**AJAX**

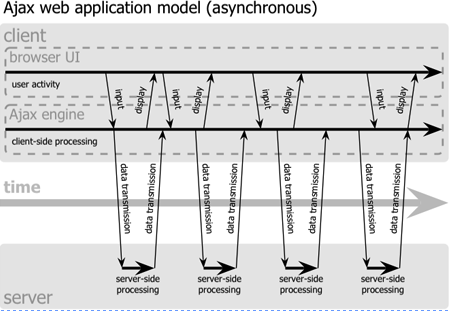
**AJAX概述**

-> Asyncchronous JavaScript And XML(AJAX)，翻译成中文则是"异步JavaScript和XML"，它是一种整合了多种技术的技术手段，其核心则是JavaScript内置的可用于和服务器进行HTTP通信交互的API函数：XMLHttpRequest

-> AJAX的设计理念分为两点，其一则是与服务器进行异步交互，其二则是针对浏览器的局部刷新，不可否认的是，AJAX的使用是实现前后端分离的一种重要的技术手段

-> 为了实现AJAX技术，早期微软的IE5、IE6浏览器内嵌了XMLHTTP组件，其他浏览器诸如：Opera、Mozila的早期版本则内嵌了XMLHttpRequest组件，XMLHTTP和XMLHttpRequest有很多相同的属性和方法，因此XMLHTTP也被一起称为XMLHttpRequest，简称XHR，后来XHR被W3C组织进行了标准化，各浏览器也逐渐按照W3C制定的标准来实现XHR，到目前为止，任然有部分的属性和方法不被部分浏览器所支持，因此在使用AJAX开发的时候应考虑一个浏览器的兼容性问题

-> 关于AJAX的处理模型



**AJAX需要注意的问题**

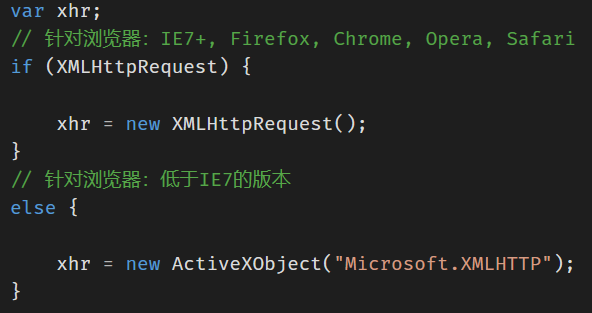
-> 当使用AJAX请求服务器的资源返回了一个重定向的标识，这个重定向虽然是成功Response到浏览器的手上，但是浏览器却不会发生重定向的现象，因为AJAX本身就是一个页面无刷新的请求API，当重定向的Response的内容返回至浏览器的时候，虽然浏览器的确是根据Response首部字段【Location】的URL地址发起了一次新的get请求，但最终get请求的结果不会交到浏览器手上，而是会被AJAX所捕获，所以这时候我们会发现AJAX在执行完Response数据下载后的回调结果为所重定向页面的HTML代码，那么如何解决这一问题？我们可以自己模拟一次重定向的过程，当AJAX请求服务器资源的时候，该资源文件不发起重定向的Response给浏览器，而是返回一段JSON，该JSON中包含了状态码【302】，还有重定向的URL地址【Location】，当AJAX下载完成数据后执行回调并获取Response Data的时候，判断状态码是否为【302】，如果是的话则指定当前浏览器的地址属性（location.href）为参数【Location】的值，当该属性改变后浏览器会自动地向该地址发起一次新的get请求，并且get请求的结果最终会被浏览器所捕获，也就是页面成功跳转到了【Location】所指示的URL当中，以此来完成一次模拟重定向的过程

**XMLHttpRequest的使用**

1. 创建XMLHttpReuqest对象

-> 正如上一节所说到的浏览器兼容性的问题，在创建XMLHttpRequest对象的时候我们需要把这一问题纳入到思考行列中的

-> XMLHttpRequest继承于【XMLHttpRequestEventTarget】接口，该接口提供了7种不同的回调函数，XMLHttpRequest所继承的这7中回调函数大多数都在数据下载的过程当中被触发，详情继续查看下面的章节

