武汉纺织大学

Web应用开发课程设计

**基于API的智能机器人聊天系统**

**学 院： 数学与计算机学院**

**班 级： 软件11801**

**姓 名： 田茁**

**学 号： 1804240710**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年12月15日**

目 录

[1 需求分析 1](#_Toc20910)

[1.1聊天窗口页面 1](#_Toc932)

[1.2消息发送按钮的功能 2](#_Toc2919)

[1.3向API发送和接收消息的功能 2](#_Toc16416)

[1.4显示接收API的消息 2](#_Toc3869)

[1.5监听输入框的回车键 2](#_Toc10012)

[1.6消息窗口的自动滚动定位功能 2](#_Toc10985)

[2 需求分析 2](#_Toc27666)

[2.1用户用例图 2](#_Toc29467)

[2.2 UML类图（Class Diagram） 3](#_Toc11744)

[2.3 UML时序图（Sequence Diagram） 3](#_Toc16288)

[2.3.1 系统登录 3](#_Toc28455)

[2.4 UML活动图（Activity Diagram） 4](#_Toc29039)

[2.4.1 用户发送消息 4](#_Toc14959)

[3 系统实现 6](#_Toc10394)

[3.1 项目结构 6](#_Toc6275)

[3.2 Controller类APIController.java 6](#_Toc6076)

[3.3 utils类APIHelper.java 7](#_Toc29353)

[3.4 mainJS.js 8](#_Toc18377)

[3.4.1 jqAjaxSendMessage()函数 8](#_Toc1953)

[3.4.2 msgFromMe(string)和msg(string)函数 9](#_Toc30621)

[3.4.3 listenEnterKey()函数 9](#_Toc4496)

[3.4.4 scrollToBottom()函数 10](#_Toc23732)

[4 系统测试 10](#_Toc27709)

[4.1消息发送按钮的功能 10](#_Toc30451)

[4.2监听回车键发送的功能 11](#_Toc17999)

[4.3自动滚动到底部功能 11](#_Toc30573)

[5 系统总结 11](#_Toc32113)

[5.1主要存在的问题及解决过程 12](#_Toc31358)

[5.1.1 前端页面使用API存在跨域问题 12](#_Toc1148)

[5.1.2 前端页面按钮鼠标移入样式的问题 13](#_Toc6083)

[5.2系统存在的不足 14](#_Toc27678)

[5.3本门课程的收获 14](#_Toc7851)

[5.4以后发展方向 15](#_Toc10650)

[5.5自评成绩 15](#_Toc19640)

