武汉纺织大学

Web应用开发课程设计

**在线翻译系统**

**学 院： 数学与计算机学院**

**班 级： 物联网11804**

**姓 名： 郑智峰**

**学 号： 1804280433**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年12月18日**

目录

[1.需求分析 3](#_Toc59207519)

[2.系统设计 3](#_Toc59207520)

[2.1 前端技术 3](#_Toc59207521)

[2.1.1 bootstrap框架 3](#_Toc59207522)

[2.1.2 jQuery库 3](#_Toc59207523)

[2.1.3 vue框架 4](#_Toc59207524)

[2.2 后端技术 4](#_Toc59207525)

[2.2.1 spring框架 4](#_Toc59207526)

[3.系统实现 5](#_Toc59207527)

[3.1项目结构 5](#_Toc59207528)

[3.2配置文件 5](#_Toc59207529)

[3.2.1applicationContext.xml文件 5](#_Toc59207530)

[3.2.2 springmvc.xml文件 6](#_Toc59207531)

[3.3 controller类TranslateController.java 6](#_Toc59207532)

[3.4 service服务层 7](#_Toc59207533)

[3.4.1 service服务层impl文件夹下TranslateServiceImpl.java 7](#_Toc59207534)

[3.4.2 service服务层 TranslateService.java 8](#_Toc59207535)

[3.5工具包Util 8](#_Toc59207536)

[3.5.1 HttpUtil.java 8](#_Toc59207537)

[3.5.2 Md5Util.java 11](#_Toc59207538)

[4.系统测试 12](#_Toc59207539)

[4.1 运行结果展示 12](#_Toc59207540)

[5.技术难点 13](#_Toc59207541)

[5.1 获得百度账号的ID和密钥 13](#_Toc59207542)

[5.2 获得英文和中文的翻译代码 14](#_Toc59207543)

[5.3 生成签名和拼接完整请求连接 14](#_Toc59207544)

[5.4 TomCat服务器的编码修改 15](#_Toc59207545)

[6.心得总结 15](#_Toc59207546)

# 1.需求分析

设计一个在线翻译系统，实现前端页面选择语言，输入文字，发送给服务器端，服务器端调用翻译API接口，将翻译结果传递至前端页面进行显示。

2.系统设计

## 2.1 前端技术

### 2.1.1 bootstrap框架

Bootstrap框架可以提高我们的开发效率，规范名称定义，便于维护，规范项目的开发流程，css代码更清晰、简单，html代码更合理。

基本结构：Bootstrap 提供了一个带有网格系统、链接样式、背景的基本结构。这将在Bootstrap 基本结构部分详细讲解。

bootstrap架构：

　　CSS：Bootstrap 自带以下特性：全局的 CSS设置、定义基本的 HTML 元素样式、可扩展的 class，以及一个先进的网格系统。这将在Bootstrap CSS部分详细讲解。

　　组件：Bootstrap 包含了十几个可重用的组件，用于创建图像、下拉菜单、导航、警告框、弹出框等等。这将在布局组件部分详细讲解。

　　JavaScript 插件：Bootstrap包含了十几个自定义的jQuery 插件。您可以直接包含所有的插件，也可以逐个包含这些插件。这将在Bootstrap插件部分详细讲解。

定制：您可以定制Bootstrap的组件、LESS 变量和jQuery 插件来得到您自己的版本。

使用途径：

我们的项目主要是使用bootstrap框架的前端UI库，帮助我们更好的设计前端界面。

### 2.1.2 jQuery库

jQuery是一个JavaScript库，有助于简化和标准化JavaScript代码和HTML元素之间的交互。JavaScript允许网站具有交互性和动态性，jQuery是一种有助于简化流程的工具。

同时jQuery也是一个紧凑，快速且功能丰富的JavaScript库。它通过易于使用的API在大量浏览器中运行，使得HTML文档遍历和操作，事件处理，动画和Ajax更加简单。jQuery结合了多功能性和可扩展性，改变了数百万人编写JavaScript的方式。换句话说，由于jQuery，JavaScript以简化形式呈现。阅读起来相当舒服。

使用途径：

发送ajax，易于使用的API在大量浏览器中运行。

### 2.1.3 vue框架

当前端和数据做一些操作的时候，可以通过AJAX请求对后端做数据持久化，不需要刷新整个页面，只需要改动DOM里需要改动的那部分数据。特别是移动端应用场景，刷新页面太昂贵，会重新加载很多资源，虽然有些会被缓存，但是页面的DOM,JS,CSS都会被页面重新解析一遍，因此移动端页面通常会做出SPA单页应用。

