# day36-ajax&json

## 学习目标:

- 1. 能够理解异步的概念
- 2. 能够使用jQuery的\$.get()进行访问
- 3. 能够使用iQuery的\$.post()进行访问
- 4. 能够掌握ison的三种数据格式
- 5. 能够使用json转换工具Jackson进行json格式字符串的转换
- 6. 能够完成用户名是否存在的查重案例
- 7. 能够完成自动补全的案例

# 案例一:使用JS的AJAX完成用户名的异步校验

## 一,需求分析

我们有一个网站,网站中都有注册的页面,当我们在注册的页面中输入用户名的时候(失去焦点的时候),这个时候会提示,用户名是否存在。

# 用户注册的页面

用户名	aaa 用户名已经被占用!
密码	
邮箱	
电话	
注册	

## 二,技术分析

- 1. AJAX的概述
- 1.1 什么是AJAX

## 

AJAX即"Asynchronous Javascript And XML"(异步JavaScript和XML),是指一种创建交互式网页应用的网页开发技术。

AJAX = 异步 JavaScript和XML(标准通用标记语言的子集)。

AJAX 是一种用于创建快速动态网页的技术。

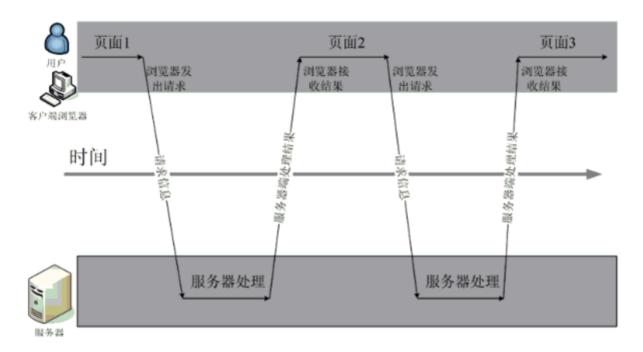
通过在后台与服务器进行少量数据交换,AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下,对网页的某部分进行更新。

传统的网页(不使用 AJAX)如果需要更新内容,必须重载整个网页页面。

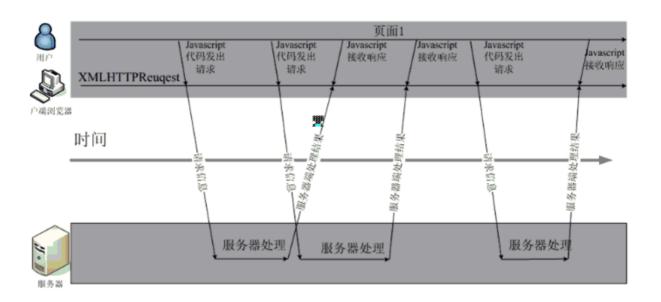
#### 说白了: 使用A Jax可以做异步的请求

#### 1.2什么是异步

• 同步



异步



#### 1.3为什么要学习AJAX

提升用户的体验。异步

#### 实现页面局部刷新。

将部分的代码,写到客户端浏览器。

## 2.js的Ajax入门

#### 2.1步骤

第一步: 创建异步请求对象。

第二步:打开连接。

第三步:发送请求。

第四步:设置监听对象改变所触发的函数,处理结果

从W3CSchool中抄写



#### 2.2 GET请求方式的入门

```
@WebServlet("/ms")

public class MyServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws

ServletException, IOException {
        doGet(request,response);
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws

ServletException, IOException {
        System.out.println("收到Ajax");
        System.out.println(request.getParameter("name"));
        response.getWriter().print("Hello");
    }
}
```

```
xmlhttp.onreadystatechange=function () {
    if(xmlhttp.readyState==4&&xmlhttp.status==200){
    alert(xmlhttp.responseText);
    }
}
</script>
```

