
WORK
STAY
TRUE
千锋教育
QIAN FENG EDUCATION
做真实的自己，用良心做教育
Ajax下
H5教学部

目录

做真实的自己，用良心做教育

千锋教育
QIAN FENG EDUCATION

POST请求

字符编码和解码

特殊字符传参产生的问题可以使用encodeURIComponent()进行编码处理, 中文字符的返回及传参, 可以将页面保存和设置为utf-8格式即可.

字符编码

```
var url = encodeURIComponent(name) +  '=  ' + encodeURIComponent(value);
```

字符解码

```
var url2 = decodeURIComponent(url);
```

POST请求

请求头

请求头包含HTTP的头部信息, 即服务器返回的响应头信息和客户端发送出去的请求头信息. 我们可以获取响应头(response)信息或者设置请求头(request)信息。

使用 getAllResponseHeaders()获取整个响应头信息

```
console.log(xhr.getAllResponseHeaders());
```

使用 getResponseHeader()获取单个响应头信息

```
console.log(xhr.getResponseHeader('Content-Type'));
```

使用 setRequestHeader()设置单个请求头信息

```
xhr.setRequestHeader(“MyHeader”, Zhang); //放在open方法之后, send方法之前
```

POST请求

POST请求

POST请求可以包含非常多的数据, 我们在使用表单提交的时候, 很多就是使用的POST传输方式。

```
xhr.open('post', 'demo.php', true);
```

POST请求向服务器发送的数据, 不会跟在URL后面, 而是通过send()方法向服务器提交数据。

```
xhr.send('name=Zhang&age=100');
```

POST请求和Web表单提交不同, 需要使用XHR来模仿表单提交.

```
xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
```

从性能上来讲POST请求比GET请求消耗更多一些, 用相同数据比较, GET最多比 POST快两倍. 这也是我们GET请求的使用率大于POST的原因.

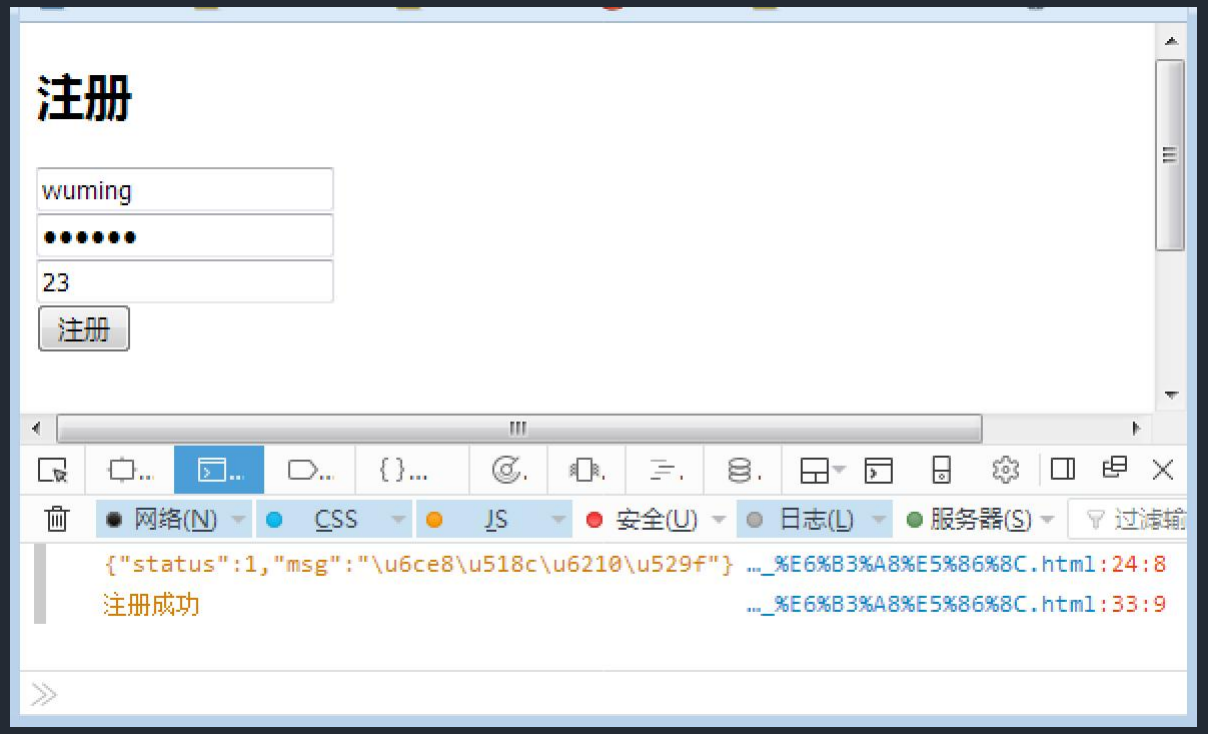
示例: 使用post请求方式验证注册表单的用户名是否可以注册.

