AJAX

> ajax：async(asynchronous) javascript and xml 异步的JS和XML
>
> 作用：
> 通过AJAX技术，客户端可以向服务器端发送请求，把需要展示给用户的数据获取到，也可以把客户端用户填写的一些信息发送给服务器端进行处理 -> `ajax实现的是前后端数据请求交互的`

AJAX中异步JS的概念

> 此处的异步和我们之前学习的同步异步还是有点区别的，这里提到的异步其实表达的意思的是 “局部刷新”

`全局刷新`

> 客户端看到的页面都是有服务器端进行渲染的：服务器把需要呈现的结构样式以及动态的数据都渲染完成，客户端浏览器拿到最终的代码呈现给用户即可
>
> 如果当前页面中的部分结构和数据需要改变，需要由服务器端重新渲染，把最新渲染的结果返回给客户端，客户端通过整体刷新来展示最新效果

`局部刷新`

> 局部刷新不是服务器端来渲染页面，客户端通过AJAX等技术向服务器发送请求，服务器只需要把数据返回给客户端即可，页面的渲染是由客户端完成的
>
> 如果有一个区域的数据需要发生改变，只需要在重新发送AJAX请求，获取最新的数据，有客户端重新的把当前区域的内容渲染即可，页面不需要重新加载

XML

html：超文本标记语言
xhtml：更加严谨的html
dhtml：页面中的部分数据是动态更新的

xml：可扩展的标记语言，相当于html来说，html中的标签都是w3c规定的，而xml中的标签大部分都是自己随意设置的

作用:
xml的作用不在展示，而在于存储，自己扩展一些有意义的标记标签，按照指定的结构，清晰明了的存储一些数据信息；在很早以前，使用AJAX实现和客户端和服务器端数据交互，所采用的数据格式一般都是xml格式的，由于xml格式解析的时候不太方便，所以目前项目中都是使用json格式代替xml格式的数据传输；

数据传输常用的格式：

- json(一般都是json格式的字符串)
- xml
- 文件流(buffer、二进制、base64…)
- …

```xml
<root>
    <student>
        <name>张三</name>
        <age>28</age>
        <score>
            <english>98</english>
            <chinese>99</chinese>
            <math>100</math>
        </score>
    </student>

    <student>
        <name>李四</name>
        <age>27</age>
        <score>
            <english>8</english>
            <chinese>140</chinese>
            <math>149</math>
        </score>
    </student>
</root>
```

基础语法

```
//->1、创建一个AJAX对象
let xhr = new XMLHttpRequest;

//->2、打开请求的链接（配置ajax请求的基础参数）
xhr.open([METHOD],[URL],[SYNC/ASYNC],[USER-NAME],[USER-PASSWORD]);

//->3、监听状态请求，获取需要的数据
xhr.onreadystatechange = function (){
    xhr.readyState：ajax状态码 0~4
    xhr.status：HTTP网络状态码

    xhr.responseText：获取服务器端返回的字符串数据(json字符串)
    xhr.responseXML：获取服务器端返回的xml格式数据
    ...
}

//->4、发送ajax请求：传递的内容可以是null也可以是其它信息
xhr.send();
```