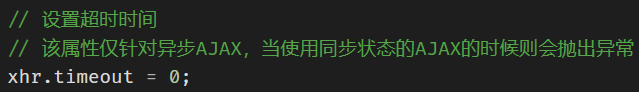


1. XMLHttpRequest中的属性
2. timeout：设置请求的超时时间

-> 该属性只能够针对异步模型的AJAX使用，当我们使用同步模型的AJAX的时候，指定该属性则会报错

-> 当我们设置该属性的值为0的时候，则表明不需要指定超时时间

-> 该属性能够配合【ontimeout】属性为其指定一个超时监听回调函数



1. responseType：用于指定Response需要返回的数据类型

-> ""：字符串类型，它是XMLHttpReuqest的默认类型，支持任何格式的Reponse的访问

-> arraybuffer：二进制缓冲数组类型，该类型不支持【XMLHttpRequest.ResponseText/ResponseXML属性的访问

-> blob：二进制大对象，该类型不支持【XMLHttpRequest.ResponseText/ResponseXML】属性的访问

-> json：JSON对象类型，该类型不支持【XMLHttpRequest.ResponseText/ResponseXML】属性的访问

-> document：XML类型，该类型不支持【XMLHttpRequest.ResponseText】属性的访问

-> text：文本类型，该类型不支持【XMLHttpRequest.ResponseXML】属性的访问



1. readyState：该属性的值用于表示XMLHttpRequest在不同阶段中的工作状态的体现

-> 0 UNSENT：最初始化状态，还未调用open函数

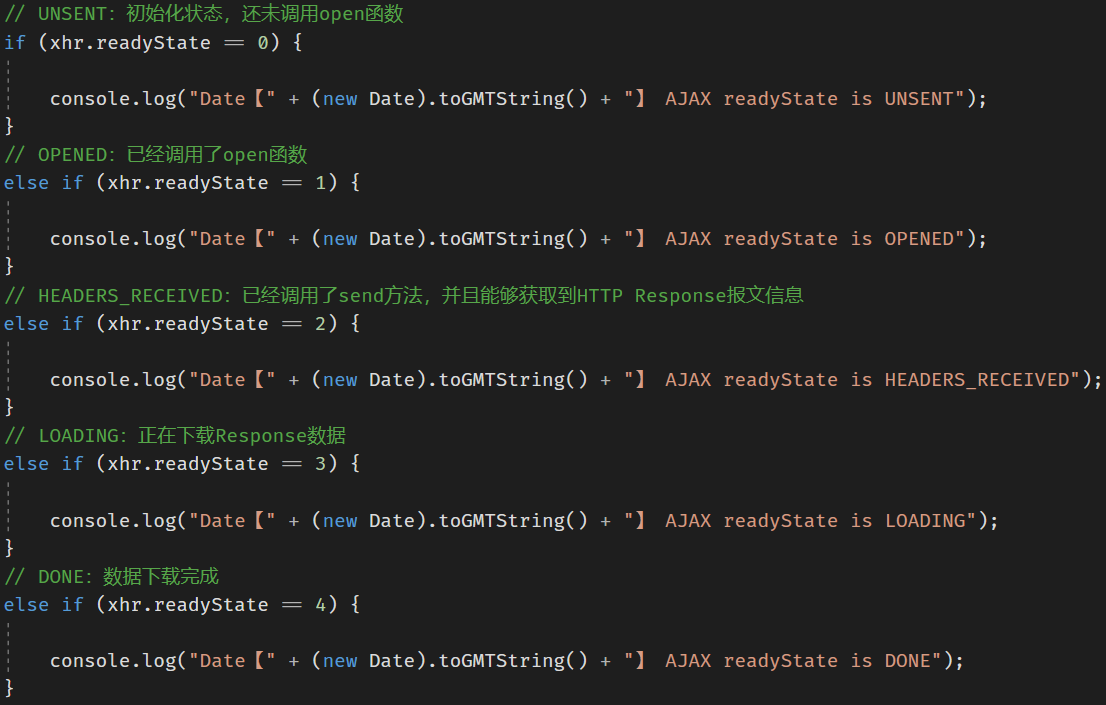
-> 1 OPENED：已经调用了open函数

-> 2 HEADERS\_RECEIVED：已经调用了send方法，并且能够获取到HTTP Response报文信息

-> 3 LOADING：正在下载Response数据

-> 4 DONE：数据下载完成

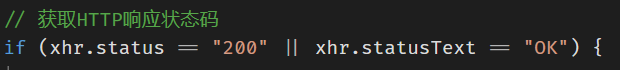
-> 该属性能够配合【onreadystatechange】属性为其指定一个用于监听状态改变的回调函数



