

## 零、基础知识准备

---

### 0.1 B/S和C/S

---

什么是B/S和C/S

### 0.2 HTTP协议

---

请求和响应

### 0.3 GET请求和POST请求

---

GET请求和POST请求的区别

## 一、AJAX简介

---

传统的网页（不使用 AJAX）如果需要更新内容，必须重载整个网页页面，在加载新的网页的过程中，用户会有一个等待。降低了用户体验。

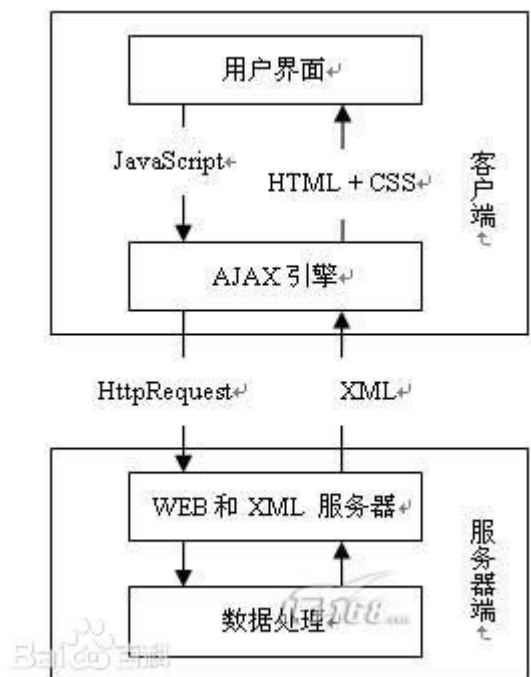
AJAX的出现就可以让网页进行部分更新。

Ajax的全称是Asynchronous JavaScript+XML，Ajax不是一个技术，它实际上是几种技术，每种技术都有其独特之处，合在一起就成了一个功能强大的新技术。Ajax结合了Java技术、XML以及JavaScript等编程技术，可以让开发人员构建基于Java技术的Web应用，并打破了使用页面重载的惯例。Ajax是使用客户端脚本与Web服务器交换数据的Web应用开发方法。这样，Web页面不用打断交互流程进行重新加载，就可以动态地更新。使用Ajax，用户可以创建接近本地桌面应用的直接、高可用、更丰富、更动态的Web用户界面。

Ajax刚出生的时候，用的比较多的数据格式是XML，后来JSON数据格式更多的去替换了XML格式的数据。

## 二、AJAX工作原理

---



### 三、使用AJAX

AJAX的核心对象是XMLHttpRequest, 这个对象在目前的所有浏览器都支持, 只是IE6 和 IE5的写法不一样而已。使用AJAX一般按照下面 4个步骤 使用即可。

#### 3.1 步骤1：创建XMLHttpRequest对象

这个对象从IE5开始支持, 后来各大浏览器都进行了跟进。IE7+和chrome、firefox写法一样, IE6以前的浏览器一种写法。实际开发中, 一般使用如下的兼容写法。

```
function createXHR() {  
    if(window.XMLHttpRequest) {  
        //标准浏览器创建XMLHttpRequest对象的方式  
        return new XMLHttpRequest();  
    }else{  
        //IE5和IE6的创建方式  
        return new ActiveXObject("Microsoft.XMLHTTP");  
    }  
}
```

#### 3.2 步骤2：调用open方法

```
/*调用open方法  
* 参数1: 请求方法, 一般是get或post  
* 参数2: 请求是url  
* 参数3: 是否为异步。true表示异步, 默认是true  
*/  
xhr.open("GET", "msg.json", true);
```

注意：

- URL相对于执行代码的当前页面（当然也可以使用绝对路径）；
- 调用 `open()` 方法并不会真正发送请求，而只是启动一个请求以备发送。
- 如果使用绝对路径,则一定要和当前页面的协议、主机和端口完全一致，否则会出现错误。

### 3.3 监听请求状态(`onreadystatechange`)

`xhr.onreadystatechange`监听请求状态，根据状态来确定数据获得之后要做的事情。

`xhr.readyState`的值保存了`xhr`的状态。一共有5种状态，每个状态用1个整数来表示

- 0 未初始化。尚未调用`open`方法
- 1 启动。已经调用`open`，但是还没有调用`send`方法
- 2 发送。已经`send`方法，但是还没有接收到相应
- 3 接收。已经开始接收数据，但是还没有完全接收。
- 4 完成。已经完全接收数据。