open

xhr.open('get','/getList')
第三个参数不写默认是true:异步，写false是同步

最后两个参数：用户名和密码
我们一般不设置，只有当服务器端做安全限制的时候，限定某些用户才可以访问服务器，此时我们需要提供安全验证的用户名和密码

第一个参数：请求方式
[GET]
get、delete、head
[POST]
post、put

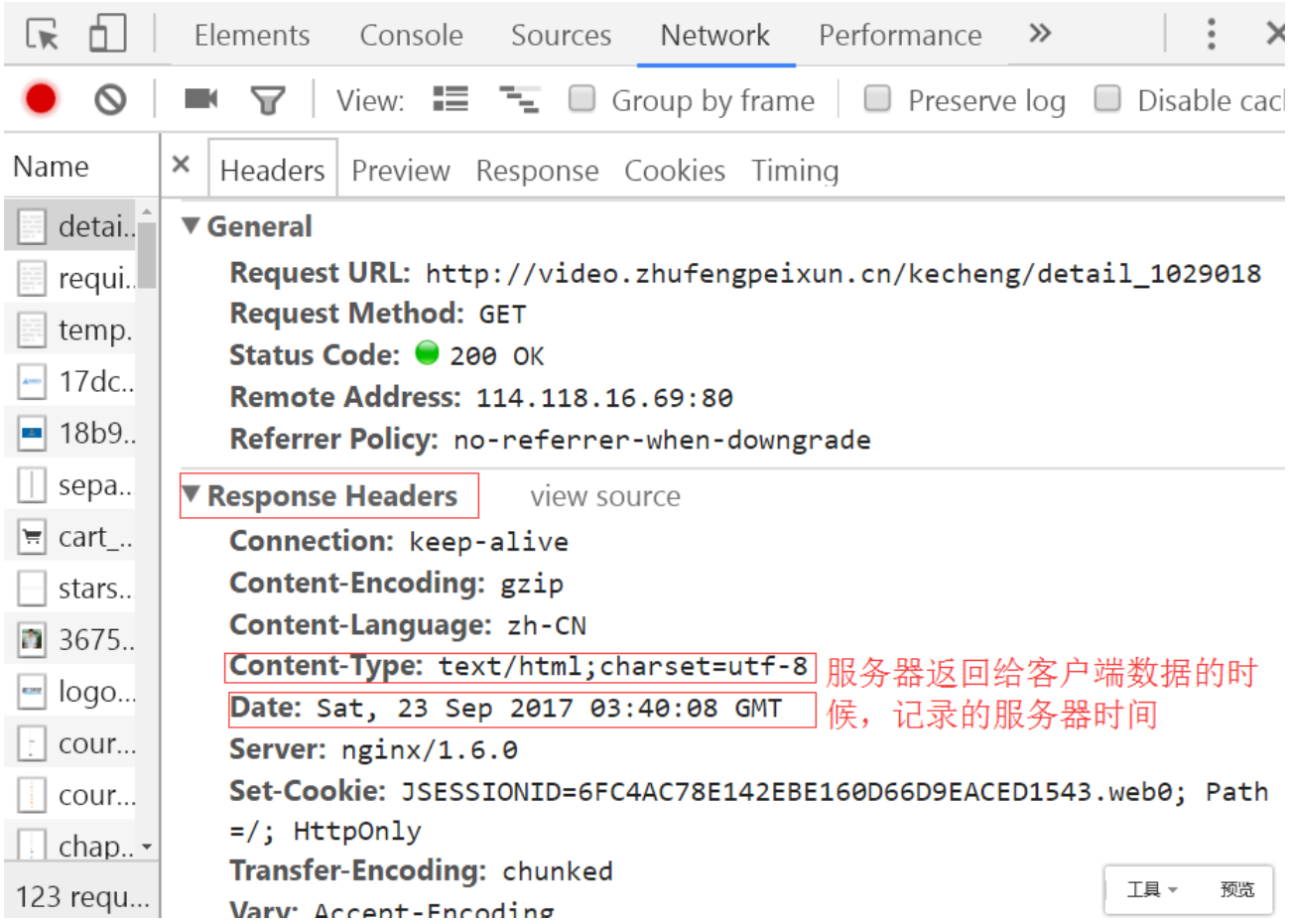
第二个参数：请求的URL地址
通过此地址我们可以向服务器发送请求，获取不同的数据，地址其实一个标识作用，告诉服务器端我们想要获取哪些数据；真实项目中，后台开发工程师会提供给我们一个API接口文档，在文档中声明了获取不同数据的不同接口地址；