#### 2.3POST请求方式的入门

```
<body>
<input type="button" value="Post方式"onclick="send()"/>
<script>
   function send() {
       var xmlhttp;
        if (window.XMLHttpRequest)
        {// code for IE7+, Firefox, Chrome, Opera, Safari
            xmlhttp=new XMLHttpRequest();
        }
        else
        {// code for IE6, IE5
            xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
        //请求方式 路径
        xmlhttp.open("post","ms");
        xmlhttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
        xmlhttp.send("name=zs&age=18");
        xmlhttp.onreadystatechange=function () {
            if(xmlhttp.readyState==4&&xmlhttp.status==200){
            alert(xmlhttp.responseText);
            }
        }
</script>
```

## 3.JS的Ajax

#### 3.1异步请求的对象XMLHttpRequest

不同的浏览器对该对象的创建的方式不一样,MSIE浏览器,比较早的浏览器,创建这个对象的时候将这个对象封装到ActiveXObject的插件中。像火狐或者谷歌浏览器则直接new出来。

#### 3.2XMLHttpRequest的对象的API

#### 3.2.1方法

open(): 打开连接。传递三个参数。第一个是请求方式(GET/POST),第二个是请求路径,第三个是否是异步的(默认就是异步,不需要这个参数)。

send([post请求的参数]): 发送请求。

setRequestHeader():解决POST请求参数的问题。 key和值 content-type

#### 3.2.2属性

onreadystatechange: 监听该对象的状态的改变,需要一个函数响应它

readyState:该属性就记录这个对象的状态。

0 (未初始化)	对象已建立,但是尚未初始化(尚未调用open方法)
1 (初始化)	对象已建立,尚未调用send方法
2 (发送数据)	send方法已调用,但是当前的状态及http头未知
3 (数据传送中)	已接收部分数据,因为响应及http头不全,这时通过responseBody和responseText获取部分数据会出现错误,
4 (完成)	数据接收完毕,此时可以通过通过responseBody和responseText获取完整的回应数据

status:状态码。

responseText:获得字符串形式的响应数据(响应体)。

responseXML:获得XML形式的响应数据(响应体)

## 三,思路分析

- 1. 创建页面和数据库
- 2. 给用户名输入框设置失去焦点(onblur)事件,创建一个checkUserName()的函数响应这个事件
- 3. 在checkUserName()函数里面

```
function checkUserName(){
    //1.获得用户输入的用户名
    //2.发送Ajax请求服务器(UserServlet),获得响应的结果
    //3. 根据响应的结果,给用户提示
}
```

4. 创建UserServlet

- 5. 创建UserService
- 6. 创建UserDao

```
sql = "select * from user where username = ?";
```

## 四,代码实现

#### 1.环境的准备

• 创建数据库和表

```
create database web36;
use web36;
create table user(
   id int primary key auto_increment,
   username varchar(20),
   password varchar(20),
   email varchar(50),
   phone varchar(20)
);
insert into user values (null, 'aaa', '123', 'aaa@163.com', '15845612356');
insert into user values (null, 'bbb', '123', 'bbb@qq.com', '15845612356');
insert into user values (null, 'ccc', '123', 'ccc@163.com', '15845612356');
```

• 创建实体类

```
public class User implements Serializable{
    private int id;
    private String username;
    private String password;
    private String email;
    private String phone;
构造get/set
toString方法
```

- 导入jar包
- 引入工具类(C3P0Utils),配置文件(c3p0-config.xml)
- 页面的准备

```
<html lang="en">
<head>
   <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<center>
   <h1>用户注册页面</h1>
   用户名:
        <input type="text" id="username" onblur="checkUserName(this)" />
            <span id="usernamespan"></span>
```

```
密码:
          <input type="password" />
          = (/td>
          <input type="password" />
          电话:
          <input type="password" />
          <input id="submitId" type="button" value="注册"/>
       </center>
</body>
<script>
  function checkUserName(o) {
      var xmlhttp;
      if (window.XMLHttpRequest)
      {// code for IE7+, Firefox, Chrome, Opera, Safari
         xmlhttp=new XMLHttpRequest();
      }
      else
      {// code for IE6, IE5
         xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
      //请求方式 路径
      xmlhttp.open("post","ms");
      xmlhttp.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
      xmlhttp.send("name="+o.value);
      xmlhttp.onreadystatechange=function () {
         if(xmlhttp.readyState==4&&xmlhttp.status==200){
             var result = xmlhttp.responseText;
             if(result=="true"){
              document.getElementById("usernamespan").innerHTML="<font color='green'>用户名可用
</font>";
                 document.getElementById("submitId").removeAttribute("disabled","disabled");
             }else{
                 document.getElementById("usernamespan").innerHTML="<font>用户名已存在</font>";
                 document.getElementById("submitId").setAttribute("disabled","disabled");
             }
         }
```

```
}
</script>
</html>
```

