1. 概念

Asynchronous JavaScript and XML 异步的js和XML

在不重新加载整个页面的情况下，可以与服务器交互数据并更新部分网页内容

1. XMLHttpRequest对象

所有现代浏览器均支持XMLHttpRequest对象，ie5 ie6使用ActiveObject

XMLHttpRequest用于在后台与服务器交互数据

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

var xmlhttp;

if (window.XMLHttpRequest) {

// IE7+, Firefox, Chrome, Opera, Safari 浏览器执行代码 xmlhttp=new XMLHttpRequest();

}

else {

// IE6, IE5 浏览器执行代码

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

}

1. 向服务器发送请求

xmlhttp.open("GET","ajax\_info.txt",true);  
xmlhttp.send();

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文件在服务器的位置

async true-异步 false-同步

send(str) 将请求发送到服务器

str:仅用于POST请求

使用POST的情况：

* 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

get请求：

xmlhttp.open("GET","/try/ajax/demo\_get.php",true);

xmlhttp.send();

给url加一个唯一的ID

xmlhttp.open("GET","/try/ajax/demo\_get.php?t=" + Math.random(),true); xmlhttp.send();

通过get方法发送消息 安全性低 通过url传递数据 适合获取(如浏览帖子) 会缓存

xmlhttp.open("GET","/try/ajax/demo\_get2.php?fname=Henry&lname=Ford",true);

xmlhttp.send();

post请求 适合长传(如注册)

xmlhttp.open("POST","/try/ajax/demo\_post.php",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","/try/ajax/demo\_post2.php",true); xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");

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

1. 服务器响应

如需获得来自服务器的响应，请使用 XMLHttpRequest 对象的 responseText 或 responseXML 属性

responseText 获得字符串形式的响应数据

responseXML 获得XML形式的响应数据

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

xmlDoc=xmlhttp.responseXML;

txt="";

x=xmlDoc.getElementsByTagName("ARTIST");

for (i=0;i<x.length;i++) {

txt=txt+x[i].childNodes[0].nodeValue+"<br>";

}

document.getElementById("myDiv").innerHTML=txt;

1. onreadystatechange事件

每当 readyState 改变时，就会触发 onreadystatechange 事件

readyState 属性存有 XMLHttpRequest 的状态信息

* 0: 请求未初始化
* 1: 服务器连接已建立
* 2: 请求已接收
* 3: 请求处理中
* 4: 请求已完成，且响应已就绪

status 200-OK 404-未找到页面

当 readyState 等于 4 且状态为 200 时，表示响应已就绪

xmlhttp.onreadystatechange=function() {

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

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

}

}

1. 使用回调函数

回调函数是一种以参数形式传递给另一个函数的函数

var xmlhttp;

function loadXMLDoc(url,cfunc)

{

if (window.XMLHttpRequest)

{// IE7+, Firefox, Chrome, Opera, Safari 代码

xmlhttp=new XMLHttpRequest();

}

else

{// IE6, IE5 代码

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

}

xmlhttp.onreadystatechange=cfunc;

xmlhttp.open("GET",url,true);

xmlhttp.send();

}

function myFunction()

{

loadXMLDoc("/try/ajax/ajax\_info.txt",function()

{

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

{

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

}

});

}

1. 安全限制

默认情况下，js在发送ajax请求时，url的域名必须和当前页面一致。

域名要相同 [www.example.com和example.com](http://www.example.com和example.com)不同

协议要相同 http和https不同

端口号相同 :80和:8080不同

1. 跨域
2. flash
3. 代理服务器
4. JSONP 只能用于get请求，且返回js
5. CORS