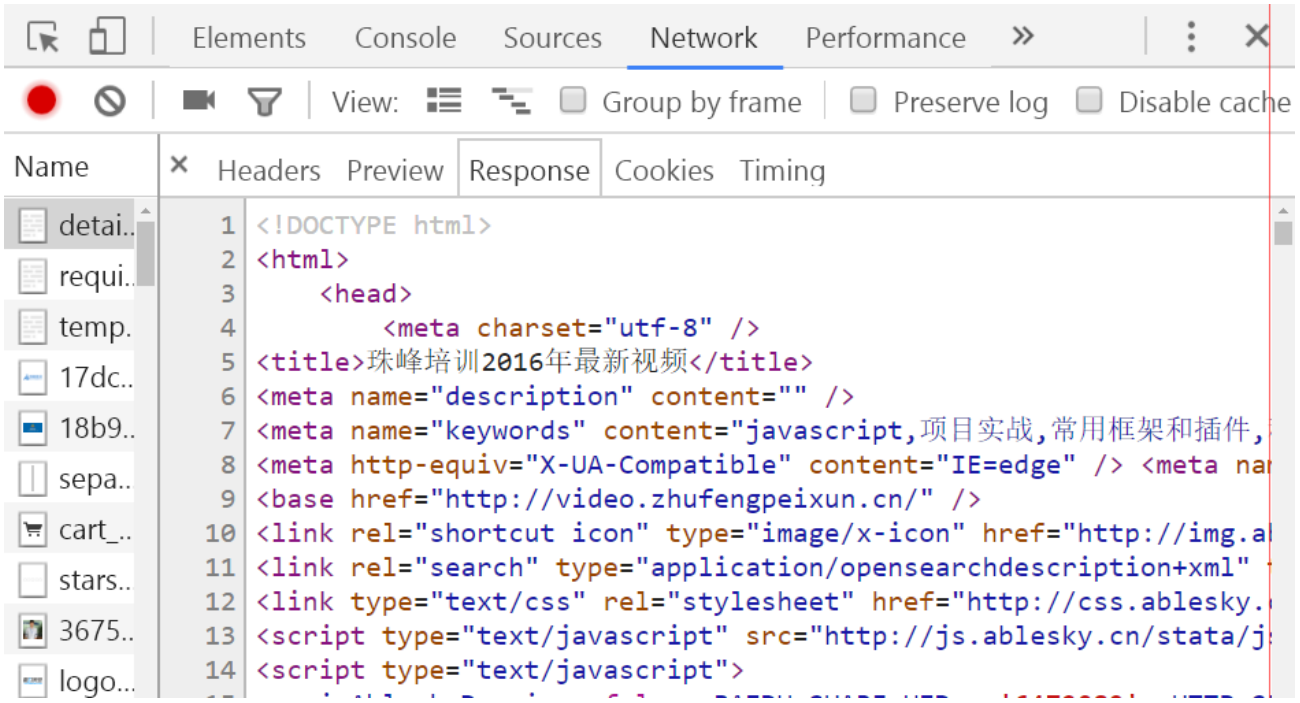
readystatechange

> readystatechange：当ajax状态发生改变会触发这个事件
>
> xhr.readyState：0 1 2 3 4
> 0：UNSENT 未发送，创建一个ajax实例，默认状态就是0
> 1：OPENED 以打开，执行xhr.open后，状态变为1
> 2：HEADERS_RECEIVED 客户端已经获取到服务器端返回的响应头信息
> 3：LOADING 服务器返回的响应主体内容正在传输中
> 4：DONE 客户端已经接收到服务器返回的响应主体内容(代表当前请求已经完成了)

响应头和响应主体

> 服务器端在交互的时候肯定会把一些数据返回给客户端，而返回给客户端的数据分为两种：通过响应头返回 和 通过响应主体返回；而获取响应头信息有两个方法：xhr.getAllResponseHeaders / xhr.getResponseHeader





请求头和请求主体

客户端向服务器端发送请求的时候，有时候也需要把一些信息传递给服务器端，而传递给服务器端有以下几种方式：

- 通过设置请求头把信息传递给服务器
- 通过请求主体把内容传递给服务器端
- 通过URL请求地址后面问号传递参数的方式把内容传递给服务器

xhr.setRequestHeader 设置请求头信息

timeout

xhr.timeout：设置请求的超时时间，当前请求如果超过了这个时间还没有请求完成，属于超时，请求中断，同时也会触发 xhr.ontimeout这个事件

AJAX请求方式

客户端向服务器端发送请求的时候，有很多的请求方式
1、GET系列的：GET、DELETE、HEAD
GET系列的目的一般都是从服务器端获取数据，例如：获取一些数据信息然后展示在页面中