登录注册

注册（POST）

注册接口: http://60.205.181.47/myPHPCode3/register.php

参数:username=zhangsan, 用户名（必须）
参数:password=123456,密码（必须）
参数:age=33,年龄（必须）

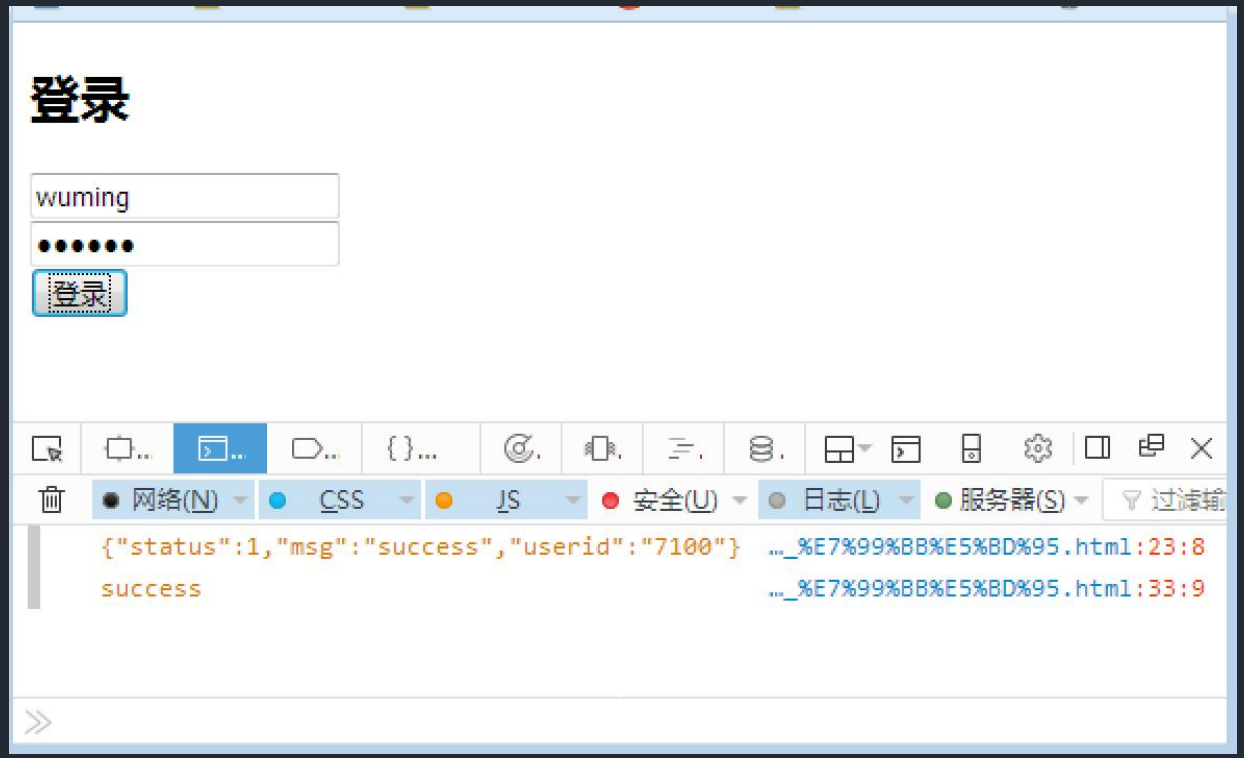


登录注册

登录（POST）

登录接口: http://60.205.181.47/myPHPCode3/login.php

参数:username=zhangsan,用户名
参数:password=123456,密码



Promise

Promise

实现Ajax异步的依赖调用

例如： 实现先注册 后登陆

```
var p = new Promise(function(resolve, reject){
    ajax({
        success: function(d){
            resolve(d);
        }
    })
})
p.then(function(data){
    ajax({
        success: function(d){
            console.log("登录成功");
        }
    })
})
```