# 1 需求分析

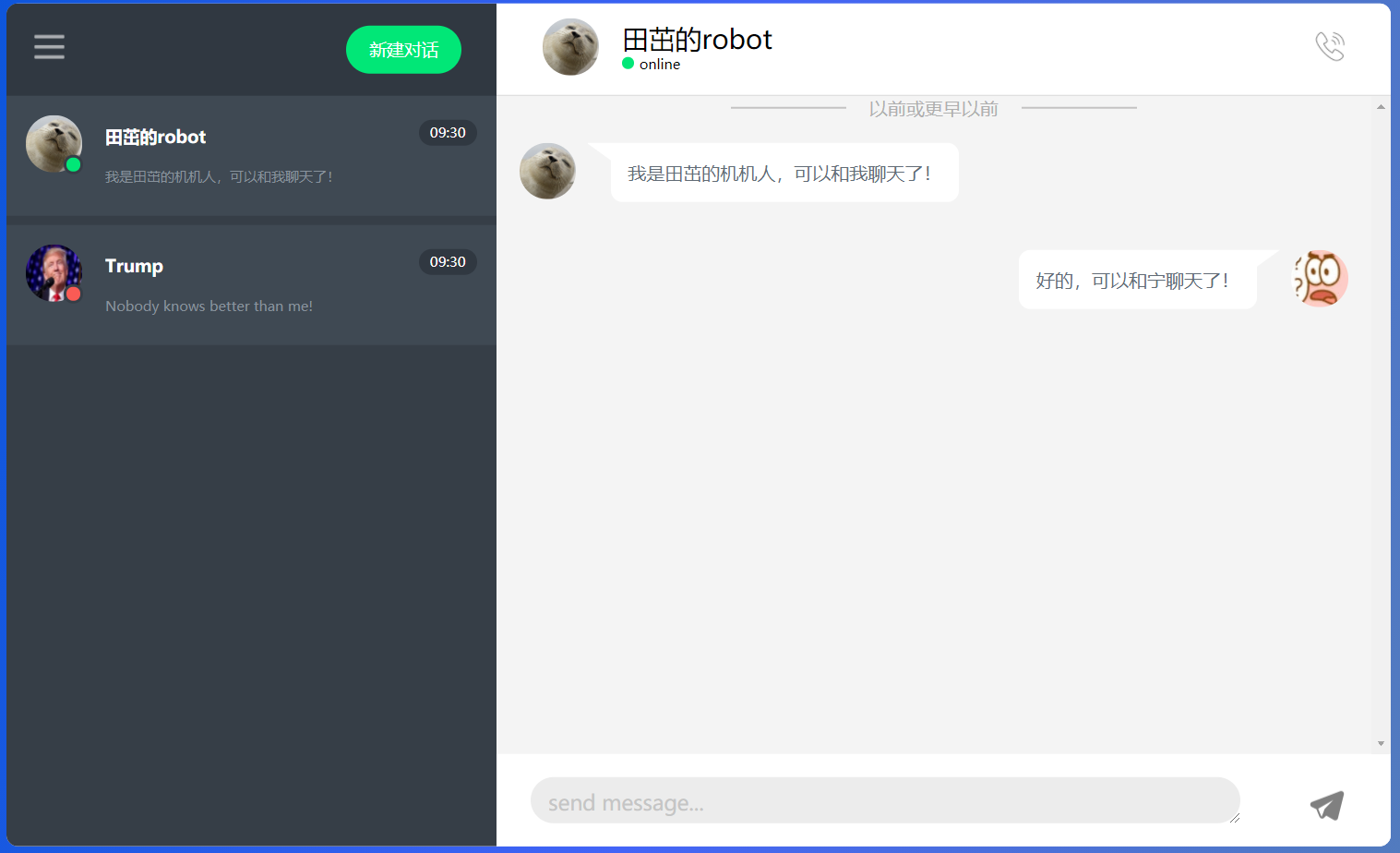
设计一个智能机器人页面（参考下图）：



前端显示页面，输入文字点击发送后ajax方式调用服务器，在服务器程序中调用图灵机器人API，把API调用的结果返回到前端页面显示

## 1.1聊天窗口页面

当程序运行时，在浏览器地址栏输入地址http://localhost:8080/ChatWithTz/，可以看见机器人聊天窗口页面，我的设计如下图：



## 1.2消息发送按钮的功能

当用户点击发送按钮时，会在消息窗体中新增自己的聊天气泡，并且清空输入框中的内容;

## 1.3向API发送和接收消息的功能

当用户点击发送按钮并显示消息气泡后，会在后端的java程序中以GET请求方式向青云客智能聊天机器人API发送用户输入的消息的数据，并用Response接收API返回的数据；

## 1.4显示接收API的消息

在Response接收到API返回的数据并且status是200时，对数据进行处理，在消息窗口中新增机器人的聊天气泡；

## 1.5监听输入框的回车键

用户在编辑完消息后，直接敲击回车键就能发送消息，起到与点击发送按钮时同样的效果；

## 1.6消息窗口的自动滚动定位功能

在用户发送消息新增消息气泡、API返回消息并新增消息气泡后，消息窗口要自动滚到最下面的那条消息，不能新增了以后非要用户自己滚轮滚半天；

# 2 需求分析

## 2.1用户用例图

系统分为用户，机器人两个角色，各角色的用户用例如下图2-1所示：

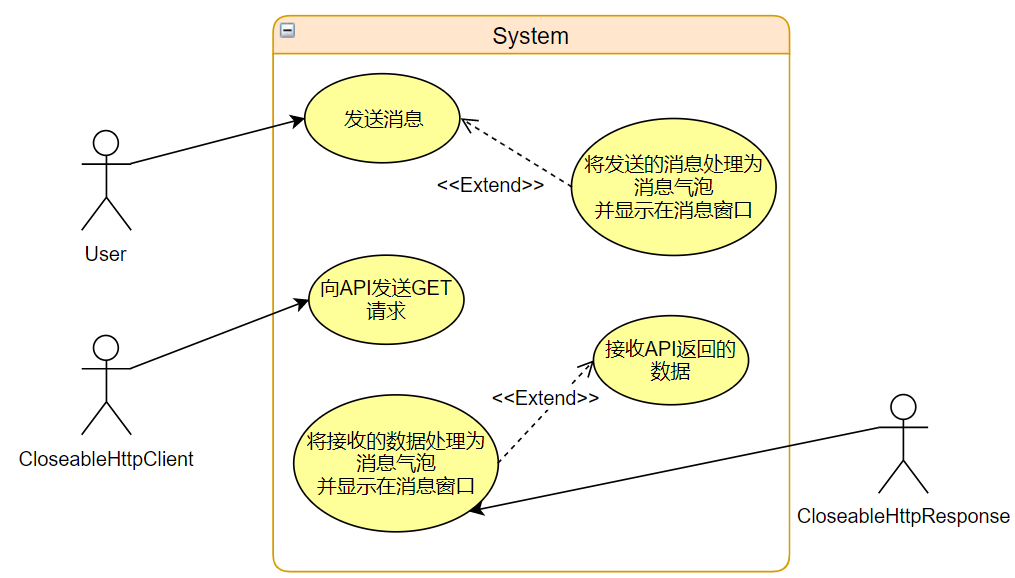


图2-1 用户用例图

## 2.2 UML类图（Class Diagram）

对于API机器人对话功能模块，共设计如下2个类。

* APIController：接收前端页面ajax传来的用户输入框中的信息，调用APIHeper请求API的数据，数据处理后以map的形式放在response中返回到前端ajax中；
* APIHeper：在有参构造函数中，参数是APIController传进来的用户前端输入的消息(String类型)，在该方法中创建Apache包下面CloseableHttpClient类、HttpGet类和CloseableHttpResponse类，来向青云客智能聊天机器人API发送GET请求，并接受返回的数据，数据处理后返回的是机器人的消息(String类型).

