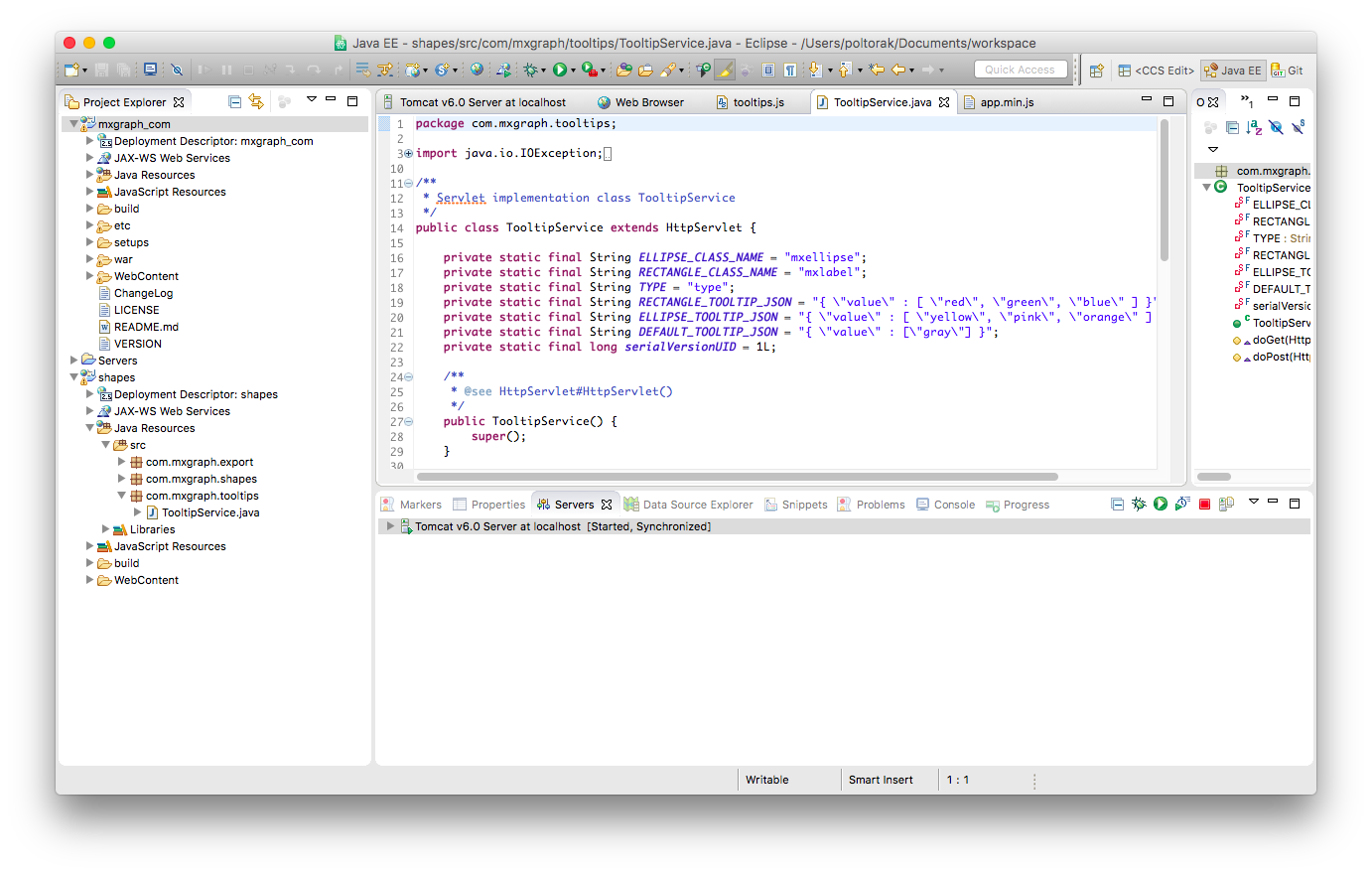


Рисунок 2. Подсказки, соответствующие типу фигуры «прямоугольник».

