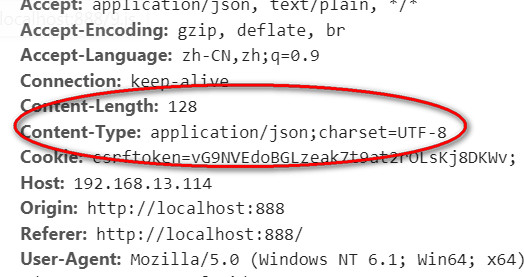
# Axios.post传输参数到后台方法总结

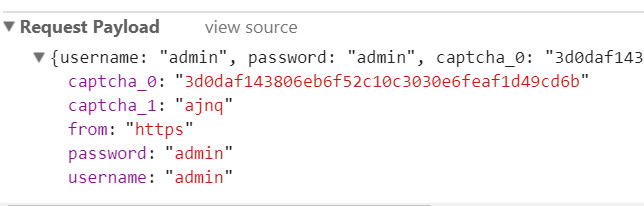
**1、使用axios发送post请求，默认以JSON(字符串)数据传输到后台**

**2、使用axios发送post请求，将JSON数据改为为form类型**

通常前端通过POST请求向服务器端提交数据格式有4中，分别是"application/x-www-form-urlencoded"格式、" multipart/form-data"格式、"application/json"格式和"text/xml"格式。通常最常见的是"application/json"格式，也就是通过JSON字符串形式。



在控制台看到的数据就是这样：



 有的时候后台需要Form形式的数据才能正常解析，所以前端需要在传过去时修改一下格式。开始我以为是修改Content-Tyoe属性，于是就

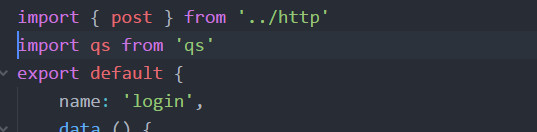
axios.defaults.headers = {

'Content-type': 'application/x-www-form-urlencoded'

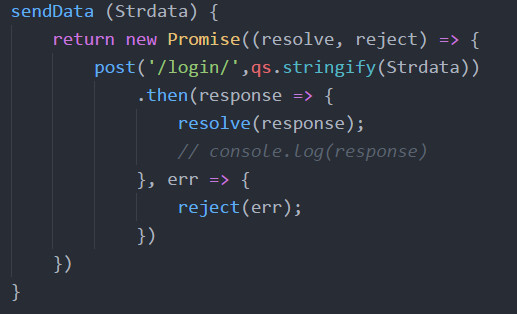
}

设置之后看到请求时的Content-type变成了application/x-www-form格式，但是数据没变。之后找了很久终于找到真正的方法。

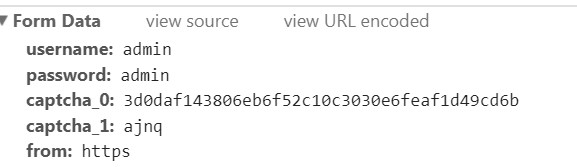
**第一步：引入axios ps**



**第二步：使用qs 转换（即qs.stringify()）**



 之后看一下请求数据格式就变成了这样：



**项目实例:**

**前端：**

//在对应的xxx.vue设置，要引入app.js

*//获取商铺信息列表*getStoreInfos() {  
 **let** para = **this**.**$qs**.stringify({  
 **page**: **this**.**currentPage**,  
 **rows**:**this**.**pagesize**,  
 **name**: **this**.**filters**.**name** });  
  
 *getStoreListPage*(para).then(res => {  
 **this**.**total** = res.**data**.**total**;  
 **this**.**storeInfos** = res.**data**.**storeInfos**;  
 *//从数据库获取的图片路径* ***console***.log(**"数据库获取图片路径"**+**this**.**storeInfos**);  
 });  
},

**在api.js设置**

**export const** *getStoreListPage* = params => {**return** axios.post(**`**${base}**/store/listPage`**,params);};

后台：@RequestParam接受参数

@RequestMapping(value="/listPage",method=RequestMethod.***POST***)

