XMLHTTPRequest的概述：

最早是在IE5中以ActiveX组件的形式实现的。非W3C标准。

创建XMLHttpRequest对象：

——IE把XMLHttpRequest实现为一个ActiveX对象；

——其他浏览器把它实现为一个本地的javascript对象；

——XMLHttpRequest在不同的浏览器上的实现是兼容的。所以可以用同样的方法访问XMLHttpRequest实例的属性和方法，而不论这个实例创建的方法是什么。

创建XMLHttpRequest对象

为了每次写Ajax时都节省时间，可以把对象检测的内容打包成一个可复用的函数：

function getHTTPObject()

{

var xmlhtt = false;

if (window.XMLHttpRequest)

{// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp=new XMLHttpRequest();

}

Else if

{// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

return xmlhttp;

}

对window.XMLHttpRequest的调用会返回一个对象或者null，如果对象存在，则实例化。

用原生写比较麻烦，以后用框架写兼容性会比较好

XMLHttpRequest的方法



主要是send和open

XMLHttpRequest的属性



服务端把内容准备好了并返回，那客户端就要准备接收，用到这些属性

服务端发送数据每一个状态的改变都会导致onreadystatechange事件被触发

发送请求：

利用XMLHttpRequest实例与服务器进行通信包含以下3个关键部分：

——onreadystatechange事件处理函数；

——open方法；

——send方法。

第一个程序：

JSP

（如果点击链接跳转到另外一个页面上去，这样并不属于AJAX）

<html>

<head>

<title>第一个JSP程序</title>

</head>

<script type="text/javascript">

window.onload = function(){

//1、获取a节点，并为其添加onclick响应函数

document.getElementsByTagName("a")[0].onclick = function(){

//3、创建一个XMLHTTPREQUEST对象

var request = new XMLHttpRequest()

//4、准备发送请求的数据

var url = this.href;

var method = "GET";

//5、调用XMLHttpRequest对象的open方法

request.open(method,url);

//6、调用XMLHttpRequest对象的send方法

request.send(null);