1. status/statusText：该属性用于获取Response的HTTP状态信息

-> status：获取HTTP Resoponse状态码

-> statusText：获取HTTP Resoponse状态文本信息



1. response/responseText/responseXML：该属性用于获取Response中的响应正文的信息

-> response：该属性通常用于获取指定了【responseType】为"arraybuffer/blob/json"的响应正文信息

-> responseText：该属性通常用于获取指定了【responseType】为"text"的响应正文信息

-> responseXML：该属性通常用于获取指定了【responseType】为"document"的响应正文信息



1. XMLHttpRequest中的函数
2. getResponseHeader(string key)：获取指定键key的Http Response的头部数据

-> 该属性只能够用在成功下载完成Resoponse数据之后

截图20190620150403

1. getAllResponseHeaders()：获取所有的Http Response头部数据

-> 该属性只能够用在成功下载完成Resoponse数据之后

截图20190620150417

1. setRequestHeader(string key,string value)：设置HTTP Request头部字段

-> 该属性只能够用在open函数之后，send函数之前，否则会抛出异常



1. abort()：终止本次AJAX请求

-> 当该函数调用后后，所指定的【onabort】回调则会触发

截图20190620152133

1. open(string methodName,string url,bool isAsync,string userName,string userPwd)：开启AJAX引擎

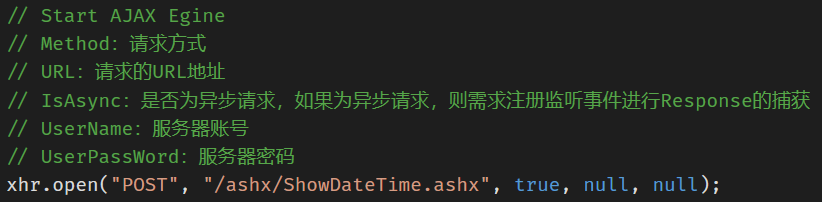
-> methhodName：当前发送的HTTP Request方法名，该属性不区分大小写

-> url：当前所发送的Http Request的URL地址，get请求的参数可以指定在该属性的url地址的结尾处([参考get请求](../../ASP.NET/HTTP协议/HTTP协议的详解.docx))

-> isAsync：指定当前AJAX引擎是否异步开启，当指定了该属性的值为true之后，我们必须要另外指定一些回调函数用于在服务器成功发送Response之后获取Response的数据

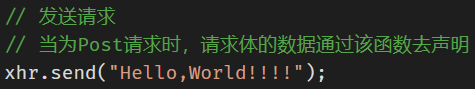
-> userName：服务器的账号名，如果没有则可以不指定该参数，其默认值为null

-> userPwd：服务器的密码，如果没有则可以不指定该参数，其默认值为null



1. send(var postData)：发送AJAX请求至服务器

-> 如果当前AJAX引擎所指定的请求方式为post，则请求体中的数据可以通过参数postData指定

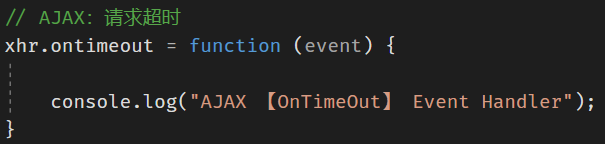


1. AJAX中的回调监听

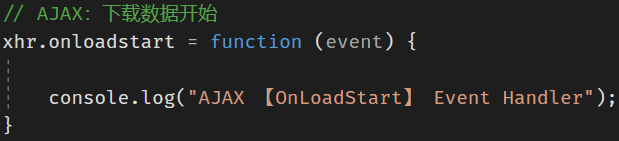
-> 在AJAX中监听回调主要体现在两个类型的实例上，第一种为XMLHttpRequest，第二种为XMLHttpReuqest.upload，它们都继承于【XMLHttpRequestEventTarget】接口，该接口提供了7种不同的回调函数，而XMLHttpRequest中的7种回调主要是集中在下载Response数据的过程当中所触发的，而XMLHttpRequest.upload的7种回调主要是集中在上传Request数据的过程当中所触发的