各类的结构及类之间的关系如图2-5所示：

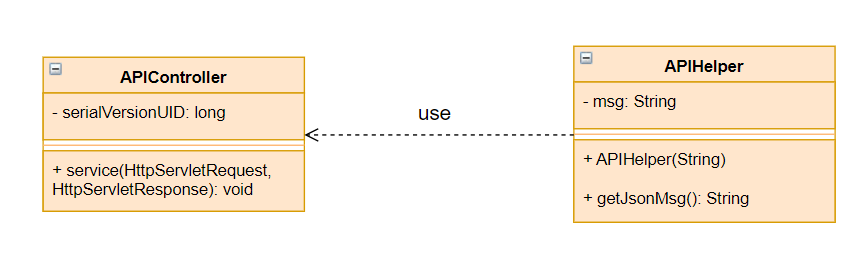
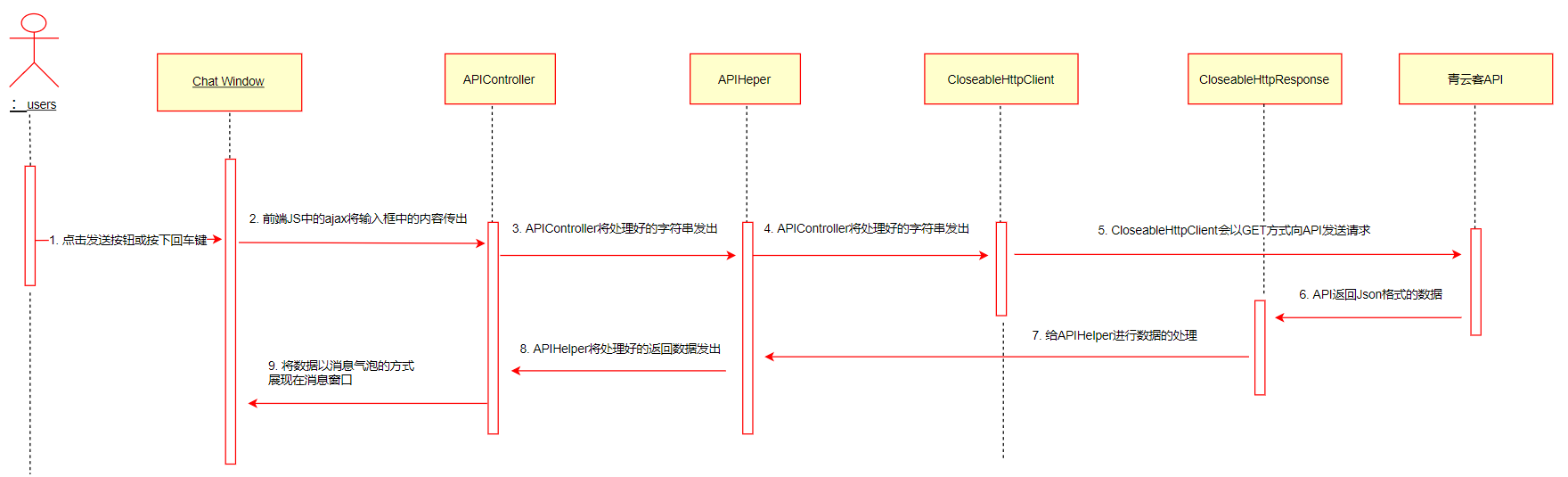


图2-5 用户功能类图

## 2.3 UML时序图（Sequence Diagram）

### 2.3.1 系统登录



## 2.4 UML活动图（Activity Diagram）

### 2.4.1 用户发送消息

用户在浏览器地址栏输入相应地址后进入聊天页面，在文本框中输入文本，点击发送按钮或者按下回车键后，后端程序向青云客智能聊天机器人API发送GET请求，接收到返回数据后对StatusCode进行判断，若为200则新增机器人消息，否则就提示机器人无响应，具体流程如图下所示。

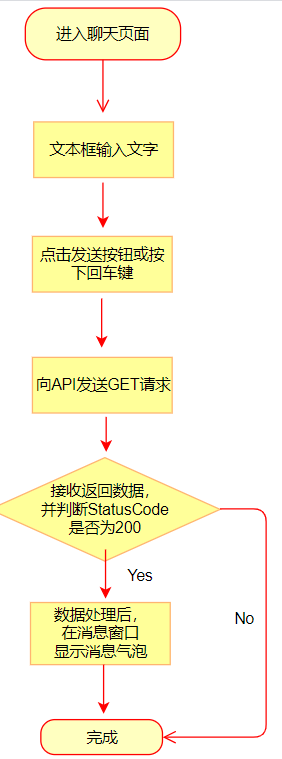
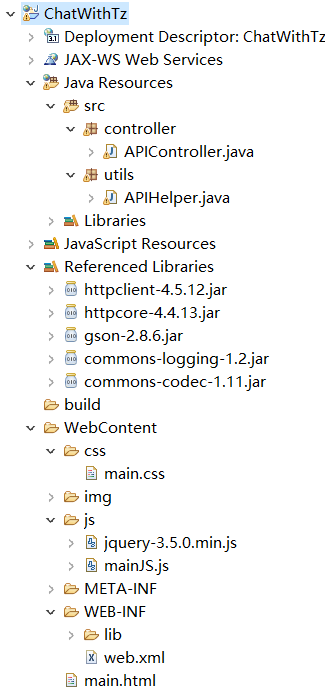


