



做真实的自己，用良心做教育

H5教学部

Ajax上

目录

做真实的自己，用良心做教育

1

Ajax的介绍

2

同步请求

3

异步请求

4

访问后台数据

Ajax介绍

1, Ajax的概念

AJAX (阿贾克斯 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, 可以访问服务器数据的局部刷新的技术. AJAX不是一种新的编程语言, 而是一种用于创建更好更快以及交互性更强的Web应用程序的技术.

虽然 Ajax 中的 x 代表的是 XML , 但 Ajax 通信和数据格式无关 , 也就是说这种技术不一定使用 XML。相反目前常用的数据格式是JSON

2, Ajax的作用

允许客户端脚本发送HTTP请求, 去请求服务器的数据来创建动态网页, 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. 也称局部刷新(常见的例子:分页, 用户名即时验证, 聊天等);

3, Ajax的优势

- JavaScript, XML, HTML , CSS在 AJAX 中使用的 Web 标准已被良好定义 , 并被所有的主流浏览器支持。
- 支持异步请求。
- 能够向服务器请求额外的数据而无须卸载页面(无需刷新页面)

Ajax介绍

4, Ajax相关的概念

进程: 程序从开始到结束的一次执行过程, 一个基于操作系统的应用程序中一般至少会有一个进程.

线程: 一个进程当中程序同时运行的多个分支, 每个进程中至少会有一个主线程, 可以有多个分线程(子线程), 多个不同的线程是可以同时执行的(并行执行).

多线程: 多个线程并发执行的一种技术.

注: 每个运行的程序操作系统都会至少分配一个进程。一个进程里面至少包含一个或多个线程

同步: 在主线程中按步骤顺序执行, 做完前一件事情, 才能下一件事情, 不能同时做多件事情.

异步: 与同步处理相对, 异步处理不用阻塞当前线程来等待处理完成, 而是允许同时可以做多件事情, 换句话说就是由多个不同的线程在同时执行(同时执行, 也叫并发).

Ajax介绍

5, Ajax的核心

Ajax 技术核心是JavaScript对象XMLHttpRequest(简称 XHR), 它是一种支持异步请求数据的技术.

XHR的出现, 提供了向服务器发送请求和解析服务器响应提供了流畅的接口. 能够以异步方式从服务器获取更多的信息, 这就意味着, 用户只要触发某一事件, 在不刷新网页的情况下, 更新服务器最新的数据。

6, 创建XHR对象

```
var xhr = new XMLHttpRequest();  
console.log(xhr); //XMLHttpRequest
```

XHR对象支持IE7+、Firefox、Opera、Chrome 和 Safari 等浏览器, 但是不支持IE6.

如果要支持IE6, 则需要使用ActiveX对象通过传入参数Microsoft.XMLHTTP来实现.

```
var xhr = new ActiveXObject("Microsoft.XMLHTTP");
```

Ajax介绍

为了支持IE6+、Firefox、Opera、Chrome 和 Safari 等浏览器
我们可以封装一个函数来创建XHR对象:

创建XHR对象

```
function createXHR() {  
    //IE7+ 谷歌, 火狐, Safari等  
    if (window.XMLHttpRequest) {  
        return new XMLHttpRequest();  
    }  
    return new ActiveXObject("Microsoft.XMLHTTP"); //IE6  
}  
  
var xhr = createXHR();
```


Ajax介绍

7, XMLHttpRequest对象的属性和方法

XHR提供的方法:

open(): 准备好需要发送给服务器的内容.

接收三个参数: 要发送的请求类型(get/post), 请求的URL和是否异步.

如: **xhr.open('get' , 'demo.json' , false);** //URL为demo.json的get请求, false表示同步.

send(): 向服务器发送请求.

接收一个参数: 请求体发送的数据. 如果是get方式请求则填null.

如: **xhr.send(null);**

XHR提供的属性:

当请求发送到服务器端后, 服务器就会向客户端发送响应, 收到响应数据后, 响应的数据会自动填充XHR对象的属性. 一共有四个属性:

status: 响应的HTTP状态码 (重要)

statusText: HTTP状态码的说明 (重要)

responseText: 作为响应主体被返回的文本 (重要)

responseXML: 如果响应主体内容类型是"text/xml"或"application/xml", 则返回包含响应数据的 XML文档, 否则是null

Ajax介绍

其中status属性:

HTTP状态码, 一般我们只需要关心HTTP状态代码为200则表示请求服务器成功.