### 2.代码实现

• Dao层

• Service层

```
public class Service {
    public boolean getuser(String name) {
        Dao dao=new Dao();
        User user=dao.find(name);
        if(user==null){
        return true;
        }else{
            return false;
        }
    }
}
```

• Servlet层

```
@WebServlet("/ms")
public class MyServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
IOException {
        doGet(request,response);
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
IOException {
        String name = request.getParameter("name");
        Service ss=new Service();
        boolean flag = ss.getuser(name);
        response.getWriter().print(flag);
    }
}
```

# 案例二:使用JQ的Ajax完成用户名异步校验

## 一,需求分析

我们有一个网站,网站中都有注册的页面,当我们在注册的页面中输入用户名的时候,这个时候会提示,用户名是否存在。

# 用户注册的页面

用户名	aaa 用户名已经被占用!
密码	
邮箱	
电话	
注册	

## 二,技术分析

## 1.JQuery的Ajax【重点】

#### 1.1 为什么要使用JQuery的AJAX

因为传统(is里面)的AJAX的开发中,AJAX有两个主要的问题:

浏览器的兼容的问题,编写AJAX的代码太麻烦而且很多都是雷同的。

在实际的开发通常使用JQuery的Ajax

#### 1.2JQuery的Ajax的API

请求方式	语法
GET请求	\$.get(url, [data], [callback], [type])
POST请求	\$.post(url, [data], [callback], [type])
AJAX请求	\$.ajax([settings])
GET请求(3.0新特性)	\$.get([settings])
POST请求(3.0新特性)	\$.post([settings])

## 2.入门

#### 2.1get()

• get方式,语法 \$.get(url, [data], [callback], [type]);

参数名称	解释
url	请求的服务器端url地址
data	发送给服务器端的请求参数,格式可以是key=value,也可以是js对象
callback	当请求成功后的回掉函数,可以在函数体中编写我们的逻辑代码
type	预期的返回数据的类型,取值可以是 xml, html, script, json, text, _defaul等

### 2.2post()

• post方式, 语法 \\$.post(url, [data], [callback], [type])

参数名称	解释
url	请求的服务器端 <b>url地址</b>
data	发送给服务器端的请求参数,格式可以是key=value,也可以是js对象
callback	当请求成功后的回掉 <b>函数</b> ,可以在函数体中编写我们的逻辑代码
type	预期的返回数据的类型,取值可以是 xml, html, script, json, text, _defaul等

#### 页面

```
<head>
     <meta charset="UTF-8">
     <title>Title</title>

     <script src="js/jquery-3.3.1.min.js"></script>
```

```
</head>
<body>
<input type="button" value="jquerry方式的ajax" id="jqaj"/>
</body>
<script>
    $("#jqaj").click(function () {
        $.post("ms",{"name":"zs","age":"23"},function (result) {
            alert(""+result);
            })
        });
    </script>
</html>
```

Servlet

```
@WebServlet("/ms")
public class MyServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        doGet(request,response);
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        String name = request.getParameter("name");
        String age = request.getParameter("age");
        System.out.println("aJAX收到了请求"+name+age);
        response.getWriter().print("Oh 游戏boy");
    }
}
```

#### 2.3ajax()

• 语法 \$.ajax([settings])

其中, settings是一个js字面量形式的对象, 格式是{name:value,name:value... ...}, 常用的name属性名如下

属性名称	解释
url	请求的服务器端url地址
async	(默认: true) 默认设置下,所有请求均为异步请求。如果需要发送同步请求,请将此选项设置为 false
data	发送到服务器的数据,可以是键值对形式,也可以是js对象形式
type	(默认: "GET") 请求方式 ("POST" 或 "GET") ,默认为 "GET"
dataType	预期的返回数据的类型,取值可以是 xml, html, script, json, text, _defaul等
success	请求成功后的回调函数
error	请求失败时调用此函数