图2-2 用户发送消息流程图

# **3 系统实现**

## 3.1 项目结构



## 3.2 Controller类APIController.java

该类是一个servlet，前端JS代码中将用户文本框中用户输入的信息传入到此servlet中，然后调用APIHelper.java进行API的调用，将API传回来的数据处理后，放入Map再以response的形式返回给前端页面；

**@WebServlet("/APIController")**

**public class APIController extends HttpServlet {**

**private static final long serialVersionUID = 1L;**

**protected void service(HttpServletRequest request, HttpServletResponse response)**

**throws ServletException, IOException {**

**// 设置请求参数编码格式为utf-8，防止中文乱码**

**request.setCharacterEncoding("utf-8");**

**// 通过request获取ajax传来的msg值**

**String msg = request.getParameter("msg");**

**System.out.println("接收到的msg为：" + msg);**

**// 准备返回的map**

**Map<String, Object> map = new HashMap<String, Object>();**

**APIHelper apiHelper = new APIHelper(msg);**

**try {**

**String jsonMsg = apiHelper.getJsonMsg();**

**System.out.println("servle中接收到处理后的msg串为："+jsonMsg);**

**map.put("msg", jsonMsg);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**String jsonStr = new Gson().toJson(map);**

**// 字符流输出字符串**

**response.setContentType("text/html;charset=utf-8");**

**PrintWriter out = response.getWriter();**

**out.print(jsonStr);**

**out.flush();**

**out.close();**

**}**

**}**

## 3.3 utils类APIHelper.java

描述该文件作用，并对里面的关键代码进行解释

**public class APIHelper {**

**private String msg;**

**public APIHelper(String msg) {**

**this.msg = msg;**

**}**

**public String getJsonMsg() throws Exception, IOException {**

**String result="";**

**// 访问API**

**CloseableHttpClient closeableHttpClient = HttpClients.createDefault();**

**HttpGet httpGet = new HttpGet("http://api.qingyunke.com/api.php?key=free&appid=0&msg=" + msg);**

**CloseableHttpResponse closeableResponse = null;**

**try {**

**closeableResponse = closeableHttpClient.execute(httpGet);**

**if (closeableResponse.getStatusLine().getStatusCode() == 200)**

**{**

**HttpEntity entity = closeableResponse.getEntity();**

**String json = EntityUtils.toString(entity, "utf-8");**

**JsonParser jsonParser = new JsonParser();**

**JsonElement jsonElement = jsonParser.parse(json);**

**JsonObject jsonObject = jsonElement.getAsJsonObject();**

**String msgJson = jsonObject.get("content").toString();**

**System.out.println("API返回的content为：" + msgJson);**

**result = msgJson.replace("\"", "");**

**}**

**return result;**

**} finally {**

**if (closeableResponse != null) {**

**closeableResponse.close();**

**}**

**closeableHttpClient.close();**

**}**

**}**

**}**

## 3.4 mainJS.js

### 3.4.1 jqAjaxSendMessage()函数

此js文件中的**jqAjaxSendMessage()**函数绑定在前端的按钮onClick事件上，



按下回车键也能触发该函数。该函数调用生成消息气泡的函数、清除文本框的数据、以ajax的形式异步向servlet传输用户输入的信息、接收servlet返回的数据并生成对方的气泡；