HTTP 状态码	状态字符串	说明
200	OK	服务器成功返回了页面
400	Bad Request	语法错误导致服务器不识别
401	Unauthorized	请求需要用户认证
404	Not found	指定的 URL 在服务器上找不到
500	Internal Server Error	服务器遇到意外错误, 无法完成请求
503	ServiceUnavailable	由于服务器过载或维护导致无法完成请求

同步请求

同步请求:

```
//使用之前写好的createXHR()函数
var xhr = createXHR();
//使用open()方法,设置请求方式为get,请求url为demo.json, false表示同步请求
xhr.open('get', 'demo.json', false);
//向服务器发送请求
xhr.send(null);

if (xhr.status == 200) {    //如果返回成功了
    console.log(xhr.responseText); //调出服务器返回的数据
} else {
    console.log( 'error. 状态代码:'  + xhr.status +  '错误信息:' + xhr.statusText);
}
```

异步请求

异步请求:(重要)

前面我们使用同步请求比较简单, 但是实际项目开发中一般不用, 因为同步会阻塞主线程, 所以我们常用的是异步请求;

使用异步调用的时候, 需要触发readystatechange事件, 然后检测readyState属性的值即可, readyState属性的值有5个, 最常用的是最后一个值:4,表示已经接收到了全部响应数据.

值	状态	说明
0	未初始化	尚未调用 open()方法
1	启动	已经调用 open()方法, 但尚未调用 send()方法
2	发送	已经调用 send()方法, 但尚未接受响应
3	接受	已经接受到部分响应数据
4	完成	已经接受到全部响应数据

异步请求

异步请求:

```
//使用之前写好的createXHR()函数
xhr = createXHR();
//使用open()方法, 设置true表示异步请求
xhr.open('get', 'demo.json', true);
//向服务器发送请求
xhr.send(null);
//监听状态改变
xhr.onreadystatechange = function () {
    if (xhr.readyState == 4) {
        if (xhr.status == 200) {
            console.log(xhr.responseText);
        } else {
            console.log( 'error. 状态码:' + xhr.status + ' 错误信息:' + xhr.statusText);
        }
    }
};
```

异步请求

abort()方法:

取消异步请求, 如果需要取消某异步请求, 则在send()之后再调用, 写在send()之前调用会报错

//取消异步请求

xhr.abort();

异步请求

示例:

使用Ajax异步请求person.json数据:

- 1, 点击按钮加载person.json中的数据并显示在table表格中(默认表格中没有数据, 只有表格标题)
- 2, 在下方显示一个滚动文字,并让其从右往左滚动

点击我加载数据

name	age	sex	phone	address	qq
张三	25	男	15823456754	北京海淀区	123456
李四	21	男	13823456754	上海市吴淞区	12345678
王五	28	男	15023456754	深圳市	10000
赵六	20	男	18723456754	陕西省延安市	12345678

滚动文字

异步请求

练习:

使用Ajax异步请求news.json数据并显示:

- 1, 点击按钮加载news.json中的数据并显示在table表格中(默认表格中没有数据, 只有表格标题)
- 2, 在下方显示一个滚动文字,并让其从右往左滚动

点击我加载数据

title	time
冰点特稿乡村匠人	04:21
专家: 年内还有较大连续降准空间	01:00
起底【流氓软件】暴利链: 恶意安装难卸载	04:21
冰点特稿乡村匠人	04:21
专家: 年内还有较大连续降准空间	01:00
起底【流氓软件】暴利链: 恶意安装难卸载	04:21
冰点特稿乡村匠人	04:21
专家: 年内还有较大连续降准空间	01:00
起底【流氓软件】暴利链: 恶意安装难卸载	04:21

滚动文字

访问后台数据

示例:

1, 在后台获取并返回person.json中的数据给客户端, 提供接口:

<http://60.205.181.47/myPHPCode2/person.php>

2, 使用该接口获取数据并显示.

点击我加载数据

name	age	sex	phone	address	qq
张三	25	男	15823456754	北京海淀区	123456
李四	21	男	13823456754	上海市吴淞区	12345678
王五	28	男	15023456754	深圳市	10000
赵六	20	男	18723456754	陕西省延安市	12345678

滚动文字

HTTP协议

HTTP协议

HTTP是一个属于应用层的面向对象的协议, 由于其简捷、快速的方式, 适用于分布式超媒体信息系统.

HTTP协议的主要特点有:

- 1, 支持客户端/服务端模式. 即请求(request)-响应(response)模式
- 2, 简单快速, 客户端向服务端发送请求时, 只需要传送请求方式和路径即可, 所以简单, 由于HTTP协议简单, 使得HTTP服务器的程序规模小, 因而速度很快;
- 3, 灵活, 传输数据类型种类多
- 4, 无连接, 请求一次服务器后立刻断开连接, 即非长连接 即短连接
- 5, 无状态, HTTP协议对事务处理没有记忆能力; session

HTTP协议的请求方式: GET, POST, HEAD, PUT等

HTTP包含: 请求头和请求体

HTTP协议

GET请求:

在通过HTTP协议向服务器请求的过程中, 有两种最常用的请求方式, 分别是: GET和POST.

