

Ajax 和 Json

Ajax:

1. Ajax 的简介

Ajax 是一种能够向服务器请求额外的数据而无需卸载页面(无刷新)的技术，是对 Asynchronous Javascript + XML 的简写，因其良好的用户体验，现已成为 web 不可或缺的技术。而 Ajax+Json 或者 Ajax+XML 的组合极大的方便了 web 开发者。

Ajax 技术的核心是 XMLHttpRequest 对象，XHR 为向服务器发送请求和解析服务器响应提供了流畅的接口，能以异步的方式从服务器取得更多的信息，意味着用户单击后，可以不必刷新页面也能取得数据。使用 Ajax 的第一步就是申明一个 XMLHttpRequest 对象。

此外关于 Ajax 的实现原理，发展历史以及 XMLHttpRequest 对象的相关知识，不再赘述，可自行谷歌，或者到 w3c 上查阅相关资料。本文的主要内容 of ajax 的使用。

2. Ajax 的使用

(1). 原生 JS 下 ajax 的使用

客户端 js 实例代码如下：

```
48 var xmlhttp;
49 if(window.XMLHttpRequest){
50     xmlhttp=new XMLHttpRequest();
51 }else{
52     xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
53 }
54
55 //此XMLHttpRequest对象，xmlhttp将会把
56 xmlhttp.open("POST",url,true);
57 xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");
58 xmlhttp.send("action=get&id="+id);
59 xmlhttp.onreadystatechange=function(){
60     if (xmlhttp.readyState==4 && xmlhttp.status==200){
61         //得到从服务器返回的数据
62         var responseContent = xmlhttp.responseText
63
64         //对从服务器返回的数据进行处理
65         hanele(responseContent);
66
67     }else if(xmlhttp.readyState==4 && xmlhttp.status!=200){
68         alert('连接出错，请检查网络!');
69     }
70 }
71
```

实例中，第 48-53 行申明了一个 XMLHttpRequest 对象，在第 56 行中通过 open 函数规定请求的类型、URL 以及是否异步处理请求。57 行中通过 setRequestHeader 函数向请求添加 HTTP 头。第 58 行 send 函数将请求发送到服务器，send 函数的参数是要发送到服务器的数据，这些数据以“&”符号隔开，并以 key=value 的形

式传递。59-65 行表示的是：如果请求传输完成，且相应已就绪时，从服务器那端取出返回的数据(`var responseContent = xmlhttp.responseText`)，并将数据交给自定义的 `handle` 函数，进行继续处理。

关于一些函数的参数值所代表的意思以及函数中出现的一些值所表示的意思，请参考 w3c。http://www.w3school.com.cn/ajax/ajax_xmlhttprequest_create.asp

服务端代码示例如下：（以 `servlet` 为例）

```
1▼ protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
2    // TODO Auto-generated method stub
3    String action = request.getParameter("action");
4    String id=request.getParameter("id");
5
6    String result = handle(action, id);
7    PrintWriter out = response.getWriter();
8    out.println(result);
9    out.flush();
10   out.close();
11 }
```

示例中，第 3-4 行，通过前端发送的数据的 `key` 的值来获取参数的值，并将获取的传递过来的参数值交给 `handler` 函数处理，并得到结果 `result`。然后再由 `response` 把得到的结果写会给客户端。（6-8 行）。

(2). JQuery 中 ajax 的使用

JQuery 对 `ajax` 的实现进行封装，使得 `Ajax` 的使用变的更加方便，JQuery 提供了一些简单的方法来使用 `ajax`，比如 `load`，`jQuery.get`，`jQuery.post` 等方法，但是如果处理复杂的逻辑，还是需要用到 `jQuery.ajax()`。将用原生 `js` 实现的 `ajax` 用 `jQuery` 重写，代码如下：

```
74 $.ajax({
75     type: 'POST',
76     url : serverUrl,
77     data:{action:'get', id: myid},
78     success:function(data){
79         handle(data);
80     },
81     error:function(){
82         alert('连接出错，请检查网络!');
83     }
84 });
85 }
86
```

代码体现的逻辑不再赘述，如果有不懂的地方可以联系助教，或者 `google`。其中，`ajax()`方法中的参数，及其意义，可以参考 w3c。

http://www.w3school.com.cn/jquery/ajax_ajax.asp

Json

JSON 的全称是“JavaScript Object Notation”，意思是 JavaScript 对象表示法，它是一种基于文本，独立于语言的轻量级数据交换格式。XML 也是一种数据交换格式，为什么没有选择 XML 呢？因为 XML 虽然可以作为跨平台的数据交换格式，但是在 JS (JavaScript 的简写) 中处理 XML 非常不方便，同时 XML 标记比数据多，增加了交换产生的流量，而 JSON 没有附加的任何标记，在 JS 中可作为对象处理，所以我们更倾向于选择 JSON 来交 JSON 有两种表示结构，对象和数组。

对象结构以“{”大括号开始，以“}”大括号结束。中间部分由 0 或多个以“,”分隔的“key(关键字)/value(值)”对构成，关键字和值之间以“:”分隔。语法结构如代码。

```
1 {  
2     key1:value1,  
3     key2:value2,  
4     ...  
5 }
```

其中关键字是字符串，而值可以是字符串，数值，true, false, null, 对象或数组。

数组结构以“[”开始，“]”结束。中间由 0 或多个以“,”分隔的值列表组成，语法结构如代码：

```
[  
    {  
        key1:value1,  
        key2:value2  
    },  
    {  
        key3:value3,  
        key4:value4  
    }  
]
```

JSON 是 JS 的一个子集，所以可以在 JS 中轻松地读，写 JSON。读和写 JSON 都有两种方法，分别是利用“.”操作符和“[key]”的方式。我们首先定义一个 JSON 对象，代码如下。

```
var obj = {
    1: "value1",
    "2": "value2",
    count: 3,
    person: [ //数组结构JSON对象，可以嵌套使用
        {
            id: 1,
            name: "张三"
        },
        {
            id: 2,
            name: "李四"
        }
    ],
    object: { //对象结构JSON对象
        id: 1,
        msg: "对象里的对象"
    }
};
```

从 JSON 中读数据：

```
function ReadJSON() {
    alert(obj.1); //会报语法错误，可以用alert(obj["1"]);说明数字最好不要做关键字
    alert(obj.2); //同上

    alert(obj.person[0].name); //或者alert(obj.person[0]["name"])
    alert(obj.object.msg); //或者alert(obj.object["msg"])
}
```

比如要往 JSON 中增加一条数据，代码如下：

```
function Add() {
    //往JSON对象中增加了一条记录
    obj.sex= "男" //或者obj["sex"]="男"
}
```

Json 在 java 中的使用是通过使用 JSONObject 和 JSONArray，两者也可相互嵌套，示例如下所示，下图只是给了 json 在 java 下的一种用法，其他用法可到网上查阅相关资料。正因 java 和 js 均可方便的使用以及解析 json，故可以通过 json 来进行客户端和服务器的数据交互。

```
JSONObject obj1 = new JSONObject();  
obj1.put(key1, value1);  
obj1.put(key2, value2);  
obj1.put(key3, value3);  
JSONObject obj2 = new JSONObject();  
JSONObject obj3 = new JSONObject();  
  
JSONArray jsoarray = new JSONArray();  
jsoarray.put(obj1);  
jsoarray.put(obj2);  
jsoarray.put(obj3);
```