实例

```
<body>
<input type="button" value="jquerry方式的ajax" id="jqaj"/>
<script>
   $("#jqaj").click(function () {
        $.post({
        url:"ms",
        async:"true",
        data:{name:"zs",age:18},
        dataText:"text",
        success:function (result) {
           alert(result);
        },
        error:function () {
            alert("服务器异常");
        }
        })
   });
</script>
```

# 三,思路分析 页面传递name给Servlet , Servlet查询后返回boolean值 四 , 代码实现 其他同上面js

# 案例三:能够完成自动补全的案例(返回JSON数据)

## 一,需求分析

实现一个搜索页面,在文本框中输入一个值以后(键盘抬起的时候),给出一些提示

## 聖马

黑马一下

## 二,技术分析

#### 1.JSON

#### 1.1什么是JSON

## JSON 📝 編



JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。它基于ECMAScript的一个子集。 JSON采用完全独 立于语言的文本格式,但是也使用了类似于C语言家族的习惯(包括C、C++、C#、Java、JavaScript、Perl、Python等)。 这些特性使JSON成为理想的数据交换语言。 易于人阅读和编写,同时也易于机器解析和生成(一般用于提升网络传输速

■ | A 1000 | ■ 1010 | ■ 01

• 用我们自己的话来说: JSON就是一个容易生成和解析的字符串; 常用作前台(前端,IOS,安卓)和后台(JavaEE)之 间的数据交换

#### 1.2. JSON数据格式

类型	语法	解释
JSON对象	{name:value,name:value}	其中name是字符串类型,而value是任意类型
JSON数组	[value,value]	其中value是任意类型
混合类型	[{},{}] 或 {key:[]}	合理包裹嵌套对象类型和数组类

## 2.json的转换工具 jackson

#### 2.1json转换工具的概述

json的转换工具是通过java封装好的一些jar工具包,直接将java对象或集合转换成json格式的字符串。

#### 2.2常见的json转换工具

工具名称	介绍
Jsonlib	Java 类库,需要导入的jar包较多
Gson	google提供的一个简单的json转换工具
Fastjson	alibaba技术团队提供的一个高性能的json转换工具
Jackson	开源免费的json转换工具, springmvc转换默认使用jackson

#### 2.3Jackson的使用

导入jar

```
    jackson-annotations-2.2.3.jar
    jackson-core-2.2.3.jar
    jackson-databind-2.2.3.jar
```

• User对象转json

```
import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.ObjectMapper;
import User.User;
public class Test {
    public static void main(String[] args) throws JsonProcessingException {
        ObjectMapper objectMapper = new ObjectMapper();
        User user = new User(1, "zs", "123", "weq");
        String s = objectMapper.writeValueAsString(user);
        System.out.println(s);
    }
}
```

• list,Map集合转json

```
public class Test {
    public static void main(String[] args) throws JsonProcessingException {
        ObjectMapper objectMapper = new ObjectMapper();
        List list=new ArrayList();
        list.add("张三");
        list.add("李四");
        list.add("王五");
        String s = objectMapper.writeValueAsString(list);
        System.out.println(s);
    }
}
```

## 三,思路分析

### 四,代码实现

#### 1.环境的准备

• 创建数据库

```
create table words(
   id int primary key auto_increment,
   word varchar(50)
);
insert into words values (null, 'all');
insert into words values (null, 'after');
insert into words values (null, 'app');
insert into words values (null, 'apple');
insert into words values (null, 'application');
insert into words values (null, 'applet');
insert into words values (null, 'and');
insert into words values (null, 'animal');
insert into words values (null, 'back');
insert into words values (null, 'bad');
insert into words values (null, 'bag');
insert into words values (null, 'ball');
insert into words values (null, 'banana');
insert into words values (null, 'bear');
insert into words values (null, 'bike');
insert into words values (null, 'car');
insert into words values (null, 'card');
insert into words values (null, 'careful');
insert into words values (null, 'cheese');
insert into words values (null, 'come');
```

```
insert into words values (null, 'cool');
insert into words values (null, 'day');
insert into words values (null, 'day');
insert into words values (null, 'dirty');
insert into words values (null, 'duck');
insert into words values (null, 'east');
insert into words values (null, 'egg');
insert into words values (null, 'every');
insert into words values (null, 'every');
insert into words values (null, 'example');
```

