1：Ajax 入门
2：XHR 对象
3：JSON/XML 双向解析、使用 jquery 解析 XML 文件
4：封装 Ajax 的实现。
5：Jquery with ajax method

Post/get

Post – 只有 <form method=”post”/>
其他的所有请求都是 get
上面的请求，都是基于同步的[阻塞的]

所有浏览器，都支持一个组件， XMLHttpRequest – 脚本对象。就是用 js 创建的对象。

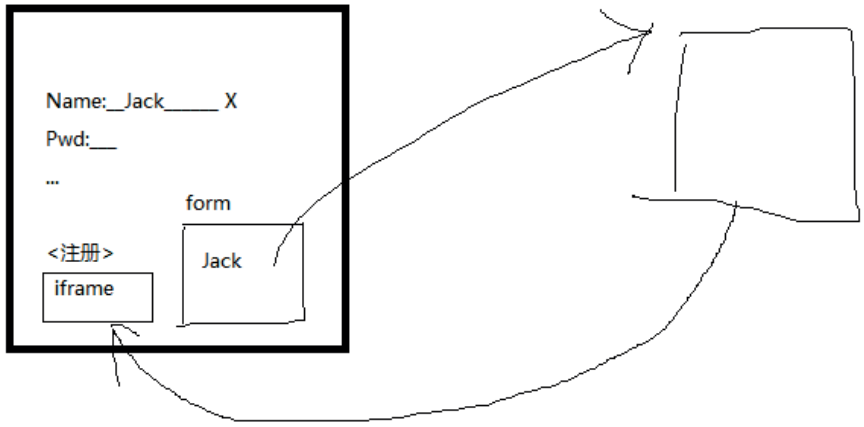
1：ajax -> Asynchronized Javacript And XML –

用 XMLHttpRequest 对象在 HTTP 协议上传递 XML 数据。(json)
组成部分：

1：核心对象[xhr] XMLHttpRequest。
2：操作 XHR 对象的脚本 JS
3：样式 CSS
4：数据传递时在页面上进行数据封装。XML – JSON – String

功能：
1：页面无刷新的局部更新。
2：异步，不阻塞。

2：非异步的示例，验证用户名是否存在

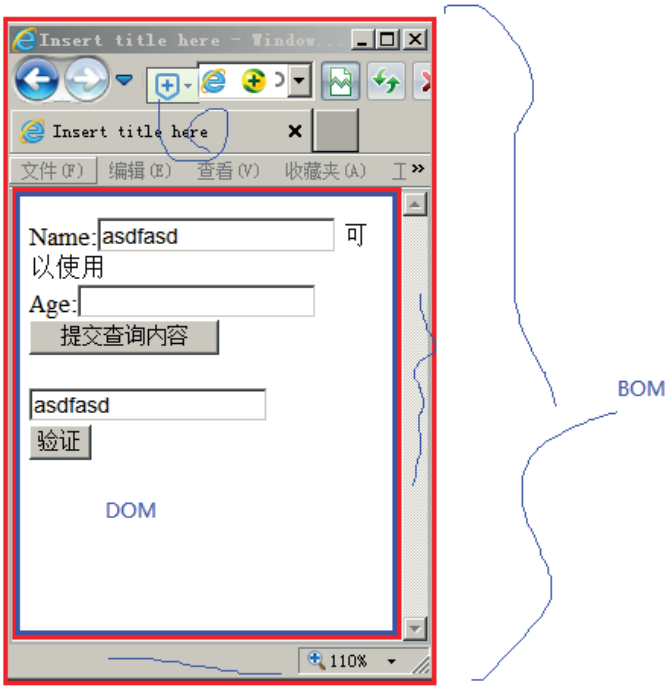



```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    <form name="f1" method="post" action="reg">
        Name:<input type="text" onblur="_blur(this);" name="name"> <label
            id="msg"></label> <br> Age:<input type="text" name="age"><br>
        <input type="submit">
    </form>
    <form name="f2" method="post" target="dataFrame" action="name">
        <div id="dataDiv" style="display:none;"></div>
```

```
    </form>
    <iframe name="dataFrame" style="display:none;"></iframe>
</body>
<script type="text/javascript">
    function _blur(obj) {
        var nm = obj.value;
        if (nm != "") {
            document.getElementById("dataDiv").innerHTML = "<input type='text'
name='name' value='"+nm+"'/>";
            document.forms["f2"].submit();
        }
    }

    function _back(mm) {
        if (mm == "1") {
            document.getElementById("msg").innerHTML = "请换一个新的";
        } else {
            document.getElementById("msg").innerHTML = "可以使用";
        }
    }
</script>
</html>


        //BOM - Browser Object Model - 浏览器对象模型 window
            //DOM - Document Object Model - 文档对象模型 - document
```



