

AJAX (一)

主要内容

- Ajax简介
- AJAX的四个主要组件
- XMLHttpRequest对象
- 发送简单请求步骤

Ajax简介

- Asynchronous JavaScript+XML
- AJAX不是一个新的技术，应算一个技巧
- JavaScript是其主要组件
- XMLHttpRequest对象(XHR)是AJAX中的重要对象
- 因特网默认的请求/响应模式有了重大转变，这正式AJAX的核心
- Web应用开发人员可以自由地与服务器异步交互

AJAX概念

- AJAX是一个客户端技术，不论使用何种服务器端技术，都能使用AJAX
- 需要注意的是，使用AJAX，服务器端程序可能不会返回一个完整页面，而只是返回一点点文本

AJAX的四个主要组件

- Ajax的主要组件
 - JavaScript
 - CSS
 - DOM
 - XMLHttpRequest对象

XMLHttpRequest对象

- XMLHttpRequest最早是在IE5中以ActiveX组件形式实现的
- 最近，很多新的浏览器都支持XHR
- XMLHttpRequest对象是整个Ajax开发的基础，提供了客户端和服务端进行异步通信的能力。一方面它向服务器提交一个请求，获取指定页面的内容；另一个方面将指定的数据提交到服务器。
- XMLHttpRequest对象由JavaScript创建并使用，客户端可以只从服务器获取需要的信息，通过与DOM和CSS的结合，可以实现局部刷新。

使用XMLHttpRequest对象和服务端进行通信

-- 创建

~~XMLHttpRequest~~对象

- 使用JavaScript创建一个XHR对象
- IE把XHR实现为一个ActiveX对象
- 其他浏览器实现为一个本地JavaScript对象
- 由于这些差别，所以创建XHR时需要用分支逻辑
- 具体实现方式：

```
var xmlHttp;  
function createXMLHttpRequest() {  
    if(window.ActiveXObject) {  
        xmlHttp=new ActiveXObject( "Microsoft.XMLHttp" );  
    }else if(window.XMLHttpRequest) {  
        xmlHttp=new XMLHttpRequest();  
    }  
}
```

使用open方法创建一个请求

- 创建完XMLHttpRequest对象后就可以开始使用它了, open方法可以创建一个请求, 并准备向服务器发送.
- `open(“method”, ” url”)`: 建立对服务器的调用, method可以是get、post.
- 例如:
 - `xhr.open(“get” ”, jax.jsp?name=“John”);`
 - `Xhr.open(“post”, jax.jsp);`

使用send方法发送一个请求

- 创建一个请求以后,可以使用send方法向服务器发送这个请求.
- `send(body);`
- 如果使用get创建的请求,`body=null`;
 - 例如:`send(null);`
- 如果使用post创建的请求,`body`是要传送的是数据.
 - 例如:`send(name= “John”);`
 - 注意:使用post,那么在send之前必须使用下面的语句
 - `xmlHttp.setRequestHeader(“Content-Type”, ” application/x-www-form-urlencoded”)`

使用onreadystatechange事件捕获请求的状态变化

- 在发送一个请求后,往往无法知道请求什么时候完成,所以必须利用事件机制来捕获请求状态,XMLHttpRequest对象提供了onreadystatechange事件实现这一功能.
- onreadystatechange:每个状态改变时都会触发这个事件处理器,通常会调用一个JavaScript函数
- 例如:
 - `xhr. onreadystatechange = processResponse;`
 - `xhr. open (...);`
 - `xhr. send (...);`
 - 注意:这里将事件绑定在调用open和send方法之前进行,是因为这两个方法都会触发onreadystatechange 事件. 如果这些事件定义在其后可能引起事件处理代码无法得到执行.

使用readyState属性判断请求状态

- readyState: 请求的状态。在事件处理中可以根据这个值来进行不同的处理. 下表介绍了各个值的含义.
 - readyState值的含义
 - readyState值
 - 0 含义: 对象已经建立, 但还未初始化, 这是尚未调用open方法
 - 1 含义: 表示open方法已经被调用, 但尚未调用send方法
 - 2 含义: 表示send方法已经调用, 其他数据未知
 - 3 含义: 表示请求已经发送成功, 正在接收数据
 - 4 含义: 表示数据已经接收成功.
 - 在整个请求过程中, onreadystatechange事件在每一次readyState值改变时都会被触发. 通常在事件中判断readyState的值是在请求完毕是才作处理. 例如:
 - ```
function processResponse(){
 if(xhr.readyState==4){请求完毕是的处理代码}
}
```

# 使用status属性判断请求的结果

- status存储服务器端返回的Http请求响应的代码,它表示请求的处理结果.下面列出了常见的响应代码的含义.
- Http请求响应代码的含义
  - Http状态码                      含义
  - 200                      请求成功
  - 404                      请求资源未找到
  - 400                      错误的请求
  - 500                      内部服务器错误
  - 在Ajax的开发中,最常用的是200这个响应代码.

例如:

```
function processResponse(){
 if(xhr.readyState==4){
 if(xhr.status==200){请求成功完成;}
 else{网络失败或者服务器错误;}
 }
}
```

# 使用responseTest属性获取文本文档

---

- 当服务器成功处理了请求并返回后,可以利用XMLHttpRequest对象的responseText属性获取返回的结果.

# 代码具体实现(1)

---

- 初始化XMLHttpRequest
- var xhr=false;
- function processXMLHttpRequest(){
  - if(window.ActiveXObject){  
    **xhr= new ActiveXObject("Microsoft.XMLHttp");**
  - }else if(window.XMLHttpRequest){  
    **xhr=new XMLHttpRequest();**
  - }
- }

# 代码具体实现(2)

- 验证form表单的方法:
- ```
function ss(){  
    - var uname=document.forms["myform"].uname.value;  
    - var psw = document.forms["myform"].psw.value;  
    - var psw2=document.forms["myform"].psw2.value;  
    - if(uname==""){  
        window.alert("用户名不能为空。")  
        document.myform.uname.focus();  
        return false;  
    - }  
    - else if(psw!=psw2){  
        window.alert("两次输入密码不同。")  
        document.myform.psw.focus();  
        return false;  
    - }else{  
        sendRequest("getup?uname="+uname+"&psw="+psw);  
    - }  
• }
```

代码具体实现(3)

- `function sendRequest(url){`
 - `processXMLHttpRequest();`
 - `xhr.onreadystatechange = processResponse;`
 - `xhr.open("get",url,true);`
 - `xhr.send(null);`
- `}`

代码具体实现(4)

- processResponse()方法
- function processResponse(){
 - if(xhr.readyState==4){//请求的状态完成
 - if(xhr.status==200){//响应成功
 - var res=xhr.responseText;//以文本的方式接收数据
 - window.alert(res);
 - //window.alert(xhr.responseXML.getElementsByTagName("res")[0].childNodes[0].nodeValue);
 - }else{
 - window.alert("你所请求的页面有异常。");
 - }
 - }
- }

代码具体实现(5)

- 服务器的处理代码
- `response.setContentType("text/html;charset=UTF-8");`
- `PrintWriter out = response.getWriter();`
- `if(name.equals("wangying")){`
 - `out.println("验证成功!");`
- `}else{`
 - `out.println("此用户不存在! ");`
- `}`
- `out.close();`

End Thanks