

ajax

一种页面局部更新的技术。

1. 异步请求，javascript代码，并和xml有关联
2. 核心对象是：XMLHttpRequest 对象是一个javascript对象，存在着浏览器差异，简称xhr对象

1. ajax的编程步骤

A. 创建xhr核心对象

- 浏览器内核主要有两种：webkit / edge
- window.XMLHttpRequest true(webkit) | false(edge)
- 创建对象

Java

```
1 var xhr;
2 if (window.XMLHttpRequest) {
3     xhr = new XMLHttpRequest();
4 }else {
5     xhr = new ActiveXObject('Microsoft.XMLHTTP');
6 }
```

B. 发送请求，传递参数

Java

```
1 xhr.open("GET", "URL?username=zhangsan"); // 设置请求方式和参数，请求路径，还没发送
2 xhr.send();
```

C. 处理响应，更新(渲染)局部页面

ajax请求的后台响应返回值必须是null!!!! 一定不能有返回值，响应值得是字符串

它只要字符串，你不要给我做什么跳转，通过response

类似于servlet中的ServletActionContext.getResponse();

response.getWriter().print("ok"); // 这个ok就是响应回去的字符串

那我的ajax怎么获取这个字符串呢？

xhr.readyState 为0时，xhr还没有进行初始化，1为xhr创建，2为发送请求，3为响应回来但不完整，4完整响应！但是响应也是有正确还是错误的

```
xhr.onreadystatechange = function(){
```

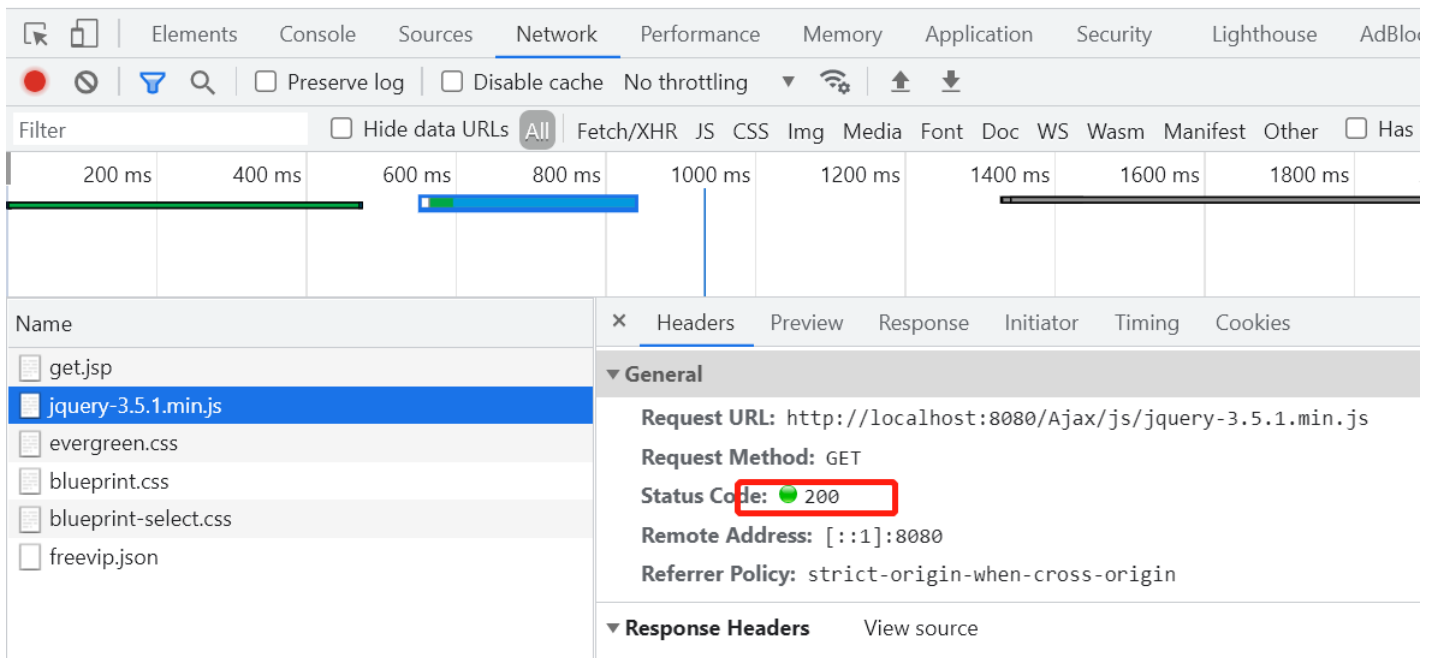
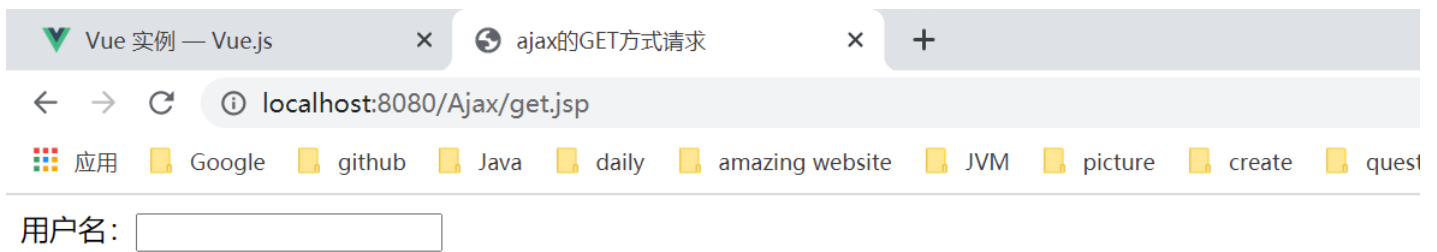
```
    if (xhr.readyState == 4 && xhr.status == 200){ // 代表完整且正确的响应
```

```
        xhr.responseText; // 用来获取action响应的字符串
```

```
        // 通过js jquery来更新页面
```

```
    }
```

```
}
```



