第14期课堂笔记(第七周第三天)

HTTP事物

HTTP传输的这件事，能称之为事物，说明这件事情比较的完整：Request + Response，只有请求和响应两个阶段都完成，才算HTTP事物完成

HTTP报文

1、客户端不仅可以向服务器发送请求，还可以把一些内容传递给服务器
2、服务器端也会把一些内容返回给客户端

> 客户端传递给服务器的内容以及服务器返回给客户端的内容统称为“HTTP报文”：请求报文、响应报文

不管是请求还是响应报文，内容中都包含三部分:

- 起始行：请求起始行、响应起始行
- 首部(头)：请求头、响应头、通用头、自定义请求头、自定义响应头…
- 主体：请求主体、响应主体

以上这些信息在谷歌浏览器控制台的NetWork选项中都可以查看到；

- 请求报文是由客户端设置，由服务器端获取
- 响应报文是由服务器端设置，由客户端获取

客户端把一些内容传递给服务器的办法：

- URL地址末尾问号传参
- 通过设置请求头，把一些内容给服务器端
- 通过设置请求主体，把一些内容传递给服务器

服务器把一些内容传递给客户端的办法：

- 设置响应头
- 设置响应主体

AJAX

Asynchronous Javascript And XML ：异步的JS和XML

扫盲

html、xhtml、html5、dhtml、xml、wxml
1、html：超文本标记语言
2、xhtml：更加严格的html
3、html5：对传统HTML的升级改革,增加了很多内容
4、dhtml：页面中的内容是动态绑定
5、xml：可扩展的标记语言(里面使用的标签基本上都是自己定义的)；用自己扩展的一些有意义的标签来清晰描述一组数据结构和存储一组数据；
6、wxml：weChat xml 微信小程序的页面

AJAX是异步的JS:
此异步不是我们想象中的那个同步异步(我们想象的同步异步是:当前这件事没完成下面的事情可以继续去做叫做异步,反之是同步)，这里面的异步，可以用一个词描述：”局部刷新”

AJAX作用

客户端可以使用AJAX技术从服务器端获取到需要的数据，然后绑定在页面中显示，在前后端分离项目中，AJAX/JSONP是最主要的一种技术之一

AJAX的语法

```
1.  var xhr=new XMLHttpRequest(); //->创
    建一个AJAX对象
2.  xhr.open([method],[request url],[asyn
    c],[user name],[user pass]); //->打开
    一个请求连接(做AJAX发送前请求前的配置准备)
3.  xhr.onreadystatechange=function(){
4.      //->只要AJAX状态发生改变，事件就会被触
    发，绑定的方法也会被执行
5.      if(xhr.readyState===4 && xhr.stat
    us===200){
6.          //->xhr.readyState:AJAX状态码
7.          //->xhr.status:服务器状态码
8.
9.          xhr.responseText; //->获取服务
    器返回的响应主体内容
10.     }
11. }
12. xhr.send(null); //->发送AJAX请求给服务
    器,在没有执行send之前,AJAX请求这件事不算开
    始,只有执行了SEND后，AJAX这件事才算开
    始；当我们客户端已经获取到响应主体内容的时
    候(xhr.readyState===4的时候)，AJAX这件
    事才算结束；SEND中放入的是请求主体内容；
```

第一步：创建AJAX对象

var xhr=new XMLHttpRequest();
在IE6及以下版本的浏览器中不兼容，我们需要使用ActiveXObject来处理

> 课后扩充：
> http://www.ablesky.com/kecheng/detail_1029018
> 第七章第二节课件3:AJAX兼容处理及惰性思想

第二步：打开一个请求的连接地址

xhr.open([method],[request url],[async],[username],[userpass]);

1、method设置请求的方式

> GET系列
>
> - get 获取
> - delete 删除
> - head 只获取服务器响应头信息
>
> POST系列
>
> - post 推送
> - put 放
>
> ……

