**武汉纺织大学数计学院**

**Web应用开发课程设计报告**

**基于SSM与百度API接口的翻译器**

**成 绩：**

**学 号： 1804280404**

**姓 名： 郑洋**

**班 级： 物联网11804**

**指导教师： 聂刚**

**完成日期： 2020 年 12月18日**

**目录**

1. 需求分析
2. 设计思路

**网页模块**

**后端模块**

**翻译控制器（TranslateController）**

**参数列表（RequestParam）**

**翻译服务器（TranslationService）**

**MD5Util加密工具**

**使用ssm框架**

1. 程序实现

**网页模块**

**复选框，用于判断语种内容**

**翻译框，主要实现工具**

**输入框，提供主要参数**

**输出框，显示结果**

**后端模块**

**创建一个翻译控制器（TranslateController）用来获取前段数据**

**参数列表（RequestParam）用于获取API接口语言参数**

**翻译服务器（TranslationService）用于连接API接口并调用数据**

**MD5Util用于对API接口的签证进行加密**

**使用ssm框架前端和后端连接起来**

Spring配置

1. 测试结果
2. 总结

**一.需求分析**

设计一个翻译网页，为复选框做一个选择判断，将两个字符串和api接口绑定，确保内容与**常见语种列表相符合。在通过翻译按钮将输入框中的内容传到后台进行校对，接着将生成的结果返回前端的文本框。**

**bootstrap.css文件和jQUery.Js文件**

**Springmvc文件**

****二.设计思路****

**网页模块**

**复选框，用于判断语种内容**

**翻译框，主要实现工具**

**输入框，提供主要参数**

**输出框，显示结果**

**后端模块**

**创建一个翻译控制器（TranslateController）用来获取前段数据**

**参数列表（RequestParam）用于获取API接口语言参数**

**翻译服务器（TranslationService）用于连接API接口并调用数据**

**MD5Util用于对API接口的签证进行加密**

**使用ssm框架前端和后端连接起来**

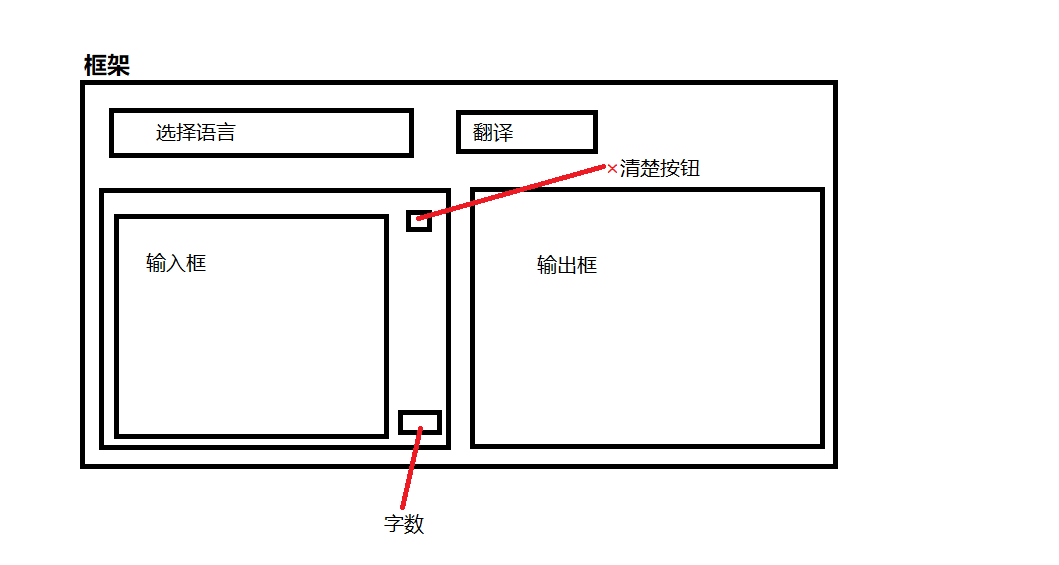
**三.程序实现**

****网页模块****

**设计网页之前，我们可以先导入平时需要用到的工具包，这里我用到了bootstrap.css样式文件和jQUery.Js文件，以方便我们的书写。**







如图所示：我们需要一个大的盒子在里面套很多下的盒子用来用来实现我们的网页。

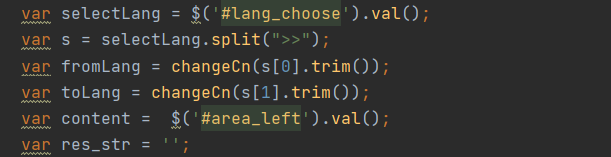
首先需要设计最外面的盒子，然后将盒子分为两行，第一行上面画选择语言的模块和翻译模块，下面画输入模块和输出模块。这里我们可以用到bootstrap。