• Words单元

```
public class Words {
   int id;
   private String word;
   构造set/get方法
   toString()方法
}
```

- 导入jar,工具类, 配置文件
- 创建页面

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Title</title>
   <script src="js/jquery-3.3.1.min.js"></script>
</head>
<body>
<center>
   <h1>黑马</h1>
   <input id="inputId" type="text" style="width: 500px; height: 38px;" /><input</pre>
       type="button" style="height: 38px;" value="黑马一下" />
   <div id="divId"
        style="width: 500px; border: 1px red solid; height: 300px; position: absolute;
left: 336px;display: none">
       </div>
</center>
</body>
<script>
```

```
var $table=$("#tabId");
   $("#inputId").keyup(function () {
       var value = this.value;
       if(value==''||value==null){
           $("#divId").hide();
           return;}
       $.post("ms",{"value":value},function (o) {
           $table.html("");
           $(o).each(function (i,obj) {
$table.append(""+obj.word+"");
           });
           $("#divId").show();
       },"json");
   });
</script>
</html>
```

• Dao层

```
public class Dao {
    public List<Words> getmylist(String value)throws Exception{
        JdbcTemplate jt=new JdbcTemplate(C3P0Utils.getds());
        List<Words> list = jt.query("select * from words where word like ? limit 0,5", new
BeanPropertyRowMapper<>(Words.class),"%"+value+"%");
        return list;
    }
}
```

Service

```
public class Service {
   public List<Words> getlist(String value) throws Exception {
        Dao dao=new Dao();
        return dao.getmylist(value);
   }
}
```

Servlet

```
@WebServlet("/ms")
public class MyServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        doGet(request,response);
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        try {
            request.setCharacterEncoding("utf-8");
        }
}
```

```
String value = request.getParameter("value");
    Service ss=new Service();
    List<Words> list = ss.getlist(value);
    ObjectMapper om=new ObjectMapper();
    String s = om.writeValueAsString(list);
    response.getWriter().print(s);
} catch (Exception e) {
    e.printStackTrace();
}
```

## 服务器内部错误时给用户提示,提升用户体验

### 代码改进

页面

```
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Title</title>
   <script src="js/jquery-3.3.1.min.js"></script>
</head>
<body>
<center>
   <h1>黑马</h1>
   <input id="inputId" type="text" style="width: 500px; height: 38px;" /><input</pre>
       type="button" style="height: 38px;" value="黑马一下" />
   <div id="divId"
        style="width: 500px; border: 1px red solid; height: 300px; position: absolute; left:
336px;display: none">
       </div>
</center>
</body>
<script>
   var $table=$("#tabId");
   $("#inputId").keyup(function () {
       var value = this.value;
       if(value==''||value==null){
           $("#divId").hide();
           return;}
       $.post("ms",{"value":value},function (o) {
           if(o.flag){
              $table.html("");
              $(o.data).each(function (i,obj) {
                  $table.append(""+obj.word+"");
              });
```

信息层

```
public class ResultInfo {
    private boolean flag;
    private Object data;
    private String message;
    构造get/set
    构造函数
}
```

Servlet

```
@WebServlet("/ms")
public class MyServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        doGet(request, response);
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        response.setContentType("text/html;charset=utf-8");
        ObjectMapper om=new ObjectMapper();
        ResultInfo resultInfo=null;
String data=null;
       String message;
        try {
            request.setCharacterEncoding("utf-8");
            String value = request.getParameter("value");
            Service ss=new Service();
            List<Words> list = ss.getlist(value);
resultInfo=new ResultInfo(true,list,"查询成功");
        } catch (Exception e) {
            e.printStackTrace();
            resultInfo=new ResultInfo(false,null,"查询失败");
        }finally{
data=om.writeValueAsString(resultInfo);
response.getWriter().print(data);
        }
   }
}
```