使用途径：

整个前端使用时数据的双向绑定、指令和插件化，是项目前端框架的核心。

## 2.2 后端技术

### 2.2.1 spring框架

Spring框架主要用于与其他技术(struts,hibernate等)进行整合,可将应用程序中的Bean组件实现低耦合关联。最终可以提高系统扩展和维护性。将来我们利用Spring框架管理系统的各个组件(Action,Service,DAO)。采用Spring的IOC和AOP机制实现各组件的关联.从而实现了低耦合调用.增强了系统可维护性和扩展性。

Spring是整个项目的核心。

**2.2.2 springMVC**

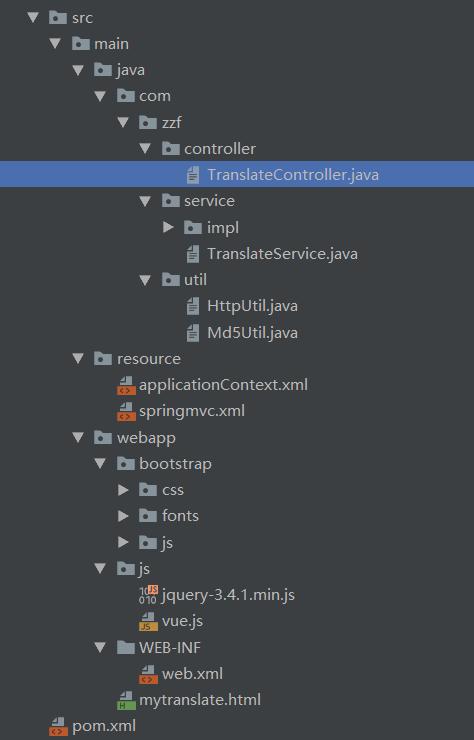
SpringMVC框架提供了一整套完善的组件，是以强大的Spring容器为基础的框架，框架的配置简单又不失灵活性，代码的可重用性很高，可扩展性好。

使用途径：

项目的SpringMVC框架和Spring容器一起使用，两个框架一起使用大大增加了开发效率。

# 3.系统实现

## 3.1项目结构

****

## 3.2配置文件

在项目下main目录下新建，名为resources，存放配置文件

### 3.2.1applicationContext.xml文件

自动扫描web包 ,将带有注解的类 纳入spring容器管理。

<context:component-scan base-package="com.zzf">

<context:exclude-filter type="annotation"

expression="org.springframework.stereotype.Controller"/>

</context:component-scan>

### 3.2.2 springmvc.xml文件

开启注解扫描，自动扫描该包，使SpringMVC认为包下用了@controller注解的类是控制器，只扫描控制器。

<!--开启注解扫描-->

<context:component-scan base-package="com.zzf">

<!--只扫描控制器-->

<context:include-filter type="annotation" expression="org.springframework.stereotype.Controller"/>

</context:component-scan>

<mvc:default-servlet-handler/>

<mvc:annotation-driven/>

## 3.3 controller类TranslateController.java

描述该文件作用，并对里面的关键代码进行解释

package com.zzf.controller;

import com.alibaba.fastjson.JSONObject;

import com.zzf.service.TranslateService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

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

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

/\*\*

\* 翻译控制层

\*/

@Controller

public class TranslateController {

/\*\*

\* Spring SpringMVC

\*/

@Autowired

private TranslateService translateService;

@RequestMapping("/translate")

@ResponseBody

public JSONObject translate(String text){

JSONObject jsonObject = translateService.translate(text);

return jsonObject;

}

}

## 3.4 service服务层

### 3.4.1 service服务层impl文件夹下TranslateServiceImpl.java

描述该文件作用，并对里面的关键代码进行解释

package com.zzf.service.impl;

import com.alibaba.fastjson.JSONObject;

import com.zzf.service.TranslateService;

import com.zzf.util.HttpUtil;

import com.zzf.util.Md5Util;

import org.springframework.stereotype.Service;

/\*\*

\* https://api.fanyi.baidu.com/doc/21

\* APP ID：20201213000646509

密钥：12ftBew9d9KRIDhFynnZ

\*/

@Service

public class TranslateServiceImpl implements TranslateService {

@Override

public JSONObject translate(String text) {

// 生成签名

String sign = Md5Util.getSign(text);

