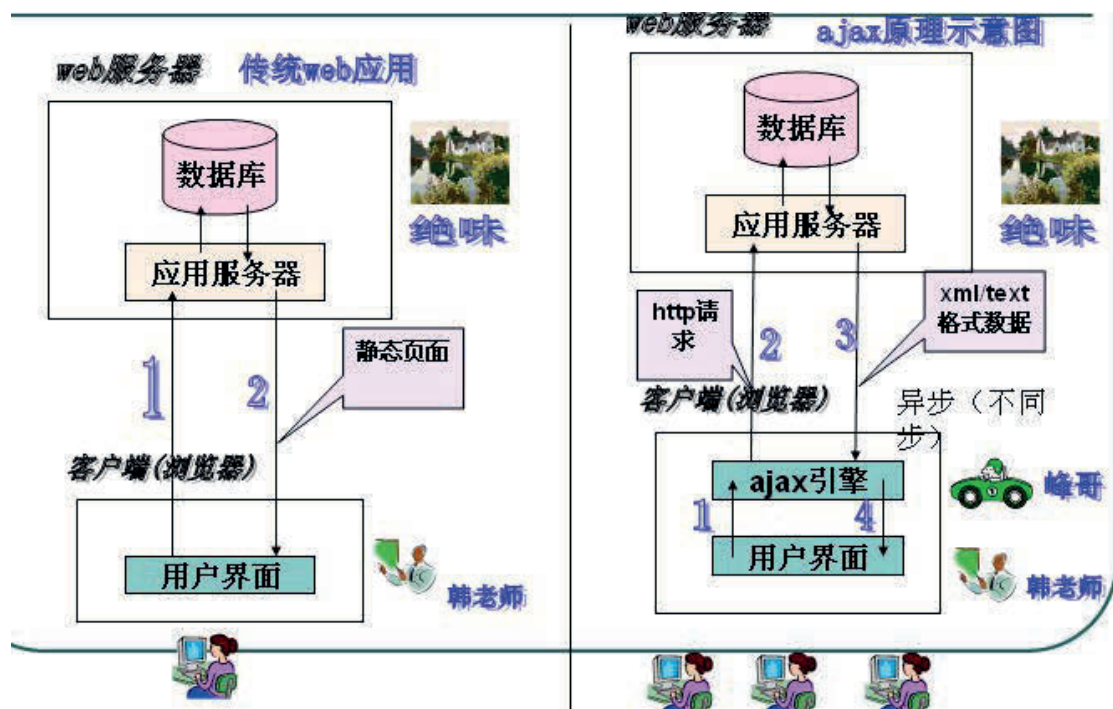


ajax 是什么 ?

- ① ajax(asynchrone javascript and xml) 异步的 javascript 和 xml
- ② 是 7 种技术的综合,它包含了七个技术( javascript xml xstl xhtml dom xmlhttprequest , css), ajax 是一个粘合剂,
- ③ ajax 是一个与服务端语言无关的技术. 即可以使用在(PHP/Java EE/.NET 网站/ ASP)
- ④ ajax 可以给客户端返回三种格式数据(文本格式 , xml , json 格式)
- ⑤ 无刷新数据交换技术有以下: flash, java applet, 框架, iframe, ajax)

◆ ajax 的运行原理分析



看一个需求:

◆ ajax 在什么地方用的多

- 1 动态加载数据, 按需取得数据。【树形菜单、联动菜单.../省市联动】
- 2 改善用户体验。【输入内容前提示、带进度条文件上传...】
- 3 电子商务应用。 【购物车、邮件订阅...】
- 4 访问第三方服务。 【访问搜索服务、rss 阅读器】
5. 数据的布局刷新

◆ 经典的案例

使用 ajax 与服务器通信的步骤

- ① 创建一个 XMLHttpRequest 对象
- ② 创建 url,data,通过 xmlhttprequest.send()
- ③ 服务器端接收 ajax 的请求,做相应处理(操作数据库), 然后返回结果(echo 语句)
- ④ 客户端通过 xmlhttprequest 的属性 responseText , responseXML 取的数据, 然后就完成局部刷新当前页面任务