作为开发者，我们一般只关注 第4中状态。

```
//监听状态
xhr.onreadystatechange = function () {
    // 如readyState的值为4，表示已经接收完数据，可以开始对数据处理。
    if(xhr.readyState == 4){
        // xhr.status 保存了服务的响应码， 200表示正常响应
        if(xhr.status == 200){
            alert(xhr.responseText);
        }
    }
}
```

### 3.4 调用`send`方法发送请求

```
//send方法才是真的发起网络请求。 参数：请求的时候想服务区传递的参数。 如果是get请求，直接传入null即可。
xhr.send(null);
```

## 四、使用AJAX是携带参数

不同的请求方式，携带参数的方式不一样。

### 4.1 GET请求

get请求携带的参数直接追加到url后面即可。?后面跟的就是url。

```
var url = http://wthrcdn.etouch.cn/weather_mini?city=深圳&
```

### 4.2 POST请求

post请求的参数是在send方法中携带，参数的格式必须是：xxx=XXX&yyy=YYY格式。注意，参数中的中文必须要经过url编码。

```
xhr.send('user='+encodeURIComponent(nameVal) + '&pass=' + passVal);
```

## 五、同源策略

### 5.1 什么是同源策略

同源策略，它是由Netscape提出的一个著名的安全策略，现在所有的可支持javascript的浏览器都会使用这个策略。最初，它的含义是指，A网页设置的Cookie，B网页不能打开，除非这两个网页"同源"。所谓"同源"指的是"三个相同"。

1. 协议相同
2. 域名相同
3. 端口号相同

例如：<http://www.yztcedu.com/index.html>

协议：http

域名：[www.yztcedu.com](http://www.yztcedu.com)

端口号：80(省略的时候默认是80)

<http://www.yztcedu.com/another.html> 同源

<https://www.yztcedu.com/a.html> 不同源 因为协议不同

<http://www.yztc.com/a.html> 不同源 因为域名不同

<http://www.yztcedu.com:8080/a.html> 不同源 因为端口号不同。

要想同源，三个必须完全相同。

### 5.2 同源策略的目的

同源策略的目的，是为了保证用户信息的安全，防止恶意的网站窃取数据。

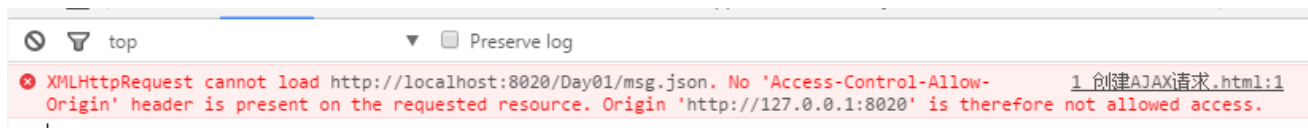
### 5.2 同源策略的限制范围

随着互联网的发展，"同源策略"越来越严格。目前，如果非同源，共有三种行为受到限制。

- (1) Cookie、LocalStorage 和 IndexDB 无法读取。
- (2) DOM 无法获得。
- (3) AJAX 请求不能发送。

## 六、AJAX中规避同源策略

在AJAX中，同源策略要求，AJAX只能发给同源的网址，否则就报错。



## 使用JSONP规避同源

JSONP是服务器与客户端跨源通信的常用方法。最大特点就是简单适用，老式浏览器全部支持，服务器改造非常小。

它的基本思想是，网页通过添加一个 `script` 元素，向服务器请求 JSON 数据，这种做法不受同源政策限制；服务器收到请求后，将数据放在一个指定名字的回调函数里传回来。

```
//函数功能：添加script标签。 参数表示AJAX要请求的地址(一个外部的网页)。  
function addScriptTag(src) {  
    var script = document.createElement('script');  
    script.setAttribute("type", "text/javascript");  
    script.src = src;  
    document.body.appendChild(script);  
}  
//请求参数中添加一个callback参数，用来指定回调函数的名字，这对于JSONP是必须的。  
window.onload = function () {  
    addScriptTag('http://localhost:8020?callback=foo');  
}  
//回调函数。 一旦响应成功，会执行该方法  
function foo(data) {  
    alert(data);  
    console.log(data);  
};
```

## 七、AJAX的数据解析