Ajax 基本原理和概念

什么是 Ajax

Ajax 是一种在无需重新加载整个网页的情况下，能够更新部分网页的技术。

Ajax 的全称是 Asynchronous JavaScript and XML，即异步 JavaScript+XML。它并不是新的编程语言，而是几种原有技术的结合体。它由以下几种技术组合而成，包括：

- HTML/XHTML——主要的内容表示语言。
- CSS——为 XHTML 提供文本格式定义。
- DOM——对已载入的页面进行动态更新。
- XML——数据交换格式。
- XSLT——将 XML 转换为 XHTML（用 CSS 修饰样式）。
- XMLHttp——用 XMLHttpRequest 来和服务器进行异步通信，是主要的通信代理。
- JavaScript——用来编写 Ajax 引擎的脚本语言。

实际上，在 Ajax 解决方案中这些技术都是可用的，不过只有三种是必须的：HTML/XHTML、DOM 以及 JavaScript。

XMLHttpRequest 对象

当需要异步与服务器交换数据时，需要 XMLHttpRequest 对象来异步交换。XMLHttpRequest 对象的主要属性有：

- onreadystatechange——每次状态改变所触发事件的事件处理程序。
- responseText——从服务器进程返回数据的字符串形式。
- responseXML——从服务器进程返回的 DOM 兼容的文档数据对象。
- status——从服务器返回的数字代码，如 404（未找到）和 200（已就绪）。
- status Text——伴随状态码的字符串信息。
- readyState——对象状态值。对象状态值有以下几个：
    - 0 - (未初始化)还没有调用 send()方法
    - 1 - (载入)已调用 send()方法，正在发送请求
    - 2 - (载入完成)send()方法执行完成
    - 3 - (交互)正在解析响应内容
    - 4 - (完成)响应内容解析完成，可以在客户端调用了

对于 readyState 的状态值，其中“0”状态是在定义后自动具有的状态值，而对于成功访问的状态（得到信息）我们大多数采用“4”进行判断。

Ajax 的核心就是是 JavaScript 对象 XmlHttpRequest，这个对象为向服务器发送请求和解析服务器响应提供了流畅的接口。XmlHttpRequest 可以使用 JavaScript 向服务器提出请求并处理响应，而不阻塞用户。

XHR 对象由 IE5 率先引入，在 IE5 中 XHR 对象是通过 MSXML 库中一个 ActiveX 对象实现的，根据 IE 版本不同可能会遇到不同版本 XHR 对象，而 IE7+与其它现代浏览器均支持原生的 XHR 对象，在这些浏览器中我们只需使用 XMLHttpRequest 构造函数就可以构造 XHR 对象，因此一个浏览器兼容的创建 XHR 对象的函数写法大概是这个样子：

```
var xmlhttp;

if (window.XMLHttpRequest) {

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

  xmlhttp=new XMLHttpRequest();

} else {

  // code for IE6, IE5

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

}
```

XMLHttpRequest 对象用法

XMLHttpRequest 对象有两个重要方法 open 与 send。

方法	描述
open(*method*,*url*,*async*)	规定请求的类型、URL 以及是否异步处理请求。 • *method*：请求的类型；GET 或 POST • *url*：文件在服务器上的位置 • *async*：true（异步）或 false（同步）
send(*string*)	将请求发送到服务器。 • *string*：仅用于 POST 请求

```
1 xmlhttp.open("GET","ajax_info.txt",true);2 xmlhttp.send();
```

对于 open 方法，有几点需要注意：

1. URL 是相对于当前页面的路径，也可以似乎用绝对路径。
2. open 方法不会向服务器发送真正请求，它相当于初始化请求并准备发送。
3. 只能向同一个域中使用相同协议和端口的 URL 发送请求，否则会因为安全原因报错。

真正能够向服务器发送请求需要调用 send 方法，并仅在 POST 请求可以传入参数，不需要则发送 null，在调用 send 方法之后请求被发往服务器。

请求发往服务器，服务器根据请求生成响应（Response），传回给 XHR 对象，在收到响应后相应数据会填充到 XHR 对象的属性，有四个相关属性会被填充：

- responseText——从服务器进程返回数据的字符串形式。
- responseXML——从服务器进程返回的 DOM 兼容的文档数据对象。
- status——从服务器返回的数字代码，如 404（未找到）和 200（已就绪）。
- status Text——伴随状态码的字符串信息。