**public** **void** listPage(@RequestParam(value="rows",required=**true**) String rows,

@RequestParam(value="page",required=**true**) String page,HttpServletResponse response) **throws** Exception {

//获取前端传过来参数

*log*.info("页数"+page +"条数"+rows);

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

**if**(page != **null** && rows !=**null**) {

PageBean pageBean = **new** PageBean(Integer.*parseInt*(page),Integer.*parseInt*(rows));

map.put("start", pageBean.getStart());

map.put("page",pageBean.getPageSize());

}

JSONObject jsonStoreInfo = **new** JSONObject();

List<StoreInfo> listStoreInfo = **new** ArrayList<StoreInfo>();

listStoreInfo = storeInfoServiceImpl.findStoreInfo(map);

jsonStoreInfo.put("storeInfos", listStoreInfo);

ResponseUtil.*writeHtml*(response, jsonStoreInfo);

*log*.info("请求结束");

}

**3、使用axios发送post请求，将JSON数据传输给后台**

此时的Content-Tyoe属性，于是就

axios.defaults.headers = {

'Content-type': ' application/json;charset=UTF-8'

}

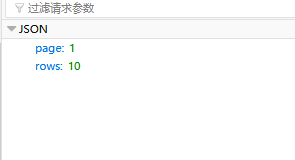
设置之后看到请求时的Content-type变成了application/json;charset=UTF-8格式，但是数据没变。之后找了很久终于找到真正的方法。

传输json格式就没必要引入axios qs:

*//获取商铺信息列表*getStoreInfos() {  
 **let** para = {  
 **'page'**: **this**.**currentPage**,  
 **'rows'**:**this**.**pagesize**,  
 };  
  
 *getStoreListPage*(para).then(res => {  
 **this**.**total** = res.**data**.**total**;  
 **this**.**storeInfos** = res.**data**.**storeInfos**;  
 *//从数据库获取的图片路径* ***console***.log(**"数据库获取图片路径"**+**this**.**storeInfos**);  
 });  
},

**export const** *getStoreListPage* = params => {**return** axios.post(**`**${base}**/store/listPage`**,params);};

之后看一下请求数据格式就变成了这样：



**后台是springboot，利用@RequestBody接受前端传输的json数据（其实@RequestBody接收的就是一个json字符串然后进行转换获取所需的字段值,@RequestBody对应的方法一定是POST，不能是GET）：**

//@RequestBody直接以String接收前端传过来的json数据(即如上图所示:{“page”:1,”rows”:10})

//后端对应的Controller:

@RequestMapping(value="/listPage",method=RequestMethod.***POST***)

**public** **void** listPage(@RequestBody String listObject,HttpServletResponse response) **throws** Exception {

//获取前端传过来参数

JSONObject jsonObject = JSONObject.*parseObject*(listObject);

// 获取到key为page的值

String page = jsonObject.getString("page");

String rows = jsonObject.getString("rows"); *log*.info("页数"+page +"条数"+rows);

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

**if**(page != **null** && rows !=**null**) {

PageBean pageBean = **new** PageBean(Integer.*parseInt*(page),Integer.*parseInt*(rows));

map.put("start", pageBean.getStart());

map.put("page",pageBean.getPageSize());

}

JSONObject jsonStoreInfo = **new** JSONObject();

List<StoreInfo> listStoreInfo = **new** ArrayList<StoreInfo>();

listStoreInfo = storeInfoServiceImpl.findStoreInfo(map);

jsonStoreInfo.put("storeInfos", listStoreInfo);

ResponseUtil.*writeHtml*(response, jsonStoreInfo);

*log*.info("请求结束");

}

综上所所述，个人更倾向于前端使用qs.stringify把数据转换为form格式，后台用@RequestBody接受参数。因为这样后台不用在对json格式进行处理。也要过根据前端的处理方式，反正前端和后台人员定好那种开发模式。