1. 选择语言模块

先写定义一个复选框，在复选框中输入我们需要的内容如（中文 >> 英文） 保证能够下拉。



通过select-option 创建下拉框，在给相应字符串进行切片，分别存放到不同的String字符里面，以便于校对。#lang\_choose{height:40px;}可设置复选框的高度。



使用jQuery语法选中复选框（id='lang-choose'）：$('#lang\_choose').val();选中之后我们将selectLang所获去的复选框内容进行切片，分成s[0,1]分别储存 >> 前后复选框中的信息。然后将两个书中分别存放在 formLang 与 tolang 中。

1. 翻译按钮

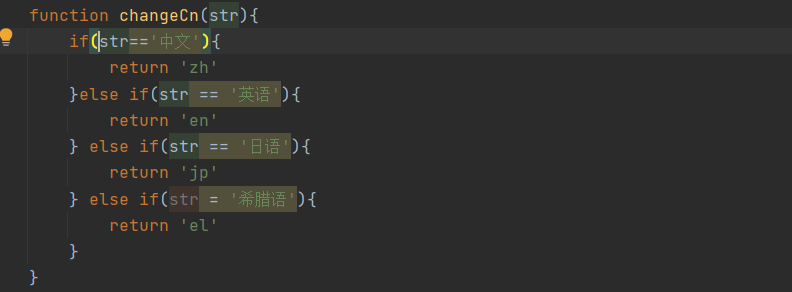
翻译付款的设计方式与语言模块类似，同样需要用到一个盒子，然后在盒子中套入一个button输入文字提示，就完成了一个翻译按钮了。



我们需要对翻译按钮做一个点击事件，用来校对复选框中的信息和传输输入框中的内容并让输出框得到返回的类容。



先用jQUery的方式选中 button 按钮的id=’translationButton’，这样我们就可以对button按钮做事件了。第二步：为按钮设置一个点击事件事件的内容是把复选空中的内容分割后储存在fromLang 与 toLang 两个字符串中，并获取左侧输入框的内容用 .post()方法对三个数据进行校对。 （.post()方法用来请求服务器载入的数据），使用founction(res): 函数来对我们获取的值进行调用（founction(res): 相当于值向function函数本身，与this差不多），在通过for each对获取值进行遍历。最后用jQUery的方式选中输出框 ，为它设置一个test方法，将res\_str打印出来就行了。



使用founction(res): 所指向的判断条件，’zh’,’en’,’jp’,’el’为百度api接口语言列表中的代码，它会通过.post("/ssm\_war/getTranslationContent", { fromLang: fromLang, toLang: toLang, content: content },function())方法后台的参数进行对比。对比完成后就可以实现我们的翻译想过了。

1. 输出框模块

输入框模块需要设计 文本框

<textarea id="area\_left" class="form-control" rows="8"></textarea>

设置span用来做清除文本框中的字符信息

<div class="col-md-1">

<div id="area-right">

<span id="clearInput" class="glyphicon glyphicon-remove" aria-hidden="true"></span>

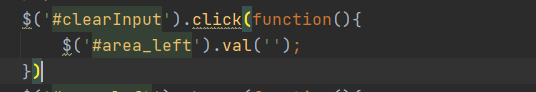
</div>

做一个前端数值信息用来统计文本框的中字符个数。

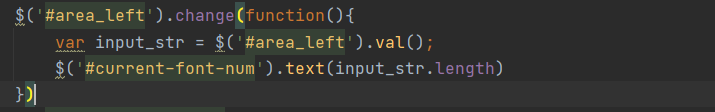
<div id="font-num"><span id="current-font-num">0</span>/5000</div>

</div>

输入信息后我们可以通过 span id=’clearInput’ 为clearInput设置一个点击事情，用来覆盖文本框中的字符



统计数字的信息后我们同样可以通过 span id=’current-font-num’来改变数值的个数，并把相对应的信息打印在span 区域中。这样我们就得到统计的信息了。



1. 输出框模块

主要用于返回.post()函数中 所执行的内容

<div id="input\_area" class="col-md-5">  
 <textarea id="tr\_result" class="form-control" rows="8"></textarea>  
</div>