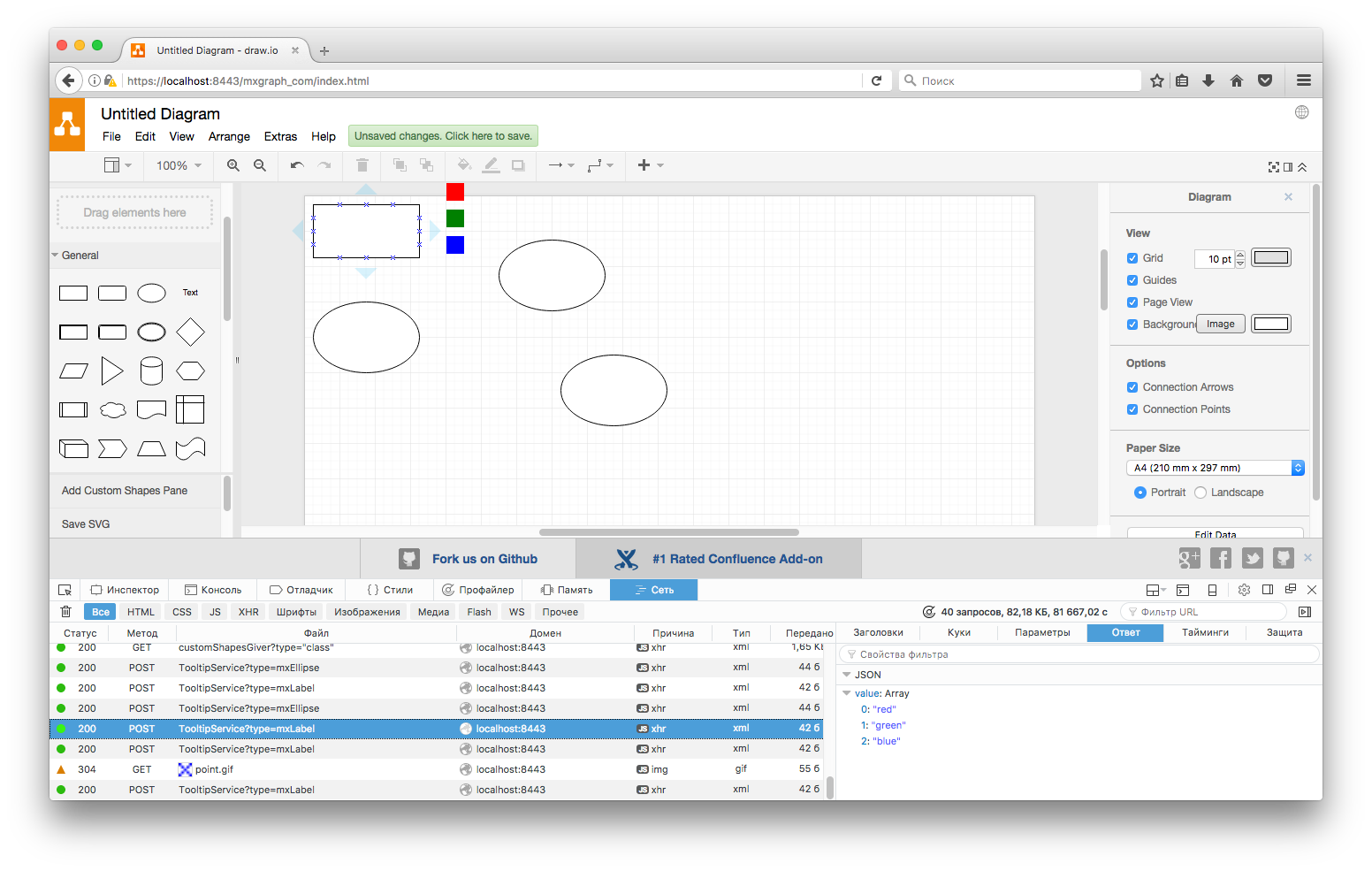
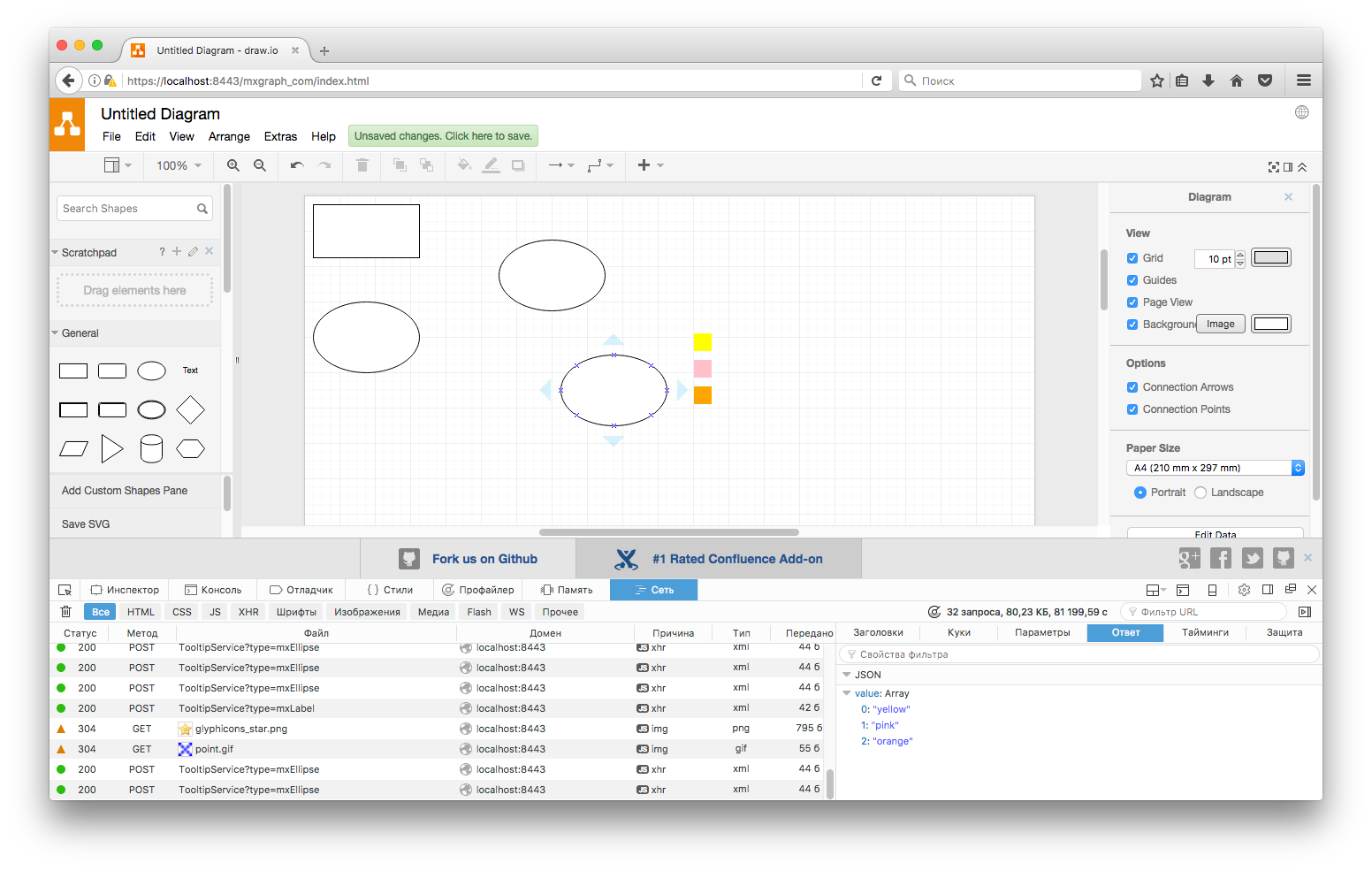


