链滴

Ajax 介绍和简单用法

作者：goker

原文链接：https://ld246.com/article/1606660578121

来源网站：链滴

一. Ajax基础

1. 异步传输+js+xml

优点：无需刷新整个页面，就能更新局部数据的技术

缺点：打破了浏览器的back回退机制

2. 同异步

异步:当一行代码还没结束，就可以执行另一行代码

同步:当一行执行结束，才能执行另一行代码

3. 连接步骤

1. 创建　XMLHttpRequest请求对象
2. 发起open请求，需指定http的请求方式、请求地址
3. 向服务器发送数据
4. 监听服务器的响应和状态码
5. 判断是否连接成功执行的函数

4. 解决同源跨域问题

同源（三源）：协议、域名、端口号

dataType:"jsonp"，<script>的src不受同源策略的影响

java:@CrossOrigin

5. readyState状态码

0, 请求未初始化

1, 服务器已建立链接

2, 请求已接受

3, 请求处理中

4, 请求完成并且响应就绪

6. 两种http(TCP)发送请求方式：get和post区别

1. **请求的内容和作用不一样**

get 是请求指定的页面信息，并返回实体主体。

post在指定资源提交数据进行处理**（提交表单、上传文件）**，数据是包含在请求体中的。

原文链接：Ajax 介绍和简单用法

2. **请求的参数直接在浏览器 地址栏中显示**

get会：参数是**通过 url 传递**

post不会：参数是**通过 Request body**

3. **发送次数不一样**

get 会产生一个数据包：浏览器会先http header和data一起发送出去，服务器响应200（返回数据）

post 会产生两个数据包(2次)：两次发包的优点，可以保证数据的完整性

浏览器会先发送header，服务器响应100(临时请求) continue，在发送data，服务器响应200（返回据）。

4. **存储的容量不一样**

get 存储的参数**不能超过 2k**

post 没有限制

7. js 和 jq ajax的码法

1. js

```
<script>
    // 1.创建一个xmlhttp请求对象
    if (window.XMLHttpRequest) {
        var xhr = new XMLHttpRequest();
    } else {
        // 兼容ie低版本
        var xhr = new ActiveXObject("Microsoft.XMLHTTP")
    }
    console.log(xhr);
    // 2.发起请求 open("请求方式","请求地址")
    xhr.open("get", "js/mydata.json")
    console.log(xhr)
    // 3.向服务器发送数据
    xhr.send();
    console.log(xhr)
    // 4.事件监听服务器响应 状态码
    xhr.onreadystatechange = function () {
        console.log(this);
        // 如果事件连接成了 4 和 200
        if (this.readyState == 4 && this.status == 200) {
            // json字符串 转换成 json对象 |【对象转字符】[JSON.stringify()]
            var res = JSON.parse(this.responseText);
            // 变量所有对象k，v
            for (var k in res) {
                console.log(k, res[k]);
                //  创建节点【标签】
                var p = document.createElement("p");
                var left = document.createElement("span");
                var right = document.createElement("span");
                //   把json对象值添加进去了
                left.innerHTML = k;
                right.innerHTML = res[k];
                //  往box里添加节点
```

```
            box.appendChild(p);
            p.appendChild(left);
            p.appendChild(right);
        }
    }
  }
</script>
```

2. jquery

```
<script>
  $(function () {
    $.ajax({
      // 请求地址
      url: "js/mydata.json",
      // 请求方式
      type: "get",
      // 是否为异步，true为异步，false为同步
      async: "true",
      // 向服务器端传送数据
      data: {},
      // 请求成功返回的函数
      success: function (res) {
        console.log(res);
        //jq的遍历方式
        $.each(res, function (k, v) {
          console.log(k, v);
          $(".box").append(" <p>\n" +
            "        <span>" + k + "</span>\n" +
            "        <span>" + v + "</span>\n" +
            "    </p>")
        })
      },
      // 请求失败返回的函数
      error: function (res) {
        console.log(res);
      }
    })
  })
</script>
```