## 1. 使用 ajax 完成用户名的验证

register.php

```
<html>
<head>
<title>用户注册</title>
<meta http-equiv="content-type" content="text/html;charset=utf-8"/>
<script type="text/javascript">

    //创建 ajax 引擎
    function getXmlHttpRequestObject(){

        var xmlhttpRequest;
        //不同的浏览器获取对象 xmlhttprequest 对象方法不一样
        if(window.ActiveXObject){

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

        }else{

            xmlhttpRequest=new XMLHttpRequest();
        }

        return xmlhttpRequest;
    }
    var myXmlHttpRequest="";

    //验证用户名是否存在
    function checkName(){

        myXmlHttpRequest=getXmlHttpRequestObject();
```

```

//怎么判断创建 ok
if(myXmlHttpRequest){

    //通过 myXmlHttpRequest 对象发送请求到服务器的某个页面
    //第一个参数表示请求的方式, "get" / "post"
    //第二个参数指定 url,对哪个页面发出 ajax 请求(本质仍然是 http 请
求)
    //第三个参数表示 true 表示使用异步机制,如果 false 表示不使用异
步

    var                url="/ajax/registerProcess.php?mytime="+new
Date()+"&username="+$("username").value;
    //打开请求.
    myXmlHttpRequest.open("get",url,true);
    //指定回调函数.chuli 是函数名
    myXmlHttpRequest.onreadystatechange=chuli;

    //真的发送请求,如果是 get 请求则填入 null 即可
    //如果是 post 请求, 则填入实际的数据
    myXmlHttpRequest.send(null);

}
}

//回调函数
function chuli(){

    //window.alert("处理函数被调回"+myXmlHttpRequest.readyState);
    //我要取出从 registerPro.php 页面返回的数据
    if(myXmlHttpRequest.readyState==4){

        //取出值,根据返回信息的格式定.text
        //window.alert("服务器返回"+myXmlHttpRequest.responseText);

        $('myres').value=myXmlHttpRequest.responseText;
    }
}

//这里我们写一个函数
function $(id){
    return document.getElementById(id);
}
</script>
</head>

```

```

<body>
  <form action="???" method="post">
    用户名 :<input type="text" onkeyup="checkName();"
name="username1" id="username"><input type="button"
onclick="checkName();" value="验证用户名">
    <input style="border-width: 0;color: red" type="text" id="myres">
    <br/>
    用户密码:<input type="password" name="password"><br>
    电子邮件:<input type="text" name="email"><br/>
    <input type="submit" value="用户注册">
  </form>
  <form action="???" method="post">
    用户名:<input type="text" name="username2" >
    <br/>
    用户密码:<input type="password" name="password"><br>
    电子邮件:<input type="text" name="email"><br/>
    <input type="submit" value="用户注册">
  </form>
</body>
</html>

```

registerpro.php

```
<?php
```

```

//这里两句话很重要,第一讲话告诉浏览器返回的数据是 xml 格式
header("Content-Type: text/xml;charset=utf-8");
//告诉浏览器不要缓存数据
header("Cache-Control: no-cache");

//接收数据
$username=$_GET['username'];

if($username=="shunping"){
    echo "用户名不可以用";//注意, 这里数据是返回给请求的页面.
}else{
    echo "用户名可以用";
}

```

?>

### ◆ ajax 的 post 方式请求

在前面案例我们修改一下：

关键代码

register.php

```
var url="/ajax/registerProcess.php";
    //这个是要发送的数据
    var data="username="+$('username').value;
    //打开请求.
    myXmlHttpRequest.open("post",url,true);
    //还有一句话，这句话必须.
    myXmlHttpRequest.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
    //指定回调函数.chuli 是函数名
    myXmlHttpRequest.onreadystatechange=chuli;

    //真的发送请求,如果是 get 请求则填入 null 即可
    //如果是 post 请求，则填入实际的数据
    myXmlHttpRequest.send(data);
registerPro.php 关键码：
$username=$_POST['username'];
```

☞ 使用 get 方式发出请求，如果提交的用户名不变化，浏览器将不会真的发请求，而是缓存取数据., url

解决方法

1. url 后带一个总是变化的参数,比如当前时间

```
var url="/ajax/registerProcess.php?mytime="+new Date()+"&username="+$("username").value;
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/085122121231011310>