之前每次引入js文件就是头疼

- `${pageContext.request.contextPath}/js/jquery.js` 【js是webapp下的路径】
- Maven -D clean
- 重新部署 后重启
- 再不行就使用cdn的方式，不推荐

2. GET方式请求

- 获取输入值
- 发送ajax请求
 - 创建xhr对象
 - 发送请求并传参

- 处理响应值

Java

```
1  // 页面加载完执行的事件
2  $(function () {
3      // 1.绑定用户名文本框失去焦点的时候触发
4      $("#username").blur(function () {
5          // console.log("我失去焦点啦，你们快点去检验好吧");
6          // 发送ajax请求，验证是否存在相同的用户名
7          // 1.获取用户输入值
8          var username = $(this).val();
9          // 2.发送ajax的get请求
10         // a.创建xhr对象
11         var xhr;
12         if (window.XMLHttpRequest) {
13             xhr = new XMLHttpRequest();
14         } else {
15             xhr = new XMLHttpRequest("Microsoft.XMLHTTP");
16         }
17         // b.发送请求并传递参数 url为请求接口
18         xhr.open("GET", "${pageContext.request.contextPath}/user/checkUsername?
username="+username);
19         xhr.send();
20
21         // c.处理响应并获取返回的字符串
22         xhr.onreadystatechange = function () {
23             if (xhr.readyState === 4 && xhr.status === 200) {
24                 // 处理正确返回字符串的响应
25                 console.log(xhr.responseText); // 响应结果
26                 // $("#msg").text(xhr.responseText);
27                 $("#msg").html("<font color='red'>" + xhr.responseText + "
</font>");
28             }
29         }
30     })
31 }
```

3. POST方式请求

与GET方式大同小异。

3.1 创建xhr

3.2 发送请求传递参数（区别之处）

Java

```
1 xhr.open("POST", "url");
2 // 用来模拟form进行参数编码
3 xhr.setRequestHeader("content-type", "application/x-www-form-urlencoded");
4 xhr.send("name=zhangsan&password=123");
```

3.3 处理响应并更新局部

以前只有form表单才能提交post请求，`enctype="multipart/form-data"`是对表单中二进制文件做编码，`enctype="application/x-www-form-urlencoded"`是默认的，如果没有二进制文件用这个效率高一点

```
<form action="" enctype="">
  用户名: <input id="username" type="text" value="" />
</form>
</body>
</html>
```

application/x-www-form-urlencoded
application/x-www-form-urlencoded
multipart/form-data
multipart/form-data
text/plain

Press Ctrl+. to choose the selected (or first) suggestion and insert a dot a

4. ★★返回数据格式JSON

前面返回的是字符串，如何处理对象比较麻烦，所以这里最好使用JSON格式来处理返回数据

Java

```
1 // js中的json数据对象定义
2 var student = {"id":21, "name":yjiewei, "age":23}; // json
3 console.log(typeof student);
4 // 可以根据对象中的属性名来获取属性
5 console.log(student.name / student.age);
```

那我们如何返回json格式数据，如何把对象转成 json 呢？

用工具啊，**fastjson / jackson ...**

1. 引入fastjson依赖

2. `JSONObject.toJSONString(user);` // user为实例对象 `{ "id":21, "name": "yjiewei", "age":23 }`
3. `return JSONObject.toJSONString(user);`
4. 前台拿到的依旧是string 不是一个object, 要转成一个json对象

```
`var jsonObj = eval("(" + xhr.responseText + ")")`
```