1. XMLHttpRequest：主要体现在下载Response数据过程当中的回调函数

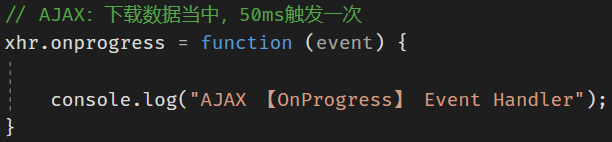
-> ontimeout：请求超时回调



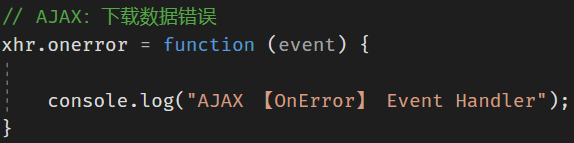
-> onloadstart：下载数据开始回调



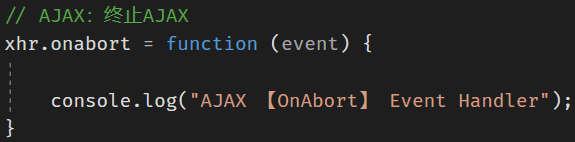
-> onprogress：下载数据过程当中回调，每50毫秒触发一次



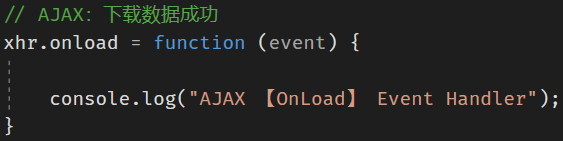
-> onerror：下载数据错误回调



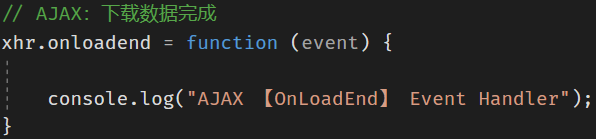
-> onabort：终止AJAX工作回调



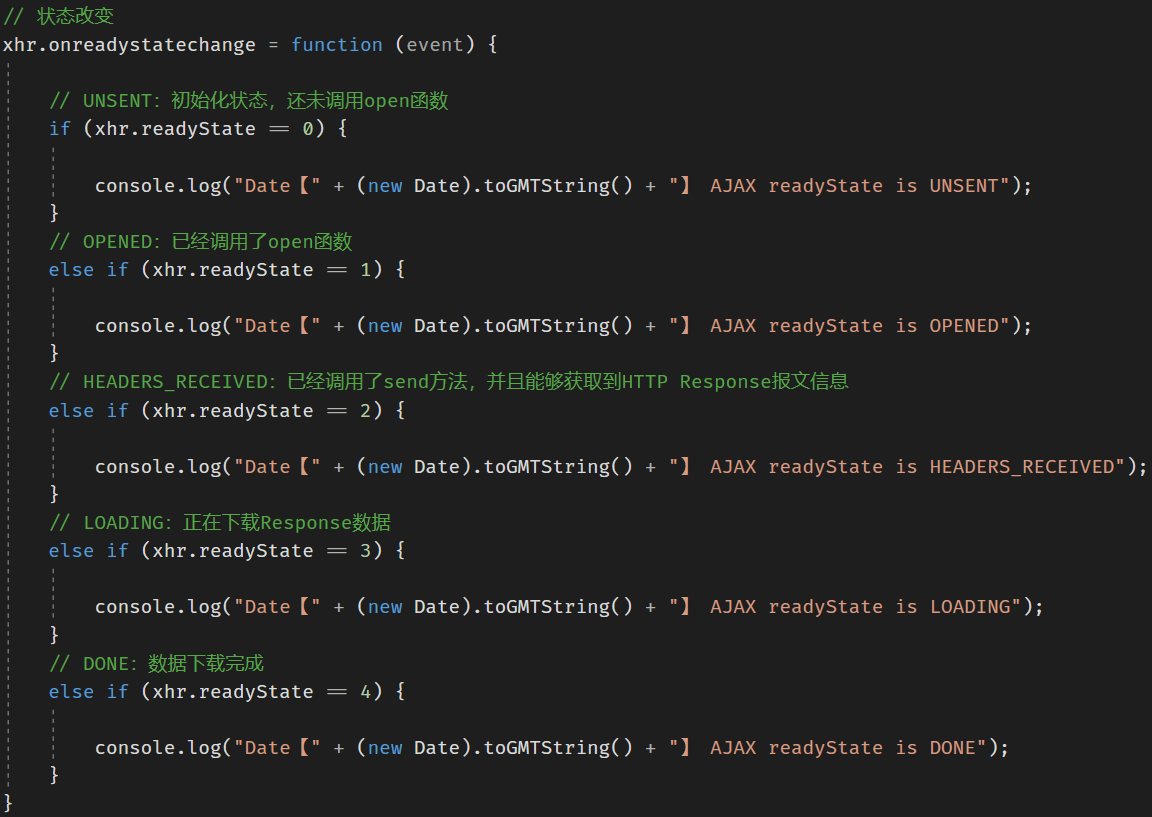
-> onload：下载数据成功回调



-> onloadend：下载数据完成回调

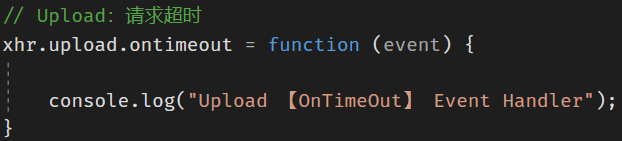


-> onreadystatechange：【readyState】属性改变后触发的回调