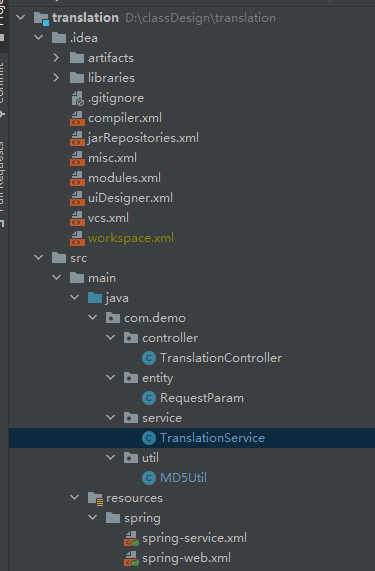
网页模块的整体样式：css

<style>  
<!-- div{-->  
<!-- border: 1px solid red;-->  
<!-- }-->  
 .container{  
 margin-top: 100px;  
 }  
 #translationButton{  
 border-radius: 3px;  
 color: #fff;  
 width:100px;  
 height:40px;  
 }  
 #input\_area{  
 background: #f2f2f2;  
 height: 200px;  
 border-radius: 5px;  
 margin-top: 20px;  
 margin-right: 20px;  
 padding-top: 10px;  
 }  
 #header{  
 min-height: 40px;  
 }  
 #lang\_choose{  
 height:40px;  
 }  
 #area\_left{  
 background: #f2f2f2;  
 border: none;  
 box-shadow: none;  
 }  
 #tr\_result{  
 background: #f2f2f2;  
 border: none;  
 box-shadow: none;  
 }  
 #clearInput:hover{  
 cursor: pointer;  
 background: #f2f2f2;  
 }  
 #area-right{  
 width: 30px;  
 padding: 0;  
 }  
 #font-num{  
 padding-top: 148px;  
 width: 35px;  
 color: gray;  
 font-size: 8px;  
 display: block;  
 float: right;  
 height: 170px;  
 }  
</style>

**后端模块**

后端主要涉及了4个类和2个spring的映射

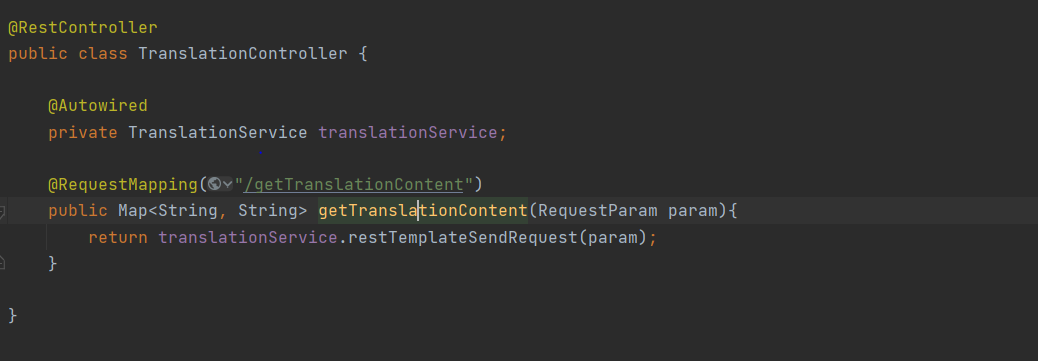
结构列表：



1. **翻译控制器（TranslateController）**

**翻译控制器类的作用是将从网页中获取的数据传给翻译服务器（TranslationService）类。获取过程中，我们需要通过注解进行说明。**

RequestMapping是一个用来处理请求地址映射的注解，可用于类或方法上。用于类上，表示类中的所有响应请求的方法都是以该地址作为父路径。然后通过param： 指定request中必须包含某些参数值。这样就可以将获取到的值通过TranslationService 定义的translationService实例传入getTranslationContent()方法中。RequsestParam 用接受返回的信息。



