类Translate.java

定义方法MSG，传递参数Word。在方法MSG中，定义接口的请求地址，用集合Map定义变量设置请求参数，用变量调用put方法用键值对的方式传值传入到Map集合中。初始化两个变量用来获取数据。

|  |
| --- |
| **public static** String MSG(String word) {  *//请求地址* String Url = **"http://www.webxml.com.cn/WebServices/TranslatorWebService.asmx**  **/getEnCnTwoWayTranslator"**;  *//请求参数* Map map = **new** HashMap();  map.put(**"Word"**, word);  *//获取数据* String result = **""**;  BufferedReader read = **null**;  **try** {  URL realurl = **new** URL(Url + **"?"** + *urlencode*(map));  URLConnection connection = realurl.openConnection();  connection.connect();  read = **new** BufferedReader(**new** InputStreamReader(  connection.getInputStream(), **"UTF-8"**));  String line;  **while** ((line = read.readLine()) != **null**) {  result += line;  }  } **catch** (IOException e) {  e.printStackTrace();  } **finally** {  **if** (read != **null**) {  **try** {  read.close();  } **catch** (IOException e) {  e.printStackTrace();  }  }  }  **return** result;  } |

方法urlencode主要就是将从接口中存入map集合的键值对，通过增强for循环的方式来转换成字符串数据。

|  |
| --- |
| **public static** String urlencode(Map<String, Object> data) {  *//将map里的参数变成像 showapi\_appid=###&showapi\_sign=###&的样子* StringBuilder sb = **new** StringBuilder();  **for** (Map.Entry i : data.entrySet()) {  **try** {  sb.append(i.getKey()).append(**"="**).append(URLEncoder.*encode*(i.getValue() + **""**, **"UTF-8"**));  } **catch** (UnsupportedEncodingException e) {  e.printStackTrace();  }  }  **return** sb.toString();  } |

方法DealXML主要就是将字符串转换成xml，用这个方法来解析获取到的数据。

|  |
| --- |
| **public static** Map DealXml(String xml) {  Map map = **new** HashMap();  Document doc = **null**;  **try** {  *// 将字符串转为XML* doc = DocumentHelper.*parseText*(xml);  *// 获取根节点* Element rootElt = doc.getRootElement();  System.***out***.println(rootElt);  *// 获取根节点下的子节点string* Iterator iter = rootElt.elementIterator(**"string"**);  System.***out***.println(iter);  *// 遍历string节点* **int** i = 0;  **while** (iter.hasNext()) {  Element node = (Element) iter.next();  **if** (i == 0) {  String word = node.getStringValue().split(**":"**)[0];  map.put(**"Input\_word"**, word);  } **else** {  map.put(**"Translation\_word"**, node.getStringValue());  }  i++;  }  } **catch** (DocumentException e) {  e.printStackTrace();  } **catch** (Exception e) {  e.printStackTrace();  }  **return** map;} |

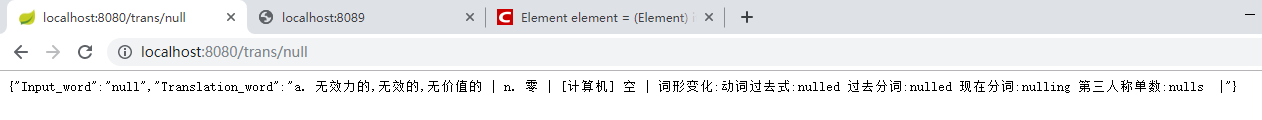
Main方法主要就是调用之前的MSG方法和Deal方法。

|  |
| --- |
| **public class** Translate {  **public static void** main(String[] args) {  String data = *MSG*(**"animal"**);  System.***out***.println(data);  Map o =Translate.*DealXml*(data);  System.***out***.println(o);  } |

类TranslatorController.java

TranslatorController主要就是给出一个url的连接，让后端的数据可以和前端进行一个交互，将后端的数据以json的格式显示在网页上。

|  |
| --- |
| **package** translate;  **import** org.springframework.web.bind.annotation.\*;  **import** java.util.HashMap; **import** java.util.Map;  @CrossOrigin @RestController **public class** TranslatorController {  @ResponseBody  @RequestMapping(**"/test"**)  **public** String test(){  **return "test success,can use"**;  }   @GetMapping(**"/trans/{word}"**) *//给url的连接* **public** Map<String,Object> trans(@PathVariable(**"word"**) String word){  Translate transutil = **new** Translate();  Map shuju = **new** HashMap();  String even=transutil.*MSG*(word);  shuju =Translate.*DealXml*(even);  **return** shuju;  } } |



前端

这个相当是主页面，Trans中的时间全部放在此页面中



Trans中主要是写后端获取到的数据与前端的事件进行绑定。

|  |
| --- |
| import React, { Component } from "react";  import axios from "axios";  import { Button, Input } from "antd";  class **Trans** extends **Component** {  constructor(props) {  super(props);  this.state = {  trans: ""  };  this.translations = this.translations.**bind**(this);  }  **translations**() {  *//将插入的值插入到"Input\_word"中*  var word = document.**getElementById**("Input\_word").value;  axios.**get**("http://localhost:8080/trans/" + word).**then**(response => {  **console**.**log**(response);  this.**setState**({  trans: response.data.Translation\_word.**replace**(/\|/g, "\n")  });  });  }  **render**() {  return (  <div *style*={{marginLeft:"500px"}}>  <**Input** *id*="Input\_word" *style*={{margin: "10px",width: "300px"}}/>  <**Button** *size*="large" *onClick*={this.translations} *style*={{background:"blue",color:"white"}}>  中英翻译  </**Button**>  <br/>  <**Input.TextArea**  *value*={this.state.trans}  *style*={{  height: "120px",  width:"530px"  }}  />  </div>  );  }  }  export default Trans; |

