Ajax学习

问题：

目前来，所有的请求的发送都是通过浏览器自己直接进行发送，响应是浏览器在接收到响应信息后自主的将响应数据覆盖当前页面显示。现在，要求在保留原有页面内容的情况下显示新的响应内容。

解决:

使用ajax技术

ajax是什么?

“Asynchronous Javascript And XML”（异步JavaScript和XML），

有什么用?

咱们的网页如果想要刷新局部内容。 那么需要重新载入整个网页。用户体验不是很好。 就是为了解决局部刷新的问题。 保持其他部分不动，只刷新某些地方。

|  |
| --- |
|  |

#### JQuery中的ajax

我们学习JQuery框架的发现里面有ajax的功能

get与post请求及响应

|  |
| --- |
| <%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %> <**html**> <**head**>  <**title**>Title</**title**>  <**script src="js/jquery.js"**></**script**>  <**script**>  **$**(**function** () {  **$**.get(  **"AjaxServlet"**,  {  **"name"**:**"张三"**,  **"age"**:20  },  **function** (data) { *//data:是发送请求到后台,接收后台响应回来的数据在data中  alert*(data)  })  })  </**script**> </**head**> <**body**>  </**body**> </**html**> |
| <%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %> <**html**> <**head**>  <**title**>Title</**title**>  <**script src="js/jquery.js"**></**script**>  <**script**>  **$**(**function** () {  **$**.post(  **"AjaxServlet"**,  {  **"name"**:**"张三"**,  **"age"**:20  },  **function** (data) { *//data:是发送请求到后台,接收后台响应回来的数据在data中  alert*(data)  })  })  </**script**> </**head**> <**body**>  </**body**> </**html**> |
| **package** com.manager.servlet;  **import** javax.servlet.ServletException; **import** javax.servlet.annotation.WebServlet; **import** javax.servlet.http.HttpServlet; **import** javax.servlet.http.HttpServletRequest; **import** javax.servlet.http.HttpServletResponse; **import** java.io.IOException;  @WebServlet(name = **"AjaxServlet"**,urlPatterns = **"/AjaxServlet"**) **public class** AjaxServlet **extends** HttpServlet {  @Override  **protected void** service(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {  req.setCharacterEncoding(**"utf-8"**);  resp.setContentType(**"text/html;chatset=utf-8"**);  String name = req.getParameter(**"name"**);  String age = req.getParameter(**"age"**);  System.***out***.println(name);  System.***out***.println(age);  resp.getWriter().println(**"AjaxServlet响应"**);  } } |

### 注册时：校验用户名是否存在

### Ajax之响应数据

服务器响应给浏览器的数据应该是字符串类型

但是如果数据量较大，我们就需要在服务器端将数据拼接成一个良好的格式的字符串数据，响应给浏览器

浏览器根据格式的解析和方便的使用里面数据

问题：

什么样的格式算是良好的格式呢?

解决：

Json的概念：

其实json就是js创建对象的一种格式

任何支持的类型都可以通过 JSON 来表示，例如字符串、数字、对象、数组等。但是对象和数组是比较特殊且常用的两种类型。

对象：对象在 JS 中是使用花括号包裹 {} 起来的内容，数据结构为 {key1：value1, key2：value2, ...} 的键值对结构。在面向对象的语言中，key 为对象的属性，value 为对应的值。键名可以使用整数和字符串来表示。值的类型可以是任意类型。

Json的格式：

var 对象名={

键：值,

键：值,

……

键：值

}

普通文本：

后台在接收到ajax请求后，处理后直接响应普通字符串给ajax

josn数据：

后台在接收到ajax请求后，处理后响应json格式的字符串给ajax

ajax处理代码中使用eval()方法将json数据转换为js对的对象，

将对象中的数据通过js的dom操作显示到页面中。

|  |
| --- |
| **package** com.manager.servlet;  **import** com.google.gson.Gson; **import** com.manager.pojo.User; **import** com.manager.util.DBUtil;  **import** javax.servlet.ServletException; **import** javax.servlet.annotation.WebServlet; **import** javax.servlet.http.HttpServlet; **import** javax.servlet.http.HttpServletRequest; **import** javax.servlet.http.HttpServletResponse; **import** java.io.IOException; **import** java.sql.PreparedStatement; **import** java.sql.ResultSet; **import** java.sql.SQLException; **import** java.util.ArrayList;  @WebServlet(name = **"ServletTest"**,urlPatterns = **"/ServletTest"**) **public class** ServletTest **extends** HttpServlet {  @Override  **protected void** service(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {  resp.setContentType(**"text/html;charset=utf-8"**);  PreparedStatement pstmt = DBUtil.*getPstmt*(**"select \* from t\_user2 where sid = 21"**);  ArrayList<User> users = **new** ArrayList<>();  User u =**null**;  **try** {  ResultSet rs = pstmt.executeQuery();  **while**(rs.next()){  **int** sid = rs.getInt(**"sid"**);  String uname = rs.getString(**"uname"**);  String pwd = rs.getString(**"pwd"**);  **int** spower = rs.getInt(**"spower"**);  u = **new** User(sid,uname,pwd,spower);  users.add(**new** User(sid,uname,pwd,spower));  }  } **catch** (SQLException e) {  e.printStackTrace();  }  Gson g = **new** Gson();*//创建gson对象* String userJson = g.toJson(users);*//可以把对象转化成对应的json格式,如果是集合或数组,则转成json的数组格式的字符串,如果是单个对象* String uJson = g.toJson(u);  resp.getWriter().write(uJson);  System.***out***.println(uJson);  } } |
| <%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %> <**html**> <**head**>  <**title**>Title</**title**> </**head**> <**script src="js/jquery.js"**></**script**> <**script**>  **$**(**function** () {  **$**(**"#but"**).click(**function** () {  **$**.get(  **"ServletTest"**,  **function** (data) {  *// alert(data);//测试是否获取到了后台的数据  eval*(**"var u ="**+ data);  *//上面等同于申明了一个json对象 var u = {"sid":21,"uname":"韩信","pwd":"54321","spower":1}   alert*(u.**uname**);  *alert*(u.**pwd**);  *//使用申明的方式创建一个json对象* **var** u2 = {**"sid"**:1,**"uname"**:**"韩信2"**,**"pwd"**:**"123"**,**"spower"**:1}  *alert*(u2.**uname**)  *alert*(u2.**pwd**)  }  )  })  }) </**script**> <**body**>  <**input type="button" id="but" value="测试gson"**> </**body**> </**html**> |