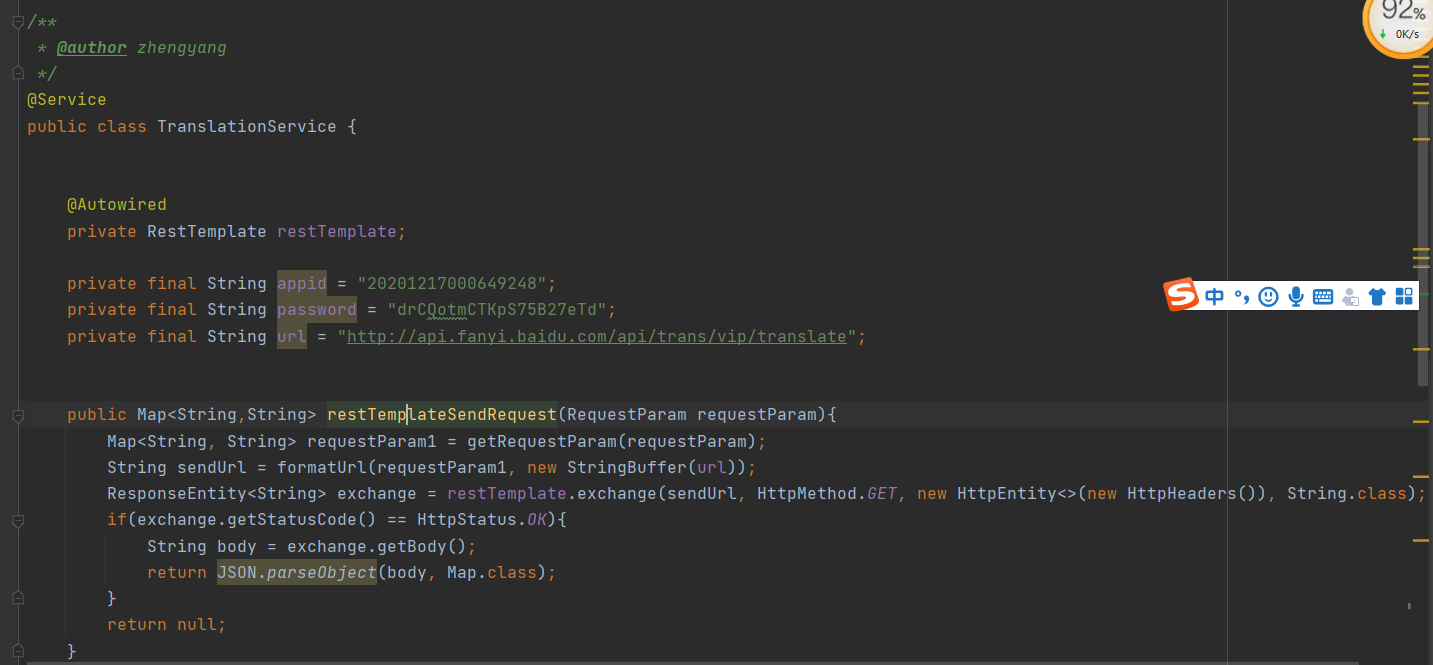
1. **参数列表（RequestParam）**



**将formLang,toLang,content一一序列化。参数列表的功能就完成了**

**RequestParam相当于一个盒子用来存放参数列表中的相关信息，方便与百度翻译API接口尽心对比。同时方便 TranslationService类和TranslationController类的调用。**

**翻译服务器（TranslationService）**



ResponseEntity<String> 调用spring框架，将web客户端的服务手机存放在exchange里面。通过if判断exchange的状态，使其返回通过JSON解析后的Map类。