- delete：删除，从服务器上删除某些资源
- head：只从服务器上获取响应头部分的信息，此请求无法获取响应主体信息

2、POST系列的：POST、PUT
POST系列的目的一般都是把客户端的一些数据推送给服务器端，例如：用户注册，客户端需要把用户填写的数据获取到发送给服务器

- put：存放，把一些资源存放在服务器上

不管哪一种方式，客户端都可以把一些信息传递给服务器，服务器也可以把一些信息返回给客户端，GET可以理解为‘给的少拿的多’，POST可以理解为‘给的多拿的少’

1、GET和POST的主要区别

GET方式：传递给服务数据信息采用的是url问号传递参数的方式来完成的
`xhr.open('get','/getInfo?name=zxt&age=27&sex=0')`

POST方式：传递给服务器的数据信息采用的是设置请求体传送
`xhr.send('name=zxt&age=27&sex=0')` 此方法中存放的就是请求主体内容

2、GET请求传递给服务器的内容有长度的限制,POST没有

GET是问号传参传递的，如果传递的内容过多，会导致URL过长
浏览器对于URL地址有一个长度限制，谷歌一般限制在8kb，火狐限制在7kb，IE限制在2kb，超过限制浏览器会把超出的部分截掉

请求主体中传递内容没有大小的限制，所以POST请求无大小限制；但是真实项目中，为了保证传输的速度，我们会人为的控制传输内容大小

3、GET请求会产生缓存（缓存是不可控的），POST不会

因为GET是问号传参传递给服务器内容，第一次通过地址传递某些参数给服务器，获取到一些最新的数据，第二次如果传递给服务器的参数没有变，很可能从服务器端获取的数据还是和上一次一样，而不是最新的数据

清除缓存：
手动在请求的URL地址末尾追加一个随机数，保证每一次请求的地址都不完全一样即可
`xhr.open('get','/getNewInfo?type=a&_='+Math.random())`

4、GET请求没有POST请求安全

因为GET请求是问号传递参数的形式，很多时候，别人可以通过黑客技术，劫持我们发送的url，如果是问号传递参数，传递给服务器的信息就会被劫持掉或者恶意篡改

一般所有涉及到安全性的信息，我们都使用POST，例如：所有涉及账号密码的都应该使用POST传输

HTTP网络状态码

通过一些标识数字反映出当前服务器的处理状态
https://baike.baidu.com/item/HTTP%E7%8A%B6%E6%80%81%E7%A0%81/5053660?fr=aladdin

200：成功，一切正常

301：Moved Permanently 永久转移 (永久重定向)
302：Move Temporarily 临时转移 (临时重定向) 服务器负载均衡
307：Temporary Redirect 临时重定向
304：Not Modified 读取的是缓存中的数据

400：Bad Request 请求参数错误
401：Unauthorized 请求权限不够
404：Not Found 请求地址不存在

500：Internal Server Error 未知的服务器错误
503：Service Unavailable 服务器超负荷请求

真实项目中的开发流程

涉及的岗位

1、产品：把一个想法转换为现实可执行可运作的需求，主要工作：需求分析、写需求文档、画原型图(AXURE)

2、UI设计师：拿到产品的原型图和需求，开始进行设计，最后设计出一版有样式、结构清晰、色彩搭配合理的图片(xxx.psd / xxx.png)

3、开发工程师：
前端开发：拿到设计稿之后，开始切图、写页面、写效果；
后台开发：拿到设计稿和需求文档后，开始构建数据库，实现一些数据处理或者业务处理的代码；

很多公司在开发工作之前，前端和后台需要聚在一起开一个项目分析会议：协定接口规范（API接口文档）

4、测试：开发完成后，提测

5、部署上线

6、产品推广