处理返回一个对象并展示

```
/**
 * 用来根据idea查询一个的ajax请求处理
 */
public String findOne() throws IOException {
    //1.收集数据
    //2.调用业务方法 userService.findOne(id)
    User user = new User(id, name: "小陈", age: 23, new Date());
    //3.处理响应 将对象转为json格式字符串
    String jsonString = JSONObject.toJSONString(user);
    HttpServletResponse response = ServletActionContext.getResponse();
    response.setContentType("application/json; charset=UTF-8");
    response.getWriter().println(jsonString);
    return null;
}
```

//处理完整且正确响应

```
if(xhr.readyState==4 && xhr.status==200){
    console.log(xhr.responseText);
    //将json格式字符串转为js对象
    var jsObj = eval("(" + xhr.responseText + ")");
    console.log(jsObj);

    var idLi = $("- 
").text("id: " + jsObj.id);
    var nameLi = $("- 
").text("name: " + jsObj.name);
    var ageLi = $("- 
").text("age: " + jsObj.age);
    var birLi = $("- 
").text("bir: " + jsObj.bir);
    $("#ul").empty().append(idLi).append(nameLi).append(ageLi).append(birLi);
}
```

处理一个数组并返回展示

```

1 // jquery封装foreach循环遍历
2 $.each(users, function(index, user){
3     console.log(user.id);
4 });
5
6 // 通常js
7 for(var i = 0; i < users.length; i++) {
8     console.log(users[i].id);
9 }

```

```

var xhrResponseText = "[{\"age\":23,\"bir\":\"2020-05-21\",\"id\":\"21\"}";
console.log(xhrResponseText);
var jsArr = eval("(" + xhrResponseText + ")");
console.log(jsArr);
$.each(jsArr, function(index, user) {
    console.log(index);
    console.log(user);
});

```

拿到的是字符串
需要转成数组，自动识别

//处理完整且正确的响应

```

if (xhr.readyState==4 && xhr.status==200) {
    console.log(xhr.responseText); //json格式字符串
    //将json格式字符串转为js对象
    var jsArr = eval("(" + xhr.responseText + ")");
    console.log(jsArr);
    //清空页面中原始ul
    $("ul").remove();
    //遍历数据
    $.each(jsArr, function(idx, user) {
        console.log(user);
        var ul = $("<ul/>");
        var idLi = $("<li/>").text("id: " + user.id);
        var nameLi = $("<li/>").html("<span style='color:gold;'> name: " + user.name);
        var ageLi = $("<li/>").text("age: " + user.age);
        var birLi = $("<li/>").text("bir: " + user.bir);
        ul.append(idLi).append(nameLi).append(ageLi).append(birLi);
        $("#bd").append(ul);
    });
}

```

5. 使用mybatis查询数据

1. 引入依赖+MySQL驱动
2. resources下添加mybatis-config.xml文件，在里面配置

3. 创建连接啥的，具体看一下文档
4. 写接口DAO，resources下的mapper文件，注意文件名要和接口类名一致
5. 环境配置、使用细节自己找找资料

6. ajax请求的改进

- 使用回调函数将想要获取的值返回；

The diagram illustrates the improvement of AJAX requests through three stages of code examples:

- Stage 1 (Left):** Basic XMLHttpRequest usage. It shows creating an xhr object, sending a GET request with a username parameter, and handling the response in an onreadystatechange event listener when readyState is 4 and status is 200.
- Stage 2 (Middle):** Similar to Stage 1, but the request URL is constructed with a phone number parameter.
- Stage 3 (Right):** The code is packaged into a reusable function named `get` in a file `ajax.js`. This function creates the xhr object and sends the request. The response handling is separated into a callback function `aa`, which is called with `xhr.responseText` as an argument. Red annotations highlight the `aa(xhr.responseText);` line, labeling it as a "回调函数" (callback function) and noting that "ajax请求不能return值" (AJAX requests cannot return values).