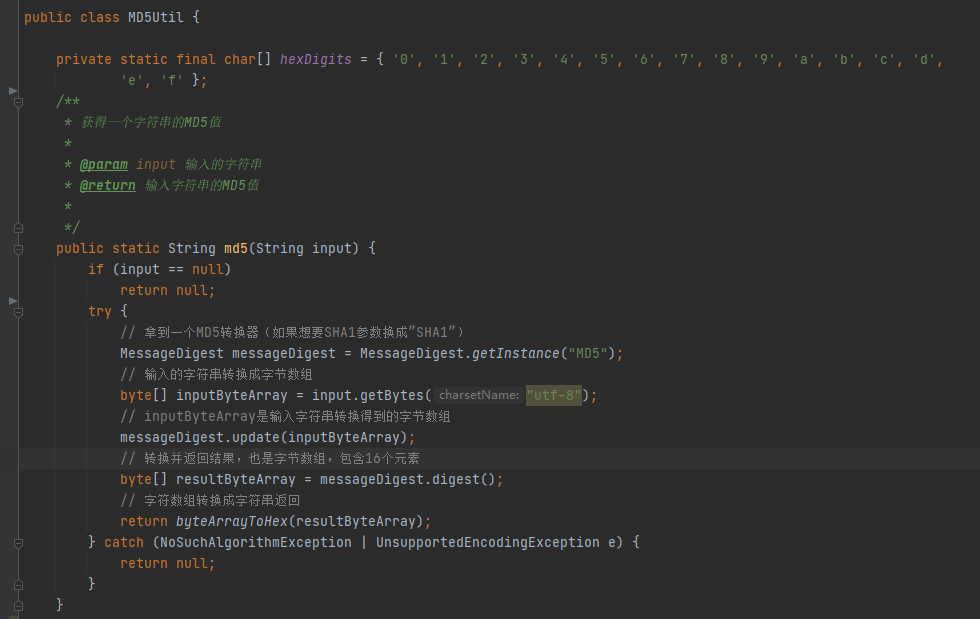


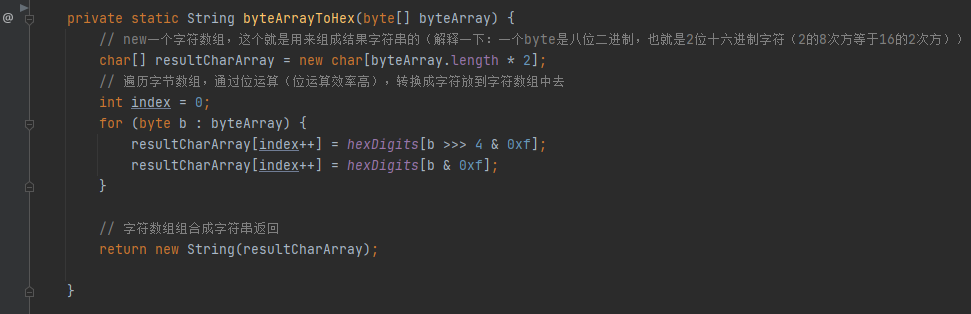
**getRequestParam()用来调用API接口，将API接口的参数输出Map<String ,string>中。**



formatUrl()用来拼接url，比如说，他接口是IMG_256www.baidu.com我们要把参数接在他后面就是IMG_256www.baidu.com?k=v&k=v 。这样才能完整的访问百度翻译的通用API接口。

1. **MD5Util加密**

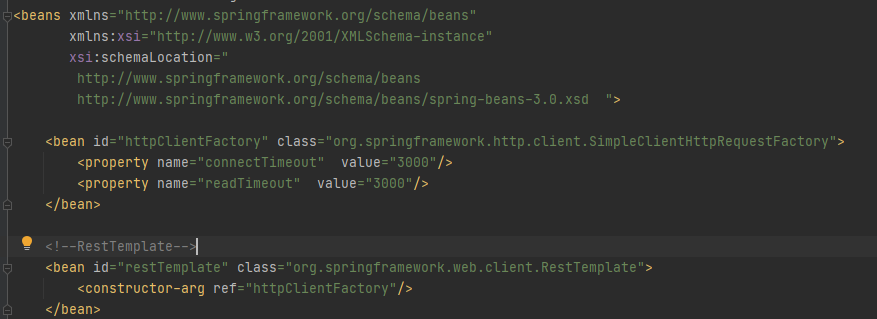




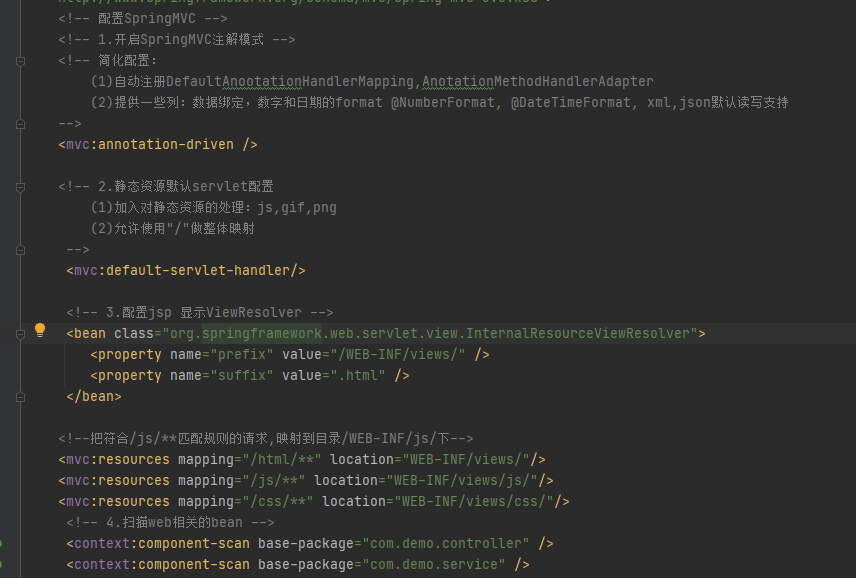
new一个字符数组，这个就是用来组成结果字符串的（解释一下：一个byte是八位二进制，也就是2位十六进制字符（2的8次方等于16的2次方））