解决Ajax跨域问题

同源策略

同源策略（Same origin policy）是一种约定，它是浏览器最核心也最基本的安全功能，如果缺少了同源策略，则浏览器的正常功能可能都会受到影响。可以说Web是构建在同源策略基础之上的，浏览器只是针对同源策略的一种实现。

同源策略，它是由Netscape提出的一个著名的安全策略，现在所有支持JavaScript的浏览器都会使用这个策略。

所谓同源是指，域名，协议，端口相同。

解决Ajax跨域问题

解决Ajax跨域问题一般有以下两种方式：

1，JSONP

2，由服务器端支持跨域

解决Ajax跨域问题

1，JSONP

什么是JSONP

JSON(JavaScript Object Notation)和JSONP(JSON with Padding)虽然只有一个字母的差别，但其实他们根本不是一回事儿：JSON是一种数据交换格式，而JSONP是一种依靠开发人员的聪明才智创造出的一种非官方跨域数据交互协议。

JSONP的优点：它不像XMLHttpRequest对象实现的Ajax请求那样受到同源策略的限制；它的兼容性更好，在更加古老的浏览器中都可以运行，不需要XMLHttpRequest或ActiveX的支持；并且在请求完毕后可以通过调用callback的方式回传结果。

JSONP的缺点：它只支持GET请求而不支持POST等其它类型的HTTP请求；

解决Ajax跨域问题

JSONP实现跨域的原理：

JSONP是一种非正式传输协议，该协议的一个要点就是允许用户传递一个callback参数给服务端，然后服务端返回数据时会将这个callback参数作为函数名来包裹住JSON数据，这样客户端就可以随意定制自己的函数来自动处理返回数据了。

JSONP本身就是一个get请求，而script节点本身也是一个get请求，这个思想是通过后端的配合（后端输出的response text必须符合js语法）更好的利用了get请求而已。

而前端封装一个方法，通过这个方法把请求注册的回调指向全局的一个具名函数，同时把具名函数的函数名和参数通过get请求传递给后端而已。

解决Ajax跨域问题

2，由服务器端支持跨域

后台文件设置支持跨域（PHP）

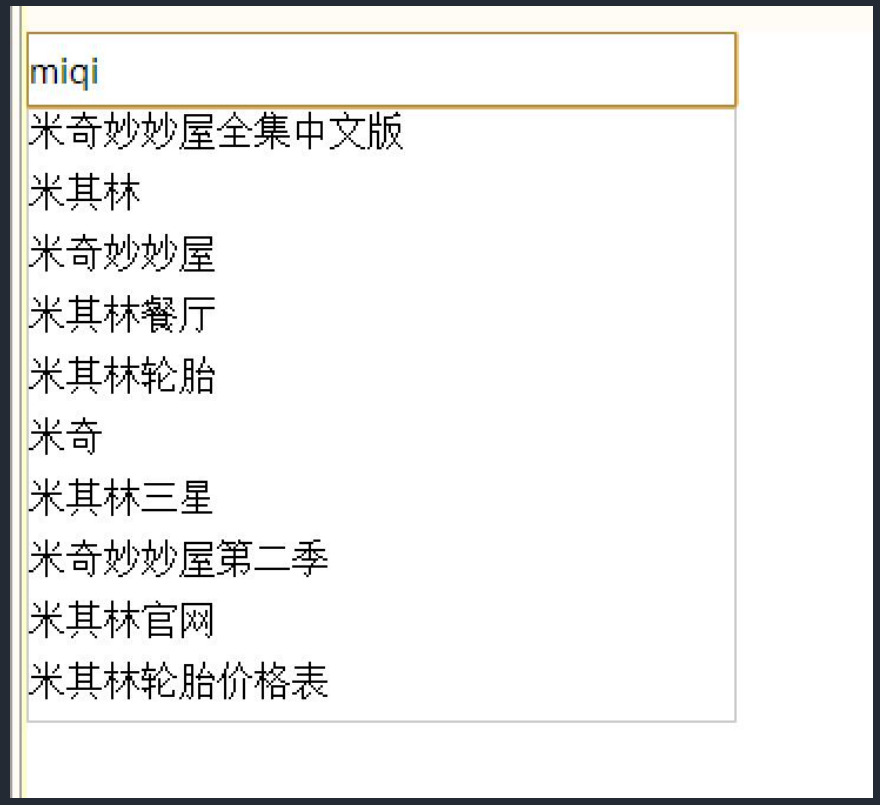
```
header('Access-Control-Allow-Origin:*');
```