**function jqAjaxSendMessage() {**

**//获取输入框的值**

**var userMessage = $("textarea").val();**

**$("textarea").val("");**

**//增加自己的气泡**

**msgFromMe(userMessage);**

**$.ajax({**

**type: "POST",**

**url: "APIController",**

**data: {key:"free",appid:0,msg:userMessage},**

**dataType:"json",**

**success: function (response) {**

**console.log(response.msg);**

**//增加对方的气泡**

**msg(response.msg);**

**}**

**});**

**}**

**3.4.2 msgFromMe(string)和msg(string)函数**

**/\* 函数：增加自己的气泡 \*/**

**function msgFromMe(userMessage){**

**var str = "<div class='message fromme'>";**

**str = str + "<div class='user-head'><img src='img/myself.jpg'></div>";**

**str = str + "<div class='content'>"+userMessage+"</div>";**

**str = str+"</div>"**

**$("#messages").append(str);**

**}**

**/\* 函数：增加对方的气泡 \*/**

**function msg(userMessage){**

**var str = "<div class='message'>";**

**str = str + "<div class='user-head'><img src='img/currentUser.jpg'></div>";**

**str = str + "<div class='content'>"+userMessage+"</div>";**

**str = str+"</div>"**

**$("#messages").append(str);**

**}**

**3.4.3 listenEnterKey()函数**

此js文件中的**listenEnterKey()**函数绑定在前端的文本框的onkeydown事件上，



用于监听文本框中回车键的按下，用于发送消息

**/\* textarea监听回车键 \*/**

**function listenEnterKey()**

**{**

**var keycode;**

**if(event.which)**

**{**

**keycode = event.which; //按键代码**

**}**

**//按键名称：var keychar = String.fromCharCode(x);**

**if(keycode == 13)**

**{**

**jqAjaxSendMessage();**

**/\* 取消textarea的换行 \*/**

**event.returnValue = false;**

**return false;**

**}**

**}**

**3.4.4 