遍历字节数组，通过位运算（位运算效率高），转换成字符放到字符数组中去

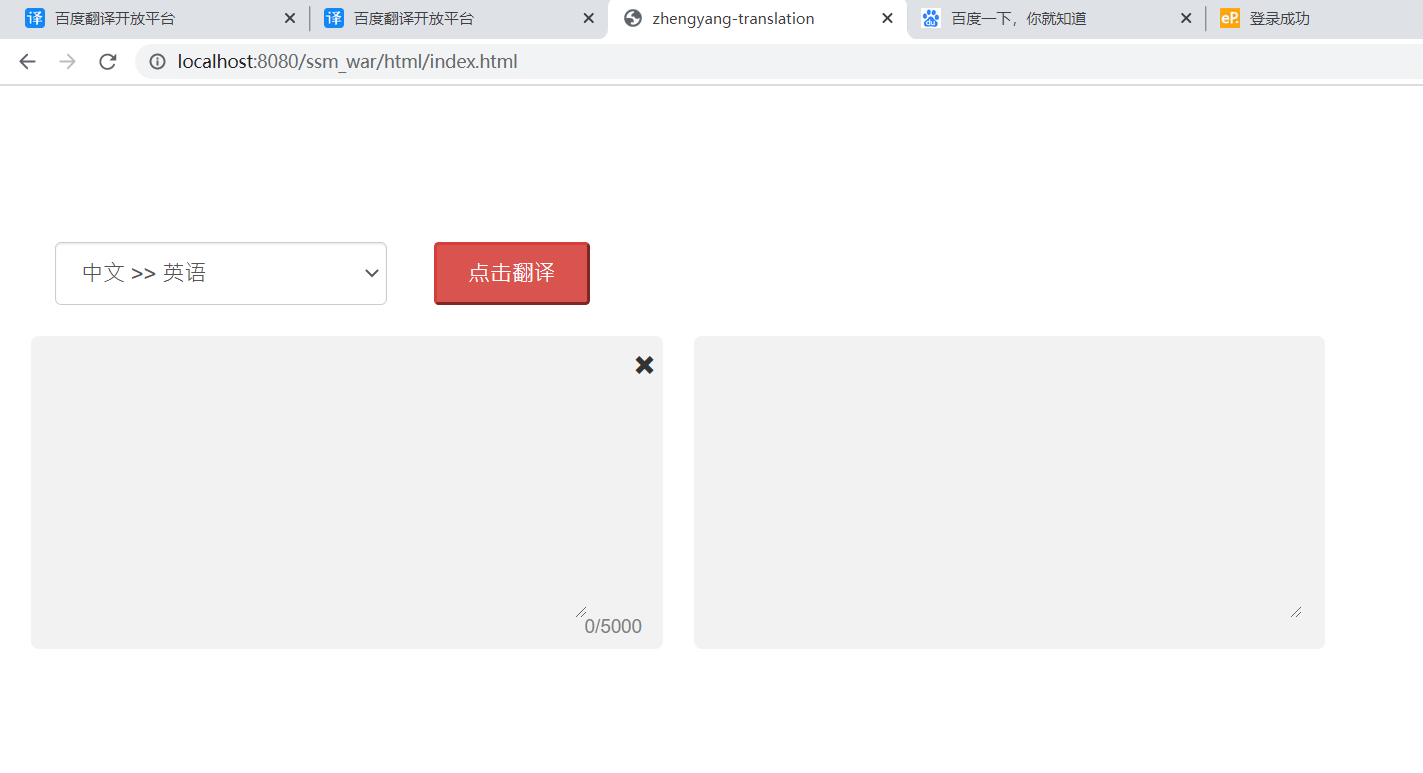
Spring-service配置信息

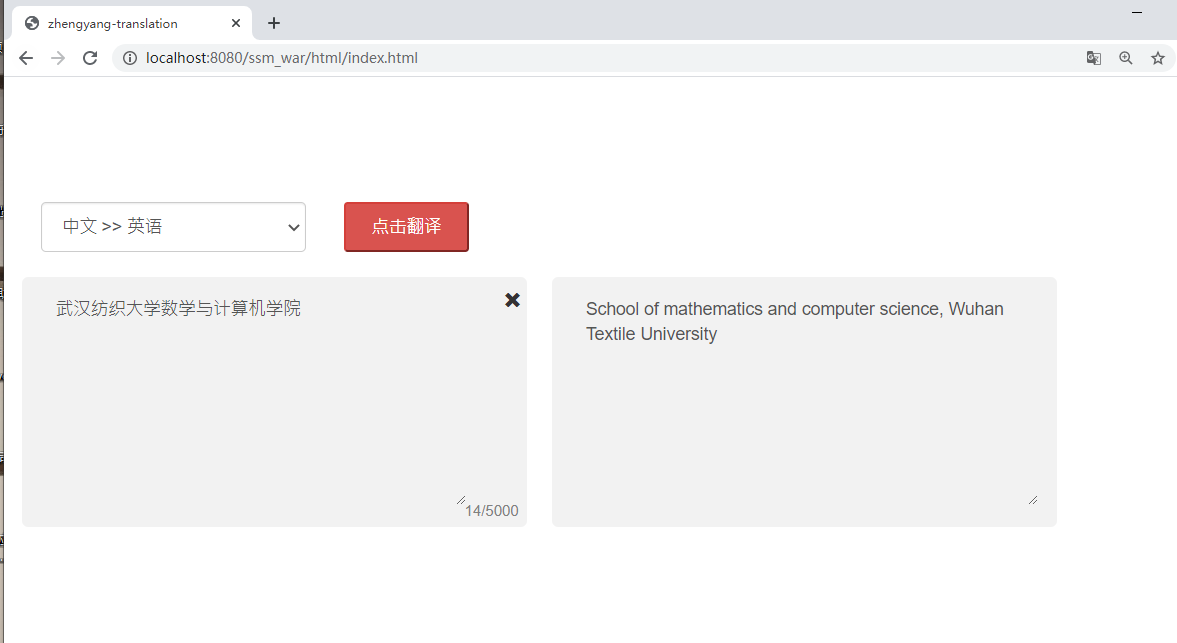


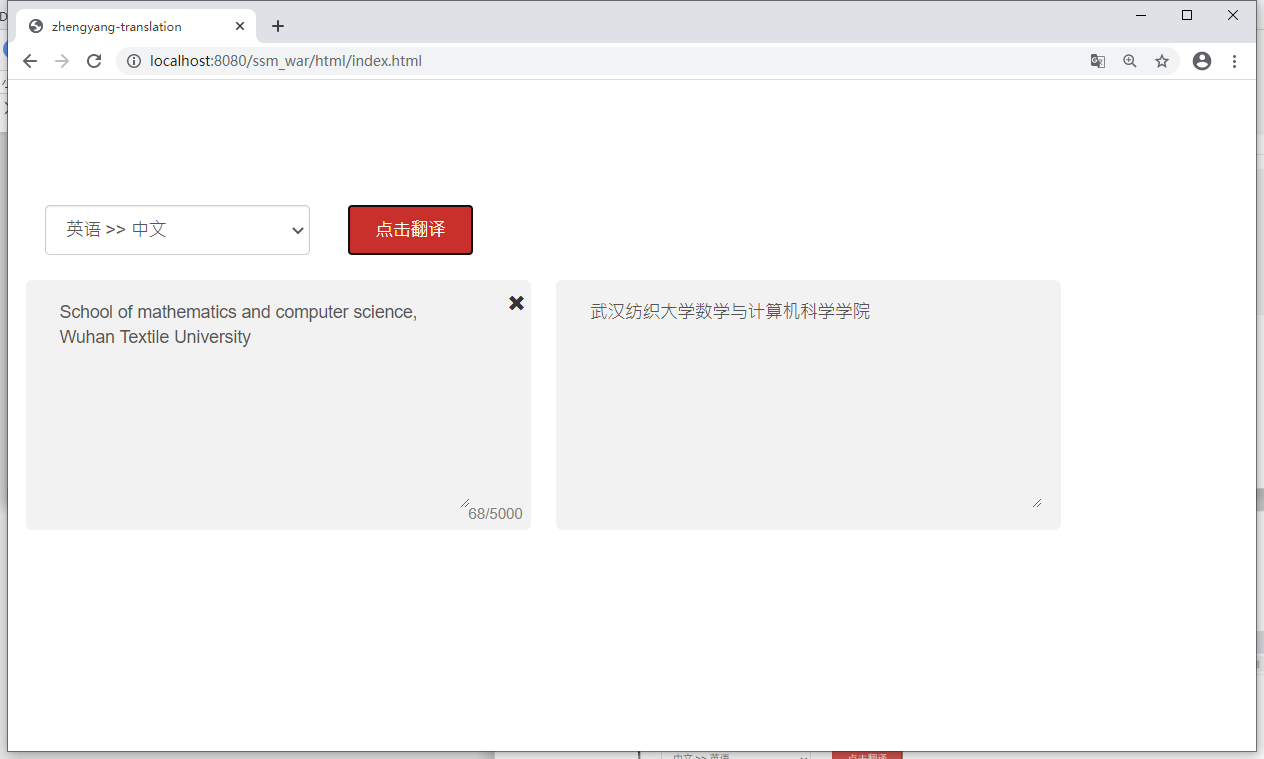
Spring-web配置信息



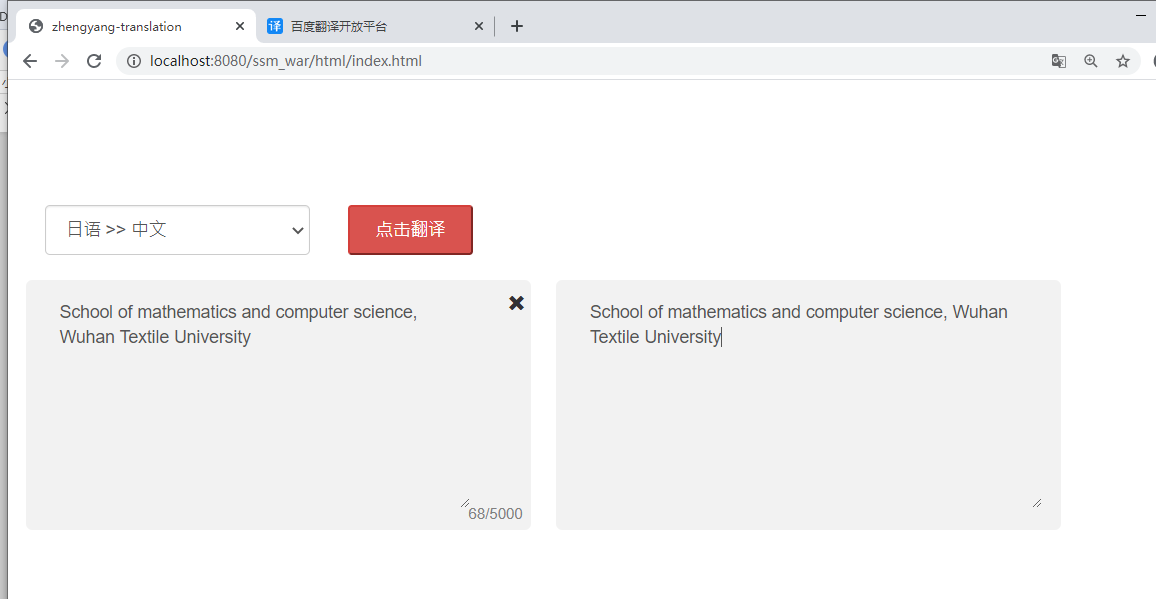