String url = "http://api.fanyi.baidu.com/api/trans/vip/translate?q=" + text + "&from=zh&to=en&appid=20201213000646509&salt=1435660288&sign=" + sign;

JSONObject jsonObject = HttpUtil.sendGet(url);

return jsonObject;

}

}

### 3.4.2 service服务层 TranslateService.java

描述该文件作用，并对里面的关键代码进行解释

package com.zzf.service;

import com.alibaba.fastjson.JSONObject;

/\*\*

\* 翻译service

\*/

public interface TranslateService {

JSONObject translate(String text);

}

## 3.5工具包Util

### 3.5.1 HttpUtil.java

描述该文件作用，并对里面的关键代码进行解释

package com.zzf.util;

import com.alibaba.fastjson.JSON;

import com.alibaba.fastjson.JSONObject;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

import java.net.URLConnection;

import java.util.List;

import java.util.Map;

public class HttpUtil {

public static JSONObject sendGet(String url) {

String result = "";

BufferedReader in = null;

JSONObject jsonObject = null;

try {

URL realUrl = new URL(url);

// 打开和URL之间的连接

URLConnection connection = realUrl.openConnection();

// 设置通用的请求属性

connection.setRequestProperty("accept", "\*/\*");

connection.setRequestProperty("connection", "Keep-Alive");

connection.setRequestProperty("Content-Type",

"application/x-www-form-urlencoded; charset=UTF-8");

// 建立实际的连接

connection.connect();

// 获取所有响应头字段

Map<String, List<String>> map = connection.getHeaderFields();

// 遍历所有的响应头字段

for (String key : map.keySet()) {

System.out.println(key + "--->" + map.get(key));

}

// 定义 BufferedReader输入流来读取URL的响应

in = new BufferedReader(new InputStreamReader(

connection.getInputStream(), "UTF-8"));

String line;

while ((line = in.readLine()) != null) {

line = decodeUnicode(line);

result += line;

// 将数据转化为json对象

jsonObject = JSON.parseObject(line);

System.out.println(1);

}

} catch (Exception e) {

System.out.println("发送GET请求出现异常！" + e);

e.printStackTrace();

}

// 使用finally块来关闭输入流

finally {

try {

if (in != null) {

in.close();

}

} catch (Exception e2) {

e2.printStackTrace();

}

}

return jsonObject;

}

/\*\*

\* 字符串含有Unicode编码转化为正常字符串

\* @param theString

\* @return

\*/

public static String decodeUnicode(String theString) {

char aChar;

int len = theString.length();

StringBuffer outBuffer = new StringBuffer(len);

for (int x = 0; x < len;) {

aChar = theString.charAt(x++);

if (aChar == '\\') {

aChar = theString.charAt(x++);

if (aChar == 'u') {

// Read the xxxx

int value = 0;

for (int i = 0; i < 4; i++) {

aChar = theString.charAt(x++);

switch (aChar) {

case '0':

case '1':

case '2':

case '3':

case '4':

case '5':

case '6':

case '7':

case '8':

case '9':

value = (value << 4) + aChar - '0';

break;

case 'a':

case 'b':

case 'c':

case 'd':

case 'e':

case 'f':

value = (value << 4) + 10 + aChar - 'a';

break;

case 'A':

case 'B':

case 'C':

case 'D':

case 'E':

case 'F':

value = (value << 4) + 10 + aChar - 'A';

break;

default:

throw new IllegalArgumentException(

"Malformed \\uxxxx encoding.");

}

}

outBuffer.append((char) value);

} else {

if (aChar == 't')

aChar = '\t';

else if (aChar == 'r')

aChar = '\r';

else if (aChar == 'n')

aChar = '\n';

else if (aChar == 'f')

aChar = '\f';

outBuffer.append(aChar);

}

} else

outBuffer.append(aChar);

}

return outBuffer.toString();

}

}

### 3.5.2 Md5Util.java

描述该文件作用，并对里面的关键代码进行解释

package com.zzf.util;

import org.apache.commons.codec.digest.DigestUtils;

public class Md5Util {

/\*\*

\* 生成签名 appid+q+salt+密钥 使用MD5 32位小写加密

\* @param q

\* @return

\*/

public static String getSign(String q){

String text = "20201213000646509" + q + "1435660288" + "12ftBew9d9KRIDhFynnZ";

String sign = DigestUtils.md5Hex(text);