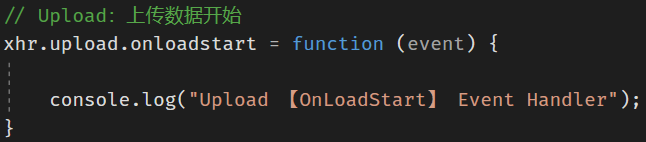


1. XMLHttpRequest.upload：主要体现在上传Request数据过程当中的回调函数

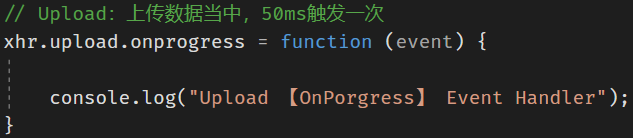
-> ontimeout：请求超时回调



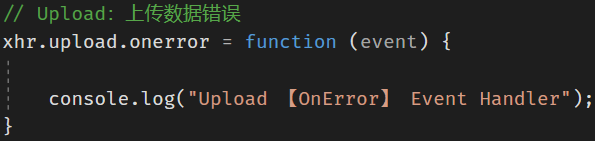
-> onloadstart：下载数据开始回调



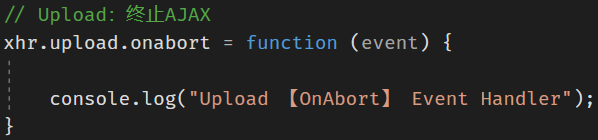
-> onprogress：下载数据过程当中回调，每50毫秒触发一次



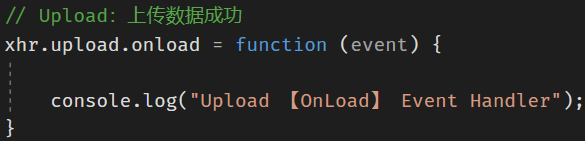
-> onerror：下载数据错误回调



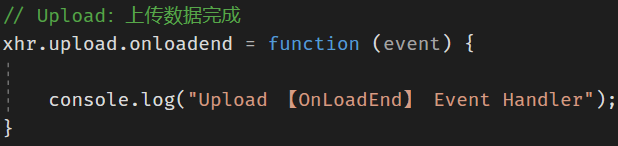
-> onabort：终止AJAX工作回调



-> onload：下载数据成功回调



-> onloadend：下载数据完成回调



1. 关于AJAX的具体使用可以参考[源代码管理](../../../源码/WEB应用程序/WEB应用程序.sln)