Рисунок 3. Подсказки, соответствующие типу фигуры «эллипс».

Приложение 1

package com.mxgraph.tooltips;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* Servlet implementation class TooltipService

\*/

public class TooltipService extends HttpServlet {

private static final String ELLIPSE\_CLASS\_NAME = "mxellipse";

private static final String RECTANGLE\_CLASS\_NAME = "mxlabel";

private static final String TYPE = "type";

private static final String RECTANGLE\_TOOLTIP\_JSON = "{ \"value\" : [ \"red\", \"green\", \"blue\" ] }";

private static final String ELLIPSE\_TOOLTIP\_JSON = "{ \"value\" : [ \"yellow\", \"pink\", \"orange\" ] }";

private static final String DEFAULT\_TOOLTIP\_JSON = "{ \"value\" : [\"gray\"] }";

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public TooltipService() {

super();

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// Does nothing

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String type = request.getParameter(TYPE);

PrintWriter out = response.getWriter();

switch (type.toLowerCase()) {

case RECTANGLE\_CLASS\_NAME:

out.println(RECTANGLE\_TOOLTIP\_JSON);

break;

case ELLIPSE\_CLASS\_NAME:

out.print(ELLIPSE\_TOOLTIP\_JSON);

break;

default:

out.print(DEFAULT\_TOOLTIP\_JSON);

break;

}

out.flush();

out.close();

}

}

Приложение 2

var xmlHttp = new XMLHttpRequest();

xmlHttp.onreadystatechange = function() {

if (xmlHttp.readyState == 4 && xmlHttp.status == 200)

{

var mainDiv = document.createElement("div");

mainDiv.style.width = "20px";

mainDiv.style.height = "100px";

mainDiv.style.position = "absolute";

var positionLeft = parseInt(rightArrow.style.left) + 20;

var positionTop = parseInt(topArrow.style.top);

mainDiv.style.left = positionLeft + "px";

mainDiv.style.top = positionTop + "px";

mainDiv.className = "MyTooltipBar"

var jsonObject = JSON.parse(xmlHttp.responseText);

for (var tooltipIndex = 0; tooltipIndex < jsonObject.value.length; tooltipIndex++)

{

var subDiv = document.createElement("div");

subDiv.style.width = mainDiv.style.width;

subDiv.style.height = subDiv.style.width;

subDiv.style.background = jsonObject.value[tooltipIndex];

subDiv.style.marginBottom = "10px";

mainDiv.appendChild(subDiv);

}

parent.appendChild(mainDiv);

}

}

xmlHttp.open("POST", "https://localhost:8443/shapes/TooltipService?type=" + this.prev.shape.constructor.name, true);

xmlHttp.send(null);