# 1 Ajax

## 1.1 ajax简介

* AJAX 是一种在无需重新加载整个网页的情况下，能够更新部分网页的技术。
* AJAX = 异步 JavaScript 和 XML。
  + 同步：
  + 异步：
* AJAX 是一种用于创建快速动态网页的技术。
* 通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。
* 传统的网页（不使用 AJAX）如果需要更新内容，必需重载整个网页面。
* 有很多使用 AJAX 的应用程序案例：新浪微博、Google 地图、开心网等等。
* **演示：搜索引擎就是通过ajax来实现的**

## 1.2 XMLHttpRequest

### XMLHttpRequest对象

* XMLHttpRequest 是 AJAX 的基础，所有现代浏览器均支持 XMLHttpRequest 对象（IE5 和 IE6 使用 ActiveXObject）。
* 所有现代浏览器（IE7+、Firefox、Chrome、Safari 以及 Opera）均内建 XMLHttpRequest 对象。
* XMLHttpRequest 用于在后台与服务器交换数据。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

### XMLHttpRequest使用流程

* 创建XMLHttpRequest对象
* 请求(同步或者异步请求)
* 响应

### Request的open方法

|  |
| --- |
|  |

* **注意：默认方法是异步的，也就是开子线程，同步方法已经被废弃，不能再用**

### ajax准备状态码

request.readyState  
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

### http响应状态

request.status

200: "OK"

404: 未找到页面

405:用来访问本页面的 HTTP 谓词不被允许（方法不被允许） ，也就是post不能请求get方法

### 获取响应数据

request.responseText

## 1.3登录的get案例

课程目的：掌握ajax的get请求

|  |
| --- |
|  |

### Step1:写一个get登录表单

|  |
| --- |
|  |

### Step2:写一个LoginServlet

* 实现get请求方法

|  |
| --- |
|  |

### Step3:ajax的get请求

|  |
| --- |
| <script type=*"text/javascript"*>  **function** getlogin() {  //1.获取表单的值  **var** username = document.getElementById('username').value;  **var** password = document.getElementById('password').value;    **var** params = 'username=' + username + '&password=' + password;  console.log(params);    //2.创建XMLHttpRequest对象  **var** request = **new** XMLHttpRequest();    //3.发送请求,把请求参数放在?号后面  request.open('get','${pageContext.request.contextPath}/LoginServlet?' + params);  request.send();    //4.监听准备状态的改变  request.onreadystatechange = **function**(){  console.log(request.readyState + ' - ' + request.status);  **if**(request.readyState == 4 && request.status == 200){  **var** rspTxt = request.responseText;  console.log('服务器响应：' + rspTxt);  **if**(rspTxt == 'success'){  alert('登录成功');  }**else**{  alert('登录失败');  }  }  }  }  </script> |

## 1.4登录的post案例

课程目的：掌握ajax的post请求

### ajax的post请求流程

1. 创建请求对象
2. 连接
3. 设置content-type请求头，**如果不设置请求头，发送请求会失败**
4. 发送请求，传递请求参数
5. 接收响应

### post请求登录的js代码

|  |
| --- |
|  |

# json

## 2.1 json字符串

* JSON 指的是 JavaScript 对象表示法（JavaScript Object Notation）
* JSON 是轻量级的文本数据交换格式
* JSON 独立于语言
* JSON 具有自我描述性，更易理解

### Json 字典

相当于 java 的map

|  |
| --- |
| { "firstName":"Bill" , "lastName":"Gates" } |

### Json数组

相当于数组里存了map,字典

|  |
| --- |
| [  { "firstName":"Bill" , "lastName":"Gates" },  { "firstName":"George" , "lastName":"Bush" },  { "firstName":"Thomas" , "lastName":"Carter" }  ] |

### Json 字典

字典里存数组

|  |
| --- |
| {  "employees": [  { "firstName":"Bill" , "lastName":"Gates" },  { "firstName":"George" , "lastName":"Bush" },  { "firstName":"Thomas" , "lastName":"Carter" }  ]  } |

## 2.2 js中的json对象和字符串相互转换

|  |
| --- |
|  |

# 3 jackson

课程目标：

如何从服务器返回json格式的数据

掌握json.jar的使用

|  |
| --- |
|  |

## 3.1 Jackson 简介

Jackson是一个简单基于Java应用库，Jackson可以轻松的将Java对象转换成json字符串和xml文档，同样也可以将json、xml转换成Java对象。Jackson所依赖的jar包较少，简单易用并且性能也要相对高些，并且Jackson社区相对比较活跃，更新速度也比较快。