//7、为XMLHttpRequest对象添加onreadystatechange响应函数

//8、判断响应是否完成：XMLHttpRequest对象的readystate属性值为4的时候

//9、再判断响应是否可用：XMLHttpRequest对象的status属性值为200

//10、打印响应结果：responseTest

//2、取消a节点的默认行为

return false;//不让点击反应

}

}

</script>

<body>

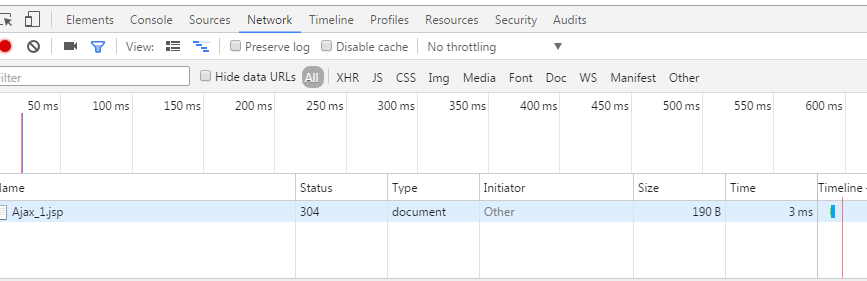
<a href = "helloajax.txt">HelloAjax</a>

</body>

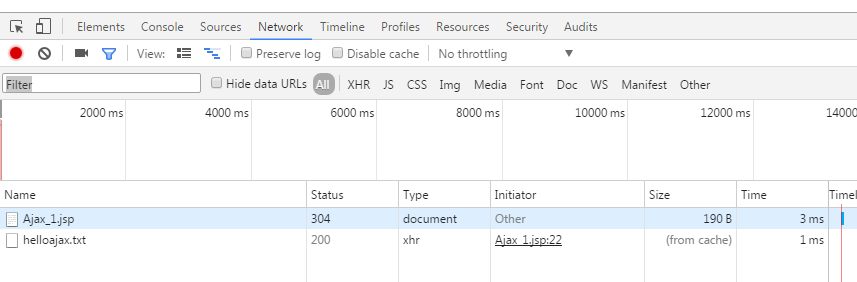
</html>

使用浏览器的审查元素：

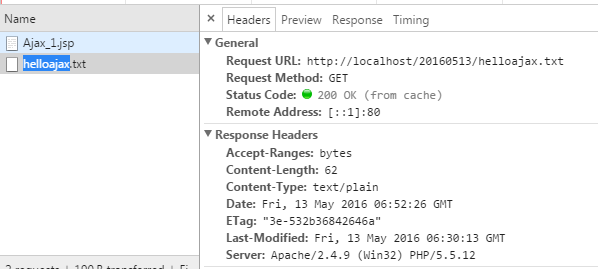
打开页面，刷新时：



点击时：



这个时候发了一个GET请求：



<html>

<head>

<title>第一个JSP程序</title>

</head>

<script type="text/javascript">

window.onload = function(){

//1、获取a节点，并为其添加onclick响应函数

document.getElementsByTagName("a")[0].onclick = function(){

//3、创建一个XMLHTTPREQUEST对象

var request = new XMLHttpRequest()

//4、准备发送请求的数据

var url = this.href;

var method = "GET";

//5、调用XMLHttpRequest对象的open方法

request.open(method,url);

//6、调用XMLHttpRequest对象的send方法

request.send(null);

//7、为XMLHttpRequest对象添加onreadystatechange响应函数

request.onreadystatechange = function(){

//8、判断响应是否完成：XMLHttpRequest对象的readystate属性值为4的时候

if(request.readyState == 4){

//9、再判断响应是否可用：XMLHttpRequest对象的status属性值为200

if(request.status == 200 || request.status == 304){

//10、打印响应结果：responseTest

alert(request.responseText);

}

}

}

//2、取消a节点的默认行为

return false;//不让点击反应

}

}

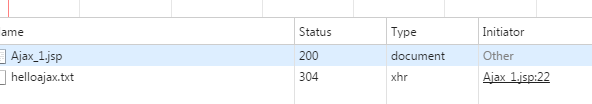
</script>

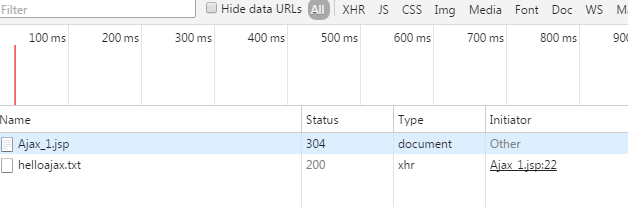
<body>

<a href = "helloajax.txt">HelloAjax</a>

</body>

</html>





添加，观察变化：

//7、为XMLHttpRequest对象添加onreadystatechange响应函数

request.onreadystatechange = function(){

alert(request.readyState)

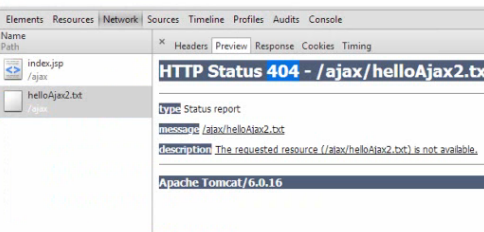
//8、判断响应是否完成：XMLHttpRequest对象的readystate属性值为4的时候

if(request.readyState == 4){

//9、再判断响应是否可用：XMLHttpRequest对象的status属性值为200

if(request.status == 200 || request.status == 304){

如果<a href = "helloajax.txt">HelloAjax</a>这个文件不存在，将返回404

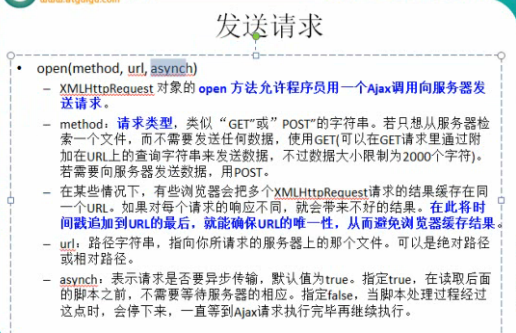


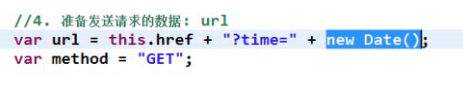
发送请求：

onreadystatechange:

——该事件处理函数由服务器触发，而不是用户

——在Ajax执行过程中，服务器会通知客户端当前的通信状态。这依靠更新XMLHttpRequest对象的readystate来实现。改变readystate属性是服务器对客户端连接操作的一种方式。每次readystate属性的改变都会触发readystatechange事件。







接收响应：

用XMLHttpRequest的方法可向服务器发送请求。在AJAX处理过程中，XMLHttpRequest的如下属性可被服务器更改：

——readystate

——status

——responseText

——responseXML