解决Ajax跨域问题

练习

使用 jsonp获取百度搜索提示接口的数据并实时显示：

https://sp0.baidu.com/5a1Fazu8AA54nxGko9WTAnF6hhy/su?wd=miqi&cb=fn



Ajax封装

Ajax的封装(扩展)

对象的遍历

```
var box = {
    name : "张三",
    age : 199,
    run: function() {
        console.log("run");
    }
}
for(var i in box) {
    console.log(i);
    console.log(box[i]);
}
```

Ajax封装

函数的回调机制

```
function func1(obj) {
    console.log(obj.name);
    console.log(obj.age);
    obj.runWithPerson("李四");  //回调
}
var box = {
    name:"张三",
    age:22,
    runWithPerson:function(person){
        console.log("和 " + person + " 跑步...");
    }
};
func1(box);
```

Ajax封装

封装 Ajax:

```
//创建XHR对象
function createXHR() {
    if (window.XMLHttpRequest) {
        return new XMLHttpRequest(); //非IE7+, Chrome, FF
    }
    return new ActiveXObject("Micrisoft.XMLHTTP");  //IE6
}

//对参数编码,并使用&连接
function params(data) {
    var arr = [];
    //for-in遍历对象
    for (var i in data) {
        arr.push(encodeURIComponent(i) + "=" + encodeURIComponent(data[i]));
    }
    return arr.join("&");
}
```

Ajax封装

```
//封装Ajax函数
function ajax(obj) {
    var xhr = createXHR();
    obj.data = params(obj.data); //参数
    //判断是否是get
    if (obj.type == "get") {
        obj.url += obj.data.length>0 ? ("?" + obj.data) : "";
    }
    //open
    xhr.open(obj.type, obj.url, obj.async);
    //send
    if (obj.type == "post") {
        xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
        xhr.send(obj.data);
    }
    else {
        xhr.send(null);
    }
```

Ajax封装

```
//异步
if (obj.async == true) {
    xhr.onreadystatechange = function() {
        console.log(xhr.responseText);
            if (xhr.readyState == 4) { //接受到服务器响应
                callback();
            }
        }
    }
//同步
else {
    callback();
}
```

Ajax封装

```
//接受到服务器数据后调用
    function callback() {
        if (xhr.status == 200) {
            obj.success(xhr.responseText); //回调
        }
        else {
            var errObj = {status: xhr.status, statusText: xhr.statusText};
            obj.failure(errObj);
        }
    }
}
```

Ajax封装

```
//调用ajax
ajax({
    type: "post",
    url:  “http://127.0.0.1/person.php” ,
    data: {regname:"lisi", age:33},
    async: true,
    success: function(responseText) {
        console.log("获取数据成功: " + responseText);
    },
    failure: function(errorObj) {
    console.log("获取数据失败! 状态码:"+errorObj.status + ", 状态信息:"+errorObj.statusText);
    }
});
```

作业

Ajax轮播图

1， 创建本地json文件，并将图片数据写在json文件中

2，使用Ajax获取json中的数据，并根据数据创建节点

THANK YOU

千锋教育
QIAN FENG EDUCATION

做真实的自己，用良心做教育