scrollToBottom()函数**

**/\* 自动滚动到底部 \*/**

**function scrollToBottom()**

**{**

**var now = new Date();**

**var div = document.getElementById('messages');**

**//div.innerHTML = div.innerHTML + 'time\_' + now.getTime() + '<br />';**

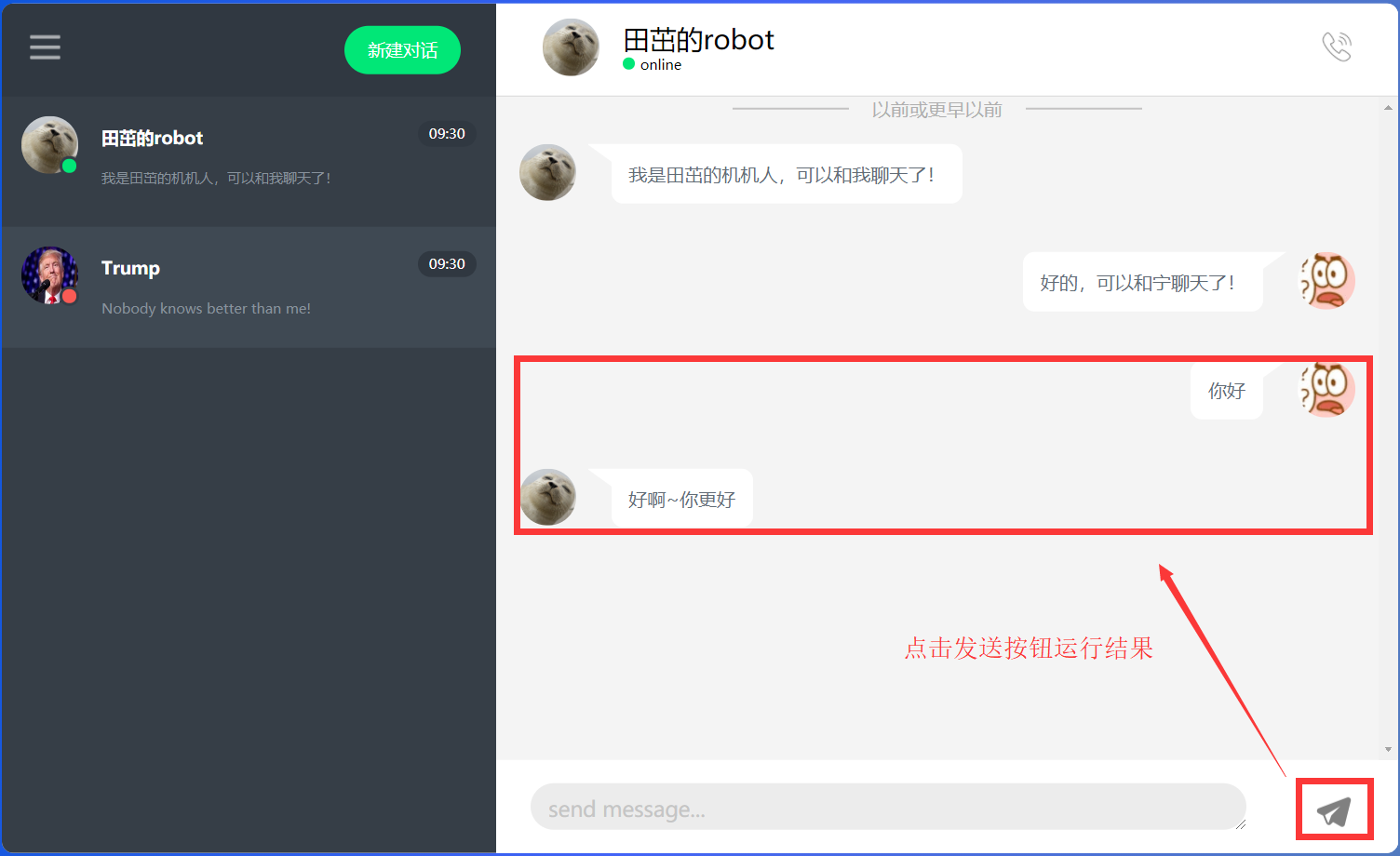
**div.scrollTop = div.scrollHeight;**

**}**

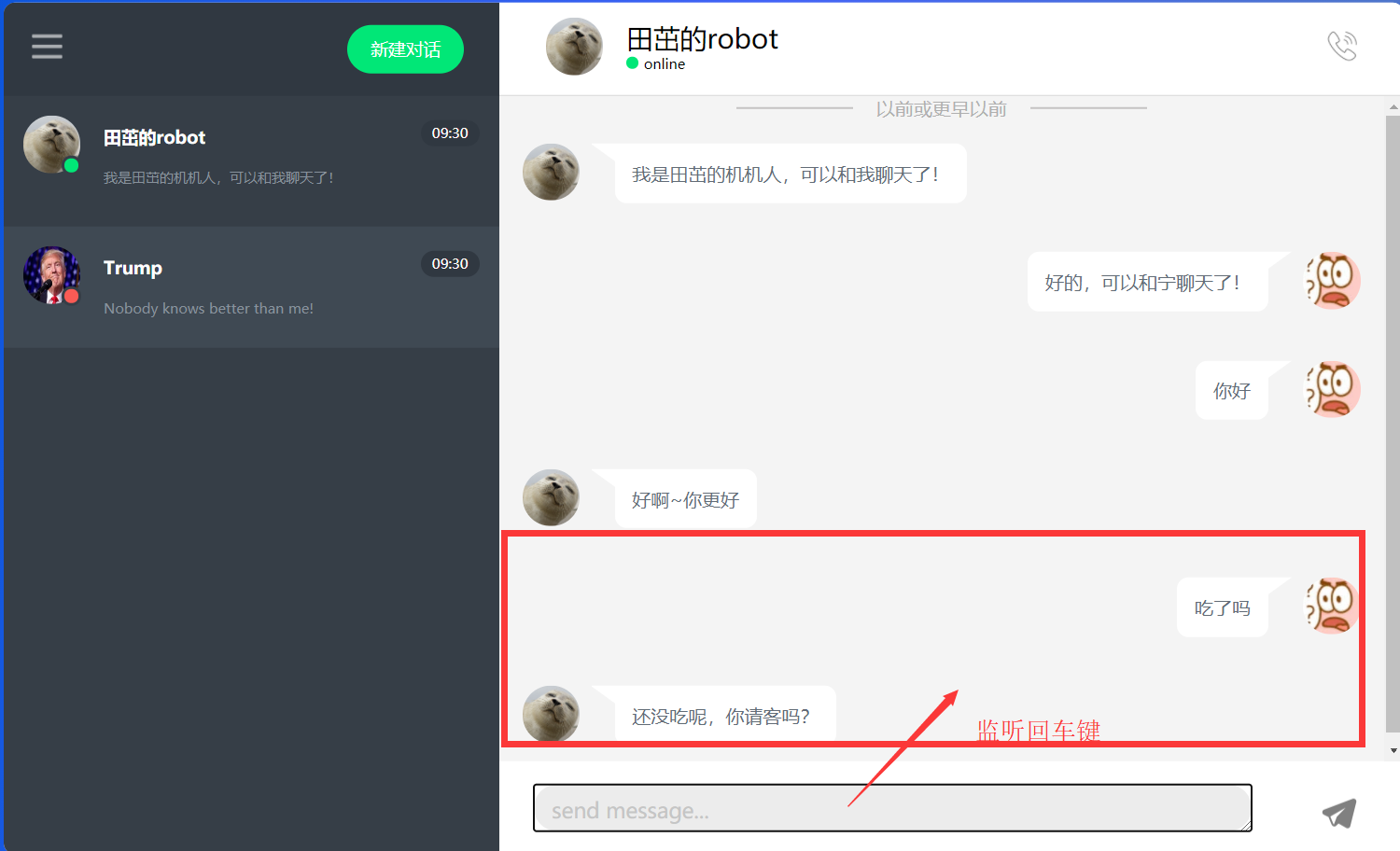
# **4 系统测试**

设计测试用例，给出程序每个功能模块的运行结果截图。

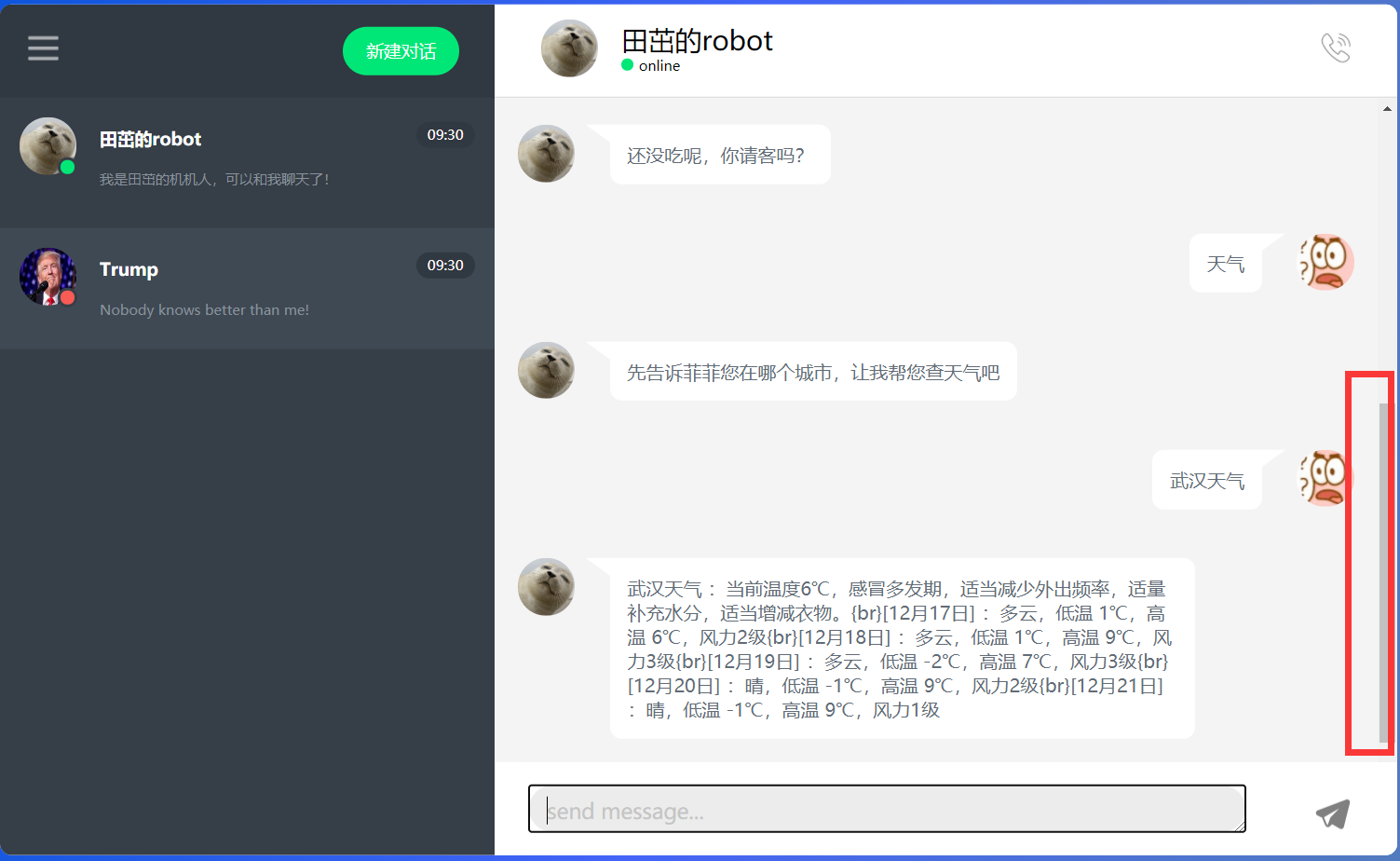
## 4.1消息发送按钮的功能



## 4.2监听回车键发送的功能



## 4.3自动滚动到底部功能



# **5 系统总结**

主要问题的解决过程、系统存在的不足，本门课程的收获，课程的建议，以后的发展方向，自评成绩。

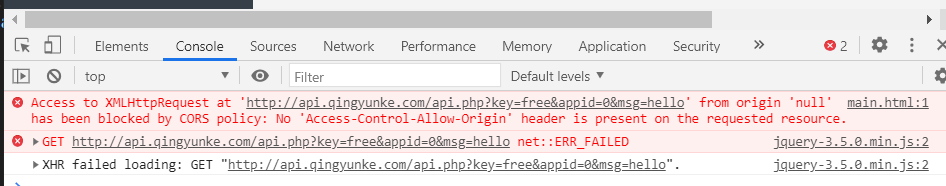
## 5.1主要存在的问题及解决过程

### 5.1.1 前端页面使用API存在跨域问题

我们平时开发前端页面的时候，一般情况下都是在JS代码中用ajax填上URL向后端java程序发送数据，所以我在做这个API调用时，我第一反应是在前端ajax的url中直接填上API的地址，想试一试能不能直接在前端页面中就向API发送请求，像这样：



