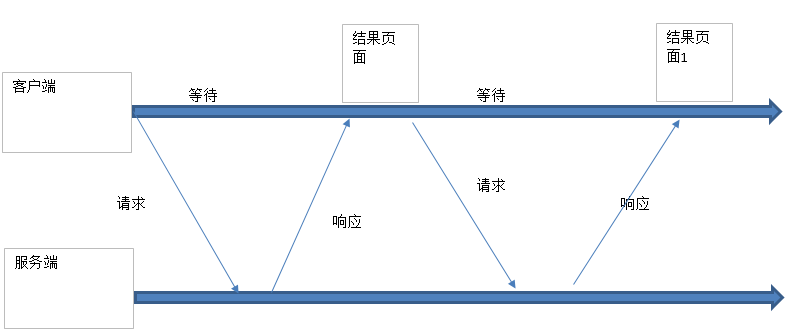
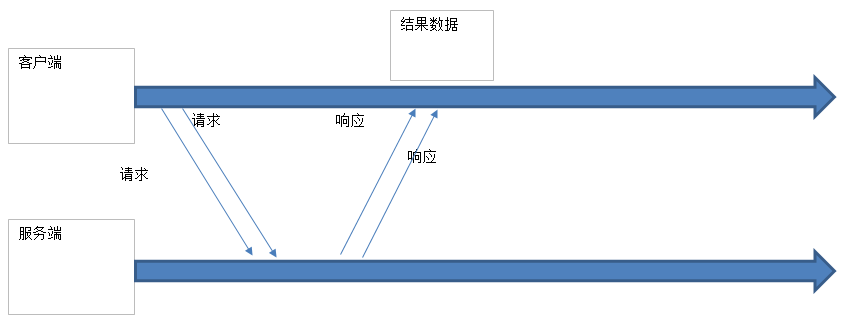
Ajax和json

# ajax



Ajax的特点:页面无刷新的请求，ajax默认是异步的，速度快，用户体验比较好



## ajax开发步骤

1. 创建XMLHttpRequest对象

|  |
| --- |
| **function** ajaxFunction(){  **try**{  // Firefox, Opera 8.0+, Safari  xmlHttp=**new** XMLHttpRequest();  }**catch** (e){  // Internet Explorer  **try** {  xmlHttp=**new** ActiveXObject("Msxml2.XMLHTTP");  }**catch** (e){  **try**{  xmlHttp=**new** ActiveXObject("Microsoft.XMLHTTP");  }**catch** (e){  alert("您的浏览器不支持AJAX！");  **return** **false**;  }  }  }  } |

2.将状态触发器绑定到一个函数

3.使用open方法建立与服务器的连接

|  |
| --- |
| **var** url = "/ajax\_demo2/valid?username="+obj.value;  //打开连接,第一个参数是请求方式，第二个参数是请求地址，第三个参数是同步还是异步，true同步，false是异步  xmlHttp.open('get', url, **false**); |

4.向服务器端发送数据

|  |
| --- |
| //发送数据,如果是get请求数据是在url上传递，不需要发送数据,send必须在最后调用。  xmlHttp.send(**null**); |

5.在回调函数中对返回数据进行处理

|  |
| --- |
| //接受服务器 的响应  xmlHttp.onreadystatechange = \_callback;  //发送数据,如果是get请求数据是在url上传递，不需要发送数据,send必须在最后调用。  xmlHttp.send(**null**); |

|  |
| --- |
| **function** \_callback(){  //readyState==4请求完成  **if**(xmlHttp.readyState == 4){  //服务器响应正确  **if**(xmlHttp.status == 200){  //接收服务器的响应数据  **var** result = xmlHttp.responseText;  alert(result);  }  }  } |

Ajax的post请求



同步 异步

Ajax返回xml文档

Servlet

|  |
| --- |
| response.setContentType("text/xml;charset=UTF-8");  **if**("zhangsan".equals(username)){  response.getWriter().print("<result>error</result>");  }**else**{  response.getWriter().print("<result>ok</result>");  } |

Ajax的回调

|  |
| --- |
| //readyState==4请求完成  **if**(xmlHttp.readyState == 4){  //服务器响应正确  **if**(xmlHttp.status == 200){  //接收服务器的响应数据,返回文档对象  **var** result = xmlHttp.responseXML;  **var** rs = result.getElementsByTagName('result');  alert(rs[0].innerHTML);  }  } |

# json数据

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。 易于人阅读和编写。同时也易于机器解析和生成。 这些特性使JSON成为理想的数据交换语言。

# 三级联动下拉框

省 市 区