**四．效果图**



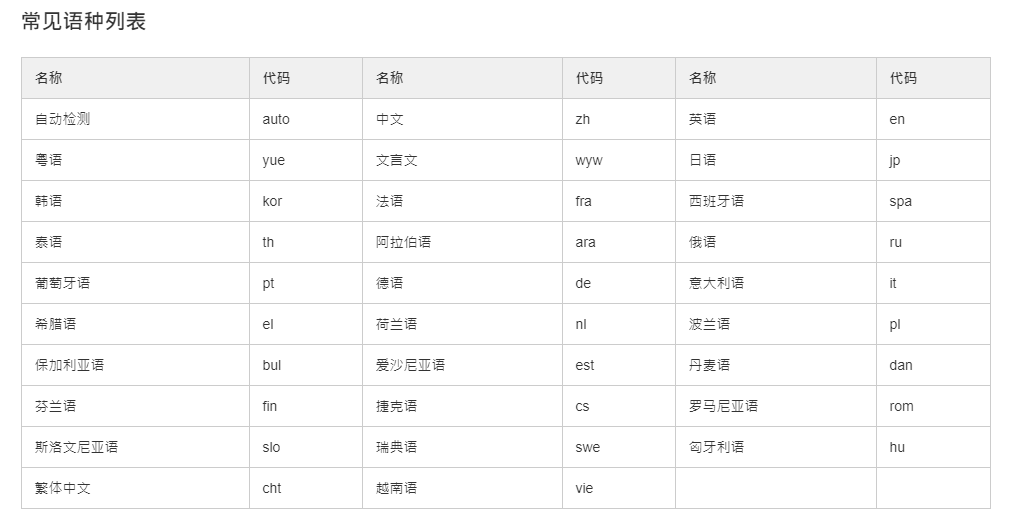




选择正确的翻译语种后，就可以将输入的内容翻译了。效果如图所示。



如果我们需要添加更多的翻译语言，我们只需通过function（res）:函数来改变对应的值就可以达成目标了；function changeCN（）{}中也需要返回相应的语言代码，下图为百度通用翻译API接口的语言代码：



**五．总结**

翻译项目需要通过前后端数据的互相访问以及api接口的基本调用来完成，很大的考验了java使用的综合能力。项目的困难点有很多，比如设计清楚时很难想到是用空字符串进行覆盖。连接API接口是的签名问题是符合进行加密的。还有就是Spring的配置，MD5加密。这些都是很生疏的内容。在加上上课听的也似懂非懂。所以在做项目上花费了大量的时间。

这次项目也让我感到了java的有趣，因为在写代码的过程中，我明白了一个网页翻译可以通过什么方式来实现，联想开来，我可以通过很多种方式去实现很多种项目。不过这也让我看到了自己的不足。上课没有认真学习，而且缺乏想象。我会进一步改善自己的不足。在java上打开自己新的领域。