马上就出现问题了，浏览器报错如下：

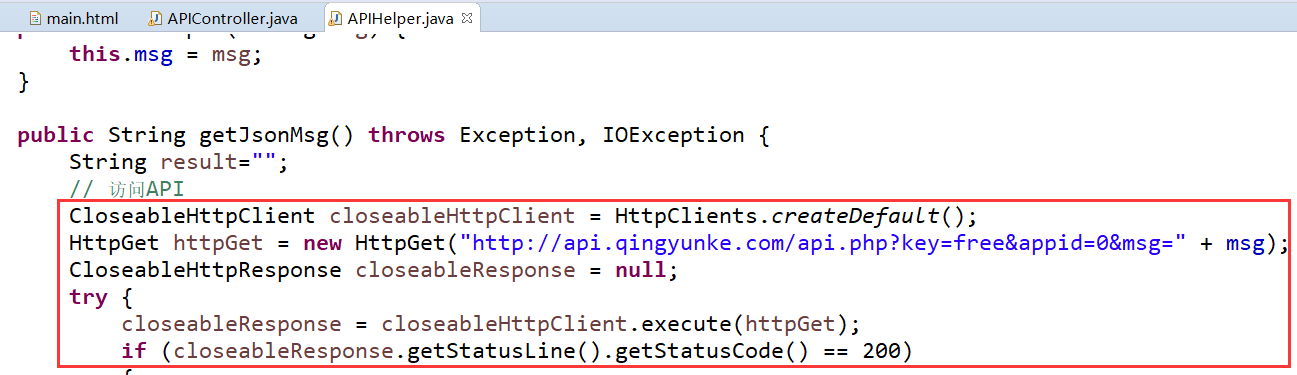


在网上查后发现，在网络协议中有一项CORS协议，只要协议、域名、端口有任何一个不同，都被当作是不同的域。为了安全起见，不同域不能直接访问。

前端页面不能直接访问，于是我便转向在后端的java程序中访问，将前端JS代码中将用户文本框中用户输入的信息传入到servlet中，然后调用APIHelper.java进行API的调用，将API传回来的数据处理后，放入Map再以response的形式返回给前端页面。于是前端ajax的URL填上了自己写的servlet：

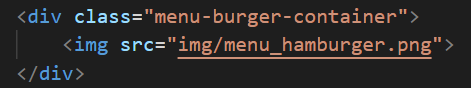


Java程序是可以直接向API发送请求的，不会存在跨域问题，我就写了一个APIHelper类用于调用，如下：

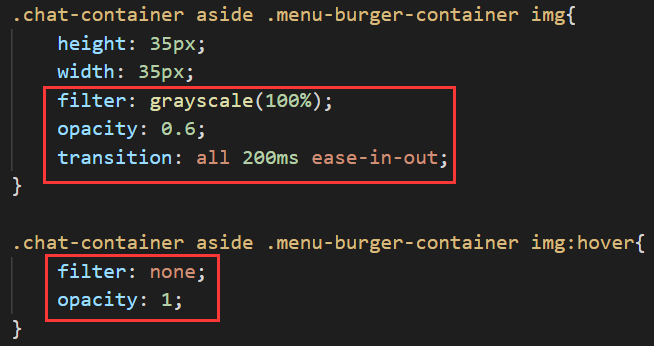


### 5.1.2 前端页面按钮鼠标移入样式的问题

前端页面中经常会有用图片做的按钮，在web中，会存在鼠标移入按钮样式发生变化的情况，在用bootstrap框架的时候，加入对应的class就能直接完成样式的设定，但是我们如果不用前端框架，纯用css去写该怎么写呢？我想到了改变图片灰度的方法：当鼠标未移入时，给按钮图片设置灰度，移入后在hover中取消对灰度的设置，具体如下：



对应的css代码如下：



## 5.2系统存在的不足

这学期学到了很多JavaWeb开发方面的知识，包括了前端和后端的以及各种框架的掌握，但我在拿到这个题目的时候感觉能用到的技术很少，不能展现出本学期完全掌握的内容，甚至连数据库都没用上(∩\_∩)，于是我打算不只是做机器人API的调用，还想做成一个聊天系统，可以和其他用户进行消息的传输。从前端样式就可看出我有这个打算，例如已经做出了消息列表，还有给每个用户都增加了在线的状态：



但是在开始设计数据库表的时候我发现，即使我每条消息都保存了，也无法做到消息框中消息的同步。于是我想到了设置一个时间戳，每隔一秒钟就从数据库中读取一次对话消息，在消息窗口新增消息，如果用户一多服务器哪里忙得过来......但是同学提醒我说，这种聊天室一般都是用socket编程。我接下来会多花点时间将单一的“API机器人聊天”做成“聊天室”。

## 5.3本门课程的收获

这学期开始前就对JavaWeb有了一些了解，十分想学习SSM框架，但是看见网上spring、springMVC、mybatis的教程是在太多了，有不知从何学起。十分幸运的是聂老师这学期就讲ssm框架。而且聂老师不是像网上的那些教程把spring和springMVC从头讲起，而是讲了这些框架的原理和概念让我们有了一个基本了解后，再把那些真正在开发中要用得到的东西拿出来讲，我觉得这一点是最棒的。

从一开始接触JavaWeb每个功能都单独写一个servlet、每一个数据库操作都写一遍驱动的连接，到现在能用MVC三层架构，将项目后端分为Controller，Service，Serviceimpl，dao，mapper和vo，我整个开发思路就更加清晰了。第一次搭建SSM框架的时候，各种配置文件都看不是太懂，但跟着老师走一遍后，了解了每个文件是干什么用的，自己下来又搭建了几遍，对配置文件也熟悉了。

## 5.4以后发展方向

在考研之前我十分想再多提升一下自己JavaWeb开发的实战技能！

打算放假就开始备战考研了，对自己实力还不是特别清楚，想先复习一下数学、英语两门公共课后再根据自己情况选择目标院校。

## 5.5自评成绩

我这次分到的题目只有API的调用，有点可惜，没能把学到的后端的SSM框架还有前端的bootstrap和Vue用上，老师可以在平时作业中看我掌握的情况^\_^。我也尽可能的完善了机器人聊天界面的功能，提升了用户的体验。自我评分：90分。