7. ★★jQuery对ajax的封装

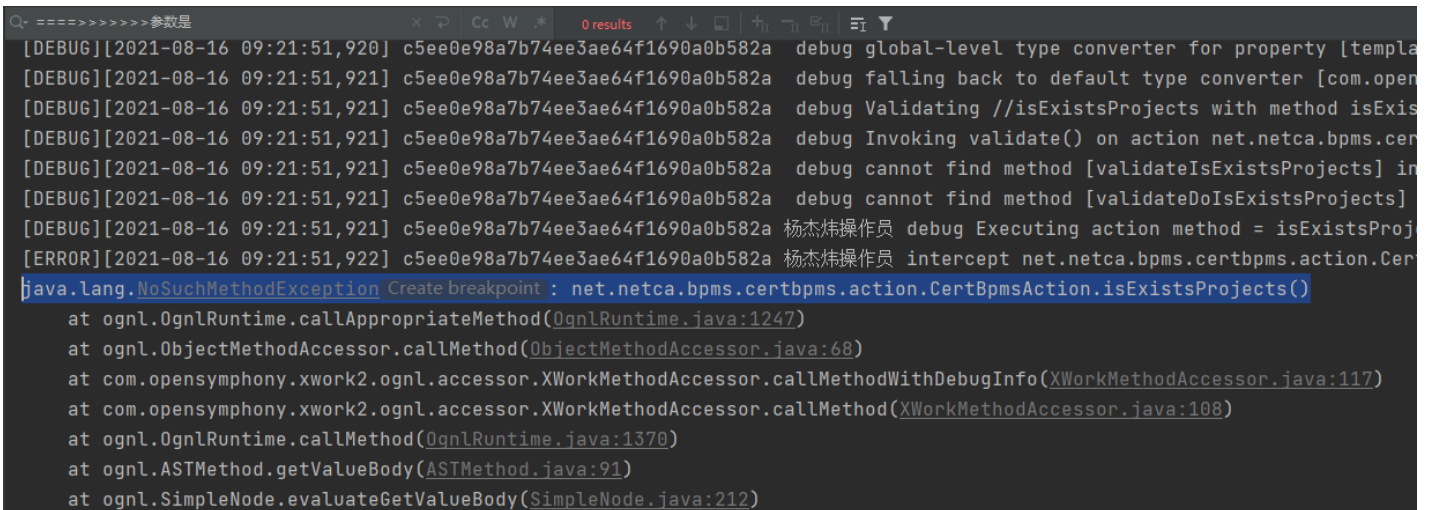
<https://jquery.cuishifeng.cn/>

<https://jquery.cuishifeng.cn/jQuery.Ajax.html>

- 引入jQuery文件
- 使用

Java

```
1 $.ajax({
2     type: "GET | POST",
3     url: "请求路径",
4     data: {} | "", // 对象或者字符串
5     success: function(data){
6         data; // 就是当前返回的响应结果
7         // 如果请求的URL返回的是json格式数据, 会自动转换成js对象
8         // 当然, 如果返回的是字符串, 自然还是字符串本身
9     }
10 });
```



```
====>>>>>>>>>>参数是
[DEBUG][2021-08-16 09:21:51,920] c5ee0e98a7b74ee3ae64f1690a0b582a debug global-level type converter for property [templa
[DEBUG][2021-08-16 09:21:51,921] c5ee0e98a7b74ee3ae64f1690a0b582a debug falling back to default type converter [com.open
[DEBUG][2021-08-16 09:21:51,921] c5ee0e98a7b74ee3ae64f1690a0b582a debug Validating //isExistsProjects with method isExis
[DEBUG][2021-08-16 09:21:51,921] c5ee0e98a7b74ee3ae64f1690a0b582a debug Invoking validate() on action net.netca.bpms.cer
[DEBUG][2021-08-16 09:21:51,921] c5ee0e98a7b74ee3ae64f1690a0b582a debug cannot find method [validateIsExistsProjects] in
[DEBUG][2021-08-16 09:21:51,921] c5ee0e98a7b74ee3ae64f1690a0b582a debug cannot find method [validateDoIsExistsProjects]
[DEBUG][2021-08-16 09:21:51,921] c5ee0e98a7b74ee3ae64f1690a0b582a 杨杰炜操作员 debug Executing action method = isExistsProj
[ERROR][2021-08-16 09:21:51,922] c5ee0e98a7b74ee3ae64f1690a0b582a 杨杰炜操作员 intercept net.netca.bpms.certbpms.action.Cer
java.lang.NoSuchMethodException: Create breakpoint : net.netca.bpms.certbpms.action.CertBpmsAction.isExistsProjects()
    at ognl.OgnlRuntime.callAppropriateMethod(OgnlRuntime.java:1247)
    at ognl.ObjectMethodAccessor.callMethod(ObjectMethodAccessor.java:68)
    at com.opensymphony.xwork2.ognl.accessor.XWorkMethodAccessor.callMethodWithDebugInfo(XWorkMethodAccessor.java:117)
    at com.opensymphony.xwork2.ognl.accessor.XWorkMethodAccessor.callMethod(XWorkMethodAccessor.java:108)
    at ognl.OgnlRuntime.callMethod(OgnlRuntime.java:1370)
    at ognl.ASTMethod.getValueBody(ASTMethod.java:91)
    at ognl.SimpleNode.evaluateGetValueBody(SimpleNode.java:212)
```

debug出真知！！后台接收参数不是在函数名上面接收的

进哪个大厂 -> 哪个部门 -> 职级多少 -> 工作时间多少 -> 哪条线

1. 返回值dataType是json
2. action后台函数不能再函数名接收参数，得是全局变量调用时注入
3. 返回值是在ajax外面，不是在里面，因为是函数的返回值，不是异步请求的
4. debug的时候没有正确走到流程，先看看有什么报错，找到问题出现的地方