3：创建 ajax 的核心对象 XHRHttpRequest

在 IE4,5,6 – 创建 XHR 对象：

Var xhr = new ActiveXObject(“Micorsoft.XMLHttp”);

在其他的所有浏览器上，包含 IE 7 以上的版本：

Var xhr = new XMLHttpRequest();

```
<script type="text/javascript">
    var xhr = null;
    //判断是否拥有 XMLHttpRequsst 对象，即是否是 IE7,及其他的浏览器
    if(window.XMLHttpRequest){
        console.log("IE7,chrome,FF...");
        xhr = new XMLHttpRequest();
    }else{
        console.log("IE6....");
        xhr = new ActiveXObject("Micorsoft.XMLHttp");
    }
    alert(xhr);
```

批注 [H1]: 浏览器插件 OCX 组件。

```
</script>
```

以下是在 IE 浏览上面的创建方式：

```
<html>
  <script type="text/javascript">
        var xhr = null;
        var xhrs = ["Msxml2.XMLHTTP.6.0","Msxml2.XMLHTTP.5.0","Msxml2.XMLHTTP.4.0",
        "Msxml2.XMLHTTP.3.0","Msxml2.XMLHTTP","Microsoft.XMLHttp"];

              for(var i=0;i<xhrs.length;i++){
                 try{
                    xhr = new ActiveXObject(xhrs[i]);
                    break;
                 }catch(e){
                     continue;
                 }
              }
              alert("创建成功:"+i);
              alert(xhr);

  </script>
</html>
```

协议：

Post 的默认的请求的类型：

```
  POST /20151018/name HTTP/1.1
Host: localhost:8080
Connection: keep-alive
Content-Length: 13
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,imag
Origin: http://localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36
HTTPS: 1
Content-Type: application/x-www-form-urlencoded
Referer: http://localhost:8080/20151018/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: JSESSIONID=575AAFDAB8E1E62DAE01BAC6DE90E913
```

如果设置了表单的类型：

```
body>
  <form name="f1" method="post" action="reg" enctype="multipart/form-data">
      Name:<input type="text" onblur="_blur(this);" name="name"> <label
          id="msg"></label> <br> Age:<input type="text" name="age"><br>
      <input type="submit">
  </form>
```

```
▼ Request Headers    view parsed
  POST /20151018/reg HTTP/1.1
Host: localhost:8080
Connection: keep-alive
Content-Length: 237
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Origin: http://localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2438
HTTPS: 1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarytG5skKwvWzi5qNg9
Referer: http://localhost:8080/20151018/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: JSESSIONID=575AAFDAB8E1E62DAE01BAC6DE90E913
```

Boundary = --------ssfsdfsfs

```
HTTPS: 1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarytG5skKwvWzi5qNg9
Referer: http://localhost:8080/20151018/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: JSESSIONID=575AAFDAB8E1E62DAE01BAC6DE90E913
▼ Request Payload
------WebKitFormBoundarytG5skKwvWzi5qNg9
Content-Disposition: form-data; name="name"

asdfasdf
------WebKitFormBoundarytG5skKwvWzi5qNg9
Content-Disposition: form-data; name="age"

asdf
------WebKitFormBoundarytG5skKwvWzi5qNg9--
```

4：xhr 对象的一些方法属性