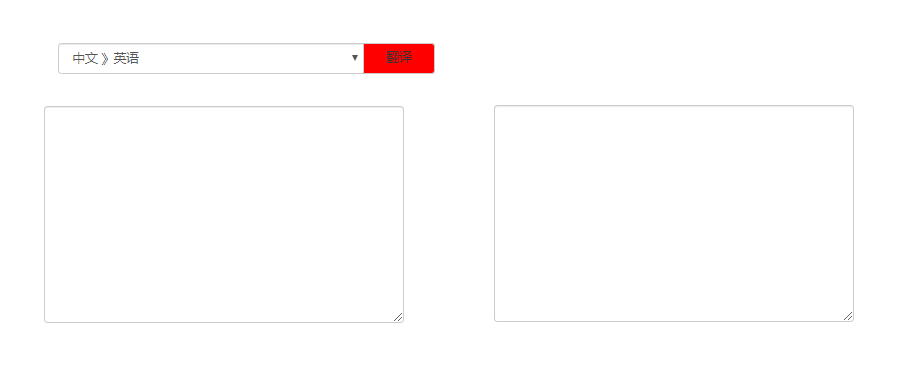
return sign;

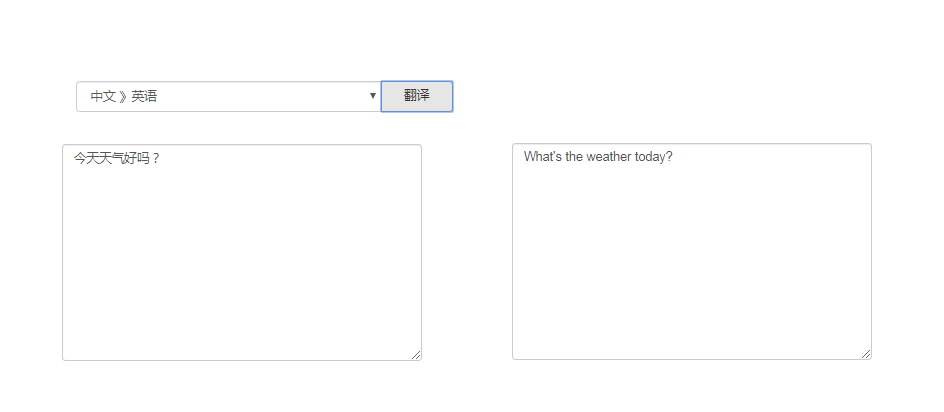
}

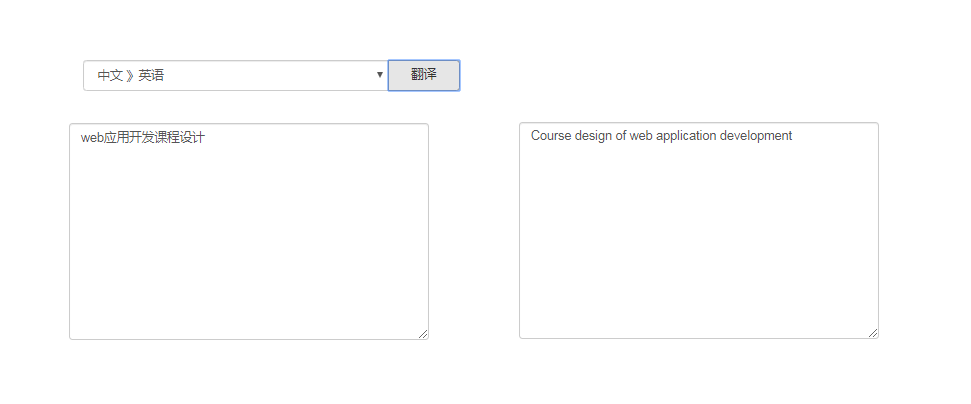
}

# 4.系统测试

## 4.1 运行结果展示

****

****

****

# 5.技术难点

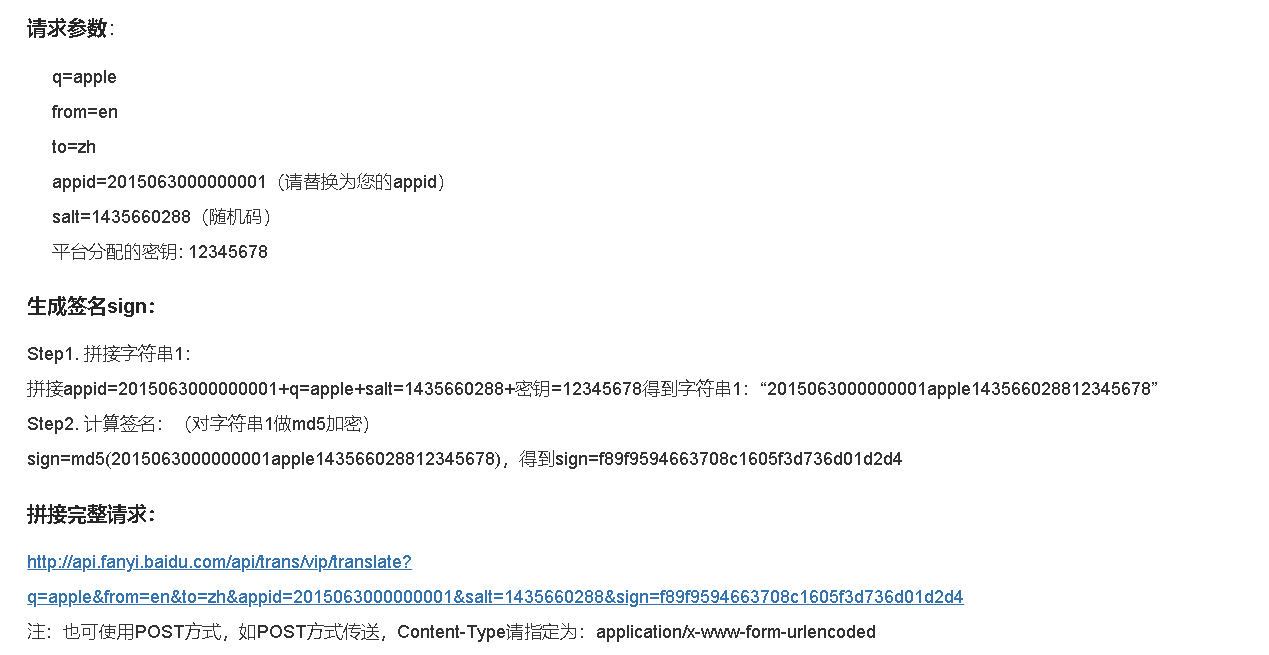
## 5.1 获得百度账号的ID和密钥

****

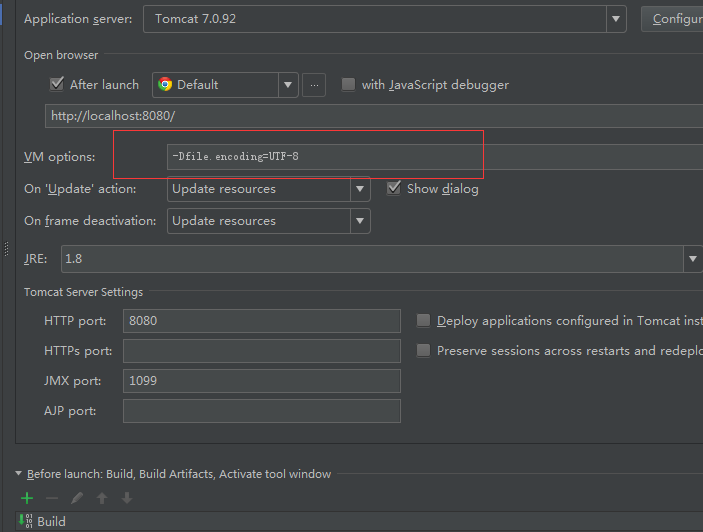
## 5.2 获得英文和中文的翻译代码

****

## 5.3 生成签名和拼接完整请求连接

****

## 5.4 TomCat服务器的编码修改

****

# 6.心得总结

　　开发一个项目，最重要的是细心，并不是一定要做到面面俱到，但也要充分考虑到客户的需求和现实意义，不管什么系统，只用运用到实际应用中，才具　　有现实意义。所以在这次课程设计，我收获的不仅仅是课程上的知识得到实际应用，还有编程的基本习惯和开发系统时应注意的流程。在实际设计中才发现，书本上理论性的东西与在实际运用中的还是有一定的出入的，所以有些问题不但要深入地理解，而且要不断地更正以前的错误思维。一切问题必须要靠自己一点一滴的解决，而在解决的过程当中你会发现自己在飞速的提升。对于在线翻译系统，其程序是比较简单的，但是相对简单是基于对API相对熟练使用的基础上，主要是解决程序设计中的问题，而程序设计是一个很灵活的东西，它反映了你解决问题的逻辑思维和创新能力，它才是一个设计的灵魂所在。