在收到响应后第一步是检查响应状态，确保响应是否成功返回（状态为 200）。

```
xhr.open('get','default.aspx,false'); //准备同步请求

xhr.send();

if(xhr.status>=200 && xhr.status<300 || xhr.status==304){

    //do something
```

```
}else{

    //error handler

}
```

上面代码在发送同步请求的时候没问题，只有得到响应后才会执行检查 status 语句，但是在异步请求时，JavaScript 会继续执行，不等生成响应就检查状态码，这样我们不能保证检查状态码语句是在得到响应后执行（实际上也几乎不可能，服务器再快一个 HTTP 请求也不会快过一条 JavaScript 执行数度），这时候我们可以检查 XHR 对象的 readyState 属性，该属性表示请求/响应过程中的当前活动阶段，每当 readyState 值改变的时候都会触发一次 onreadystatechange 事件。

我们可以利用这个事件检查每次 readyState 变化的值，当为 4 的时候表示所有数据准备就绪，有一点我们需要注意：**必须在 open 方法之前指定 onreadtstatechange 事件处理程序。**

```
xmlhttp.onreadystatechange=function(){

  if (xmlhttp.readyState==4 && xmlhttp.status==200){

    document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

    }

  }

xmlhttp.open("GET","/try/ajax/ajax_info.txt",true);

xmlhttp.send();
```

GET 还是 POST?

与 POST 相比，GET 更简单也更快，并且在大部分情况下都能用。

然而，在以下情况中，请使用 POST 请求：

- 无法使用缓存文件（更新服务器上的文件或数据库）
- 向服务器发送大量数据（POST 没有数据量限制）
- 发送包含未知字符的用户输入时，POST 比 GET 更稳定也更可靠

一个简单的 GET 请求：

```
xmlhttp.open("GET","demo_get.html",true);

xmlhttp.send();
```

在上面的例子中，可能得到的是缓存的结果。

为了避免这种情况，向 URL 添加一个唯一的 ID：

```
xmlhttp.open("GET","demo_get.html?t=" + Math.random(),true);

xmlhttp.send();
```

如果希望通过 GET 方法发送信息，向 URL 添加信息：

```
xmlhttp.open("GET","demo_get2.html?fname=Henry&lname=Ford",true);

xmlhttp.send();
```

一个简单 POST 请求：

```
xmlhttp.open("POST","demo_post.html",true);

xmlhttp.send();
```

如果需要像 HTML 表单那样 POST 数据，请使用 setRequestHeader() 来添加 HTTP 头。然后在 send() 方法中规定您希望发送的数据：

```
xmlhttp.open("POST","ajax_test.html",true);

xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");

xmlhttp.send("fname=Henry&lname=Ford");
```

方法	描述
setRequestHeader(*header*,*value*)	向请求添加 HTTP 头。 • *header*: 规定头的名称 • *value*: 规定头的值

open() 方法的 url 参数是服务器上文件的地址：

```
xmlhttp.open("GET","ajax_test.html",true);
```

该文件可以是任何类型的文件，比如 .txt 和 .xml，或者服务器脚本文件，比如 .asp 和 .php（在传回响应之前，能够在服务器上执行任务）。

异步 - True 或 False?

XMLHttpRequest 对象如果要用于 AJAX 的话，其 open() 方法的 async 参数必须设置为 true：

```
xmlhttp.open("GET","ajax_test.html",true);
```

对于 web 开发人员来说，发送异步请求是一个巨大的进步。很多在服务器执行的任务都相当费时。AJAX 出现之前，这可能会引起应用程序挂起或停止。

通过 AJAX，JavaScript 无需等待服务器的响应，而是：

- 在等待服务器响应时执行其他脚本
- 当响应就绪后对响应进行处理

当使用 async=true 时，规定在响应处于 onreadystatechange 事件中的就绪状态时执行的函数：

```
xmlhttp.onreadystatechange=function(){
  if (xmlhttp.readyState==4 && xmlhttp.status==200){
    document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
    }
  }
xmlhttp.open("GET","ajax_info.txt",true);
xmlhttp.send();
```