## 3.2 jackson特点

* 容易使用 - jackson API提供了一个高层次外观，以简化常用的用例。
* 无需创建映射 - API提供了默认的映射大部分对象序列化。
* 性能高 - 快速，低内存占用，适合大型对象图表或系统。
* 干净的JSON - jackson创建一个干净和紧凑的JSON结果，这是让人很容易阅读。
* 不依赖 - 库不需要任何其他的库，除了JDK。
* 开源代码 - jackson是开源的，可以免费使用。

## 3.3对象转json字符串案例：

|  |
| --- |
| book模型 |
|  |
| 测试对象与json字符串的转换 |
|  |

## 3.4 json字符串转对象

|  |
| --- |
| String str = "{\"id\":\"1001\",\"name\":\"Java入门\",\"price\":19.88,\"pnum\":100,\"category\":\"IT\",\"description\":\"Good Book\"}";  System.***out***.println(str);    //创建 "对象映射" 对象  ObjectMapper mapper = **new** ObjectMapper();    Book book = mapper.readValue(str, Book.**class**);  System.***out***.println(book); |

# 4 ajax处理json数据

|  |
| --- |
|  |

## 4.1 修改LoginServlet,返回json数据

|  |
| --- |
|  |

## 4.2 修改JS代码

|  |
| --- |
| <script type=*"text/javascript"*>  **function** postlogin() {  //1.获取表单的值  **var** username = document.getElementById('username').value;  **var** password = document.getElementById('password').value;  **var** params = 'username=' + username + '&password=' + password;  console.log(params);    //2.创建XMLHttpRequest对象  **var** request = **new** XMLHttpRequest();    //3.发送请求,把请求参数放在send方法中  request.open('post','${pageContext.request.contextPath}/LoginServlet');  //设置请求头,一定要设置请求头  request.setRequestHeader("Content-Type","application/x-www-form-urlencoded");  request.send(params);    //4.监听准备状态的改变  request.onreadystatechange = **function**(){  console.log(request.readyState + ' - ' + request.status);  **if**(request.readyState == 4 && request.status == 200){  **var** rspTxt = request.responseText;  console.log(rspTxt);  //把返回的字符串转成json对象  **var** jsonObj = JSON.parse(rspTxt);  console.log(jsonObj);  **var** tip = document.getElementById('tip');//添加一个span标签用于显示登录状态    //通过对象点属性的方式取值  **if**(jsonObj.success == 1){  tip.innerHTML = "登录成功";  }**else**{  tip.innerHTML = jsonObj.message;  }  }  }  }  </script> |

# 5 get 请求中文处理

* get请求的url后面的参数不能传中文
* 工作中对get请求对中文进行url编码
* url编码是什么

URL编码遵循下列规则： 每对name/value由&；符分开；每对来自表单的name/value由=符分开。如果用 解码软件 解码软件 户没有输入值给这个name，那么这个name还是出现，只是无值。任何特殊的字符（就是那些不是简单的七位ASCII，如汉字）将以百分符%用十六进制编码，当然也包括象 =,&；，和 % 这些特殊的字符。其实url编码就是一个字符ascii码的十六进制。

* js提供的url编码和解码
* encodeURIComponent 编码与decodeURIComponent 解码
* encodeURI与decodeURIL
* encodeURI()与encodeURIComponent()区别
  + 两者的区别在于,encodeURI函数不会对以下的字符进行处理： “! @ # $ & \* ( ) = : / ; ? + ' ”
* 推荐使用encodeURIComponent()
* **默认情况下ajax,浏览器会对url进行编码**

### 案例：

|  |
| --- |
| <script type=*"text/javascript"*>  **var** urlStr = 'LoginServlet?username=张三丰&password=123';  //1.对字符串进行url编码,会对一些特殊字符串,比如中文,:,&,?号进行编码  **var** u1 = encodeURIComponent(urlStr);  **var** u2 = encodeURI(urlStr)  console.log(u1);  console.log(u2);    //2.对url进行解码,还原原始模样  console.log(decodeURIComponent(u1));  console.log(decodeURIComponent(u2));  </script> |

### 对前面get请求进行url编码

|  |
| --- |
|  |

# 6 ajax的兼容性问题

|  |
| --- |
| var request;  if (window.XMLHttpRequest)  {  // IE7+, Firefox, Chrome, Opera, Safari 浏览器执行代码  request=**new XMLHttpRequest();**  }  else  {  // IE6, IE5 浏览器执行代码  request=new ActiveXObject("Microsoft.XMLHTTP");  } |

# 7 总结

工作中使用ajax是用于发送get/post的网络异步请求，工作中用的更多的是**jquery封装好的get/post请求**，后面我们来学习jquery