GET系列和POST系列的区别:

> GET主要应用于给服务器的少，从服务器获取的多，例如：获取服务器上的数据进行展示和数据绑定…
> POST主要应用于给服务器的多，从服务器获取的少，例如：用户注册…
> 在真实项目中，我们使用GET请求，会把传递给服务器的内容“通过问号传参的方式”传递给服务器；而POST请求是把内容放在“请求主体”中传递给服务器的；

1)GET请求专递给服务器的内容有大小限制

> xhr.open(‘get’,’/temp?name=zf&age=8…’);
> 每一个浏览器对于URL的长度是有限制的(谷歌8KB、火狐7KB、IE2KB)，如果传递的内容比较的多，URL会很长，超出的部分，浏览器会截取。
> POST理论上没有大小的限制，真实项目中为了节约网络消耗和提高性能，我们自己会做最大的限制

2)GET请求不安全,POST相对安全

> GET是通过问号传参传递信息的，很容易被url劫持，导致信息的泄露或者恶意修改等

3)GET请求容易出现缓存(并且这个缓存不被我们控制)

> 还是因为GET是通过问号传参传给服务器的 xhr.open(‘get’,’/temp?id=12’); 只要我们两次及以上请求相同的地址传递相同的内容，浏览器很容易就把第一次的结果作为缓存
> 我们可以使用在URL末尾追加随机数的方式清除缓存：xhr.open(‘get’,’/temp?id=12&_=’+Math.random())

2、request url 请求的URL地址(真实项目中的这个地址是服务器告诉我们的)

3、设置同步异步，默认是true，代表异步，传递的是false，代表同步

4、设置用户名和密码(基本上不用)

第三步：监听AJAX状态的改变

```
xhr.onreadystatechange=function(){
//->js code
}
```

AJAX的各种状态

> xhr.readyState
>
> - 0 UNSEND 未发送，创建一个AJAX对象，默认的状态就是零
> - 1 OPENED 已打开，也就是把xhr.open(…)执行了
> - 2 HEADERS_RECEIVED 已经获取到服务器的响应头信息了
> - 3 LOADING 响应主体内容正在加载
> - 4 DONE 响应主体内容加载完成

服务器响应状态码(HTTP网络状态码):

> 200 成功
> 301 永久重定向(永久转移)
> 302 临时重定向(临时转移) 服务器负载均衡
> 304 读取缓存数据(对于不经常改变的CSS/JS等文件,我们可以做304缓存)
> 400 传递参数有误
> 401 没权限
> 404 找不到
> 500 未知的服务器错误
> 503 服务器超负载

获取服务器响应主体的内容：

> xhr.responseText 获取的一般都是JSON格式的字符串数据或者是纯文本数据再或者二进制文本数据
> xhr.responseXML 获取的是XML格式的数据

第四步：SEND发送请求

xhr.send([requestBodyContent]); 只有POST系列请求，才会把传递给服务器的内容，通过请求主体传递给服务器，GET系列下，我们直接传递null即可;

AJAX实战开发

案例一：客户端获取服务器端的时间
在限时抢购或者倒计时的案例中，我们不能用客户端本地的时间做为参考依据，需要从服务器上获取服务器时间来处理…

```
var xhr = new XMLHttpRequest;
xhr.open('head', 'student.xml?_=' + Math.random());
xhr.onreadystatechange = function () {
    if (xhr.status !== 200) return;
    if (xhr.readyState === 2) {
        //->HEADERS_RECEIVED
        //响应头信息已经接收了,在响应头中就已经有服务器的时间了

        //->xhr.getResponseHeader:获取响应头信息
        //获取到的服务器时间是格林尼治时间GMT(字符串类型),我们还需要把获取的时间转换为北京标准时间
        var time = xhr.getResponseHeader('Date');
        time = new Date(time);
        console.log(time);
    }
};
xhr.send(null);
```