在Ajax使用的过程中, GET的使用频率又要比POST高得多. GET请求最常用于向服务器获取数据, 也可以将少量字符串参数提交给服务器.

如:

```
xhr.open('get', 'login.php?username=zhang&pwd=123456', true);
```

通过URL后的问号?给服务器传递键值对数据, 服务器接收到请求后可以从中获取到对应的数据

HTTP协议

示例:

表单验证(在服务端验证)

输入用户名, 通过在服务器验证该用户名是否被注册过. 获取接口中的数据进行验证, 其中

<http://60.205.181.47/myPHPCode2/checkname.php?regname=zhangsan>

参数: regname表示用户名;

返回值: {

status: 1,

msg: "用户名可以注册!"

}

说明: status为0表示不可以注册, status为1表示可以注册.



访问后台数据

练习：加载更多数据（翻页，每页显示5条数据）

1, 在后台获取并返回football.json中的数据给客户端, 提供接口:

<http://60.205.181.47/myPHPCode2/football.php?page=1>

2, 使用该接口获取数据并显示.

申方剑

观众朋友们，球迷朋友们！大家晚上好，欢迎收看风云足球频道，现在我们为您带来的是一场宇宙足球甲级联赛的直播，是由拜仁慕尼黑迎战巴塞罗那的比赛！

韩乔生

随着守门员的一声哨响，比赛开始了！

张路

嘿嘿

宫磊

如果这场比赛巴萨能获胜，那么他们将在客场全取三分，然后带着优势回到自己的主场伯纳乌！

刘建宏

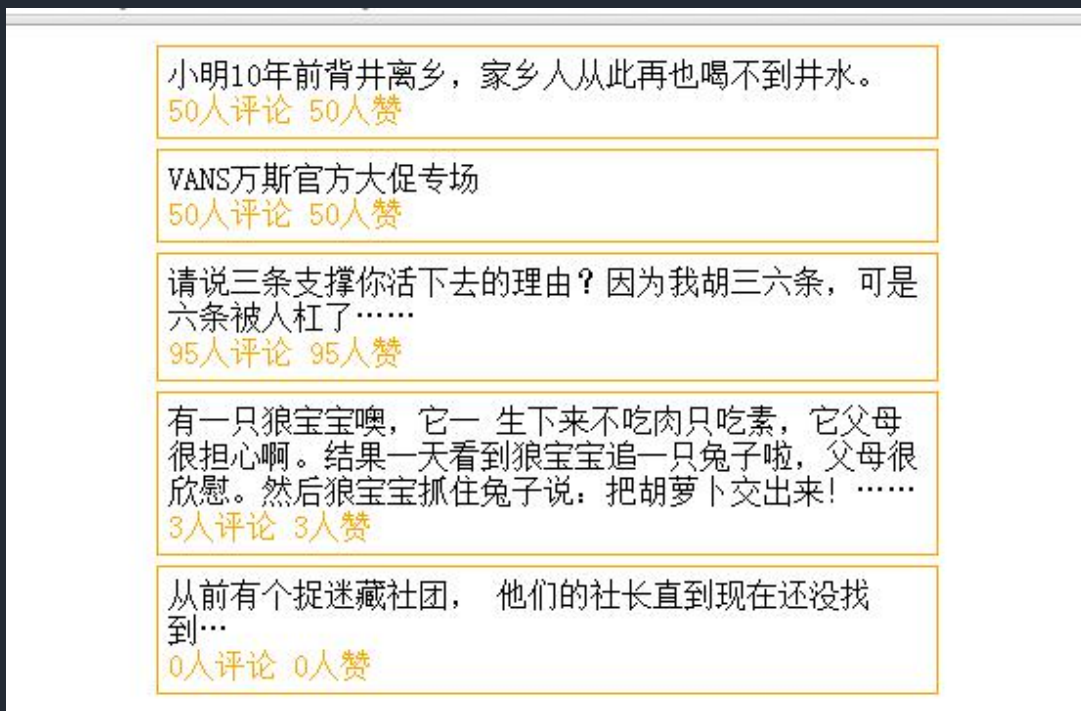
看这球！比利亚打门！球打在门柱上！再打！球又打在门柱上！再补！球进啦！进啦进啦进啦！

加载下一页数据

作业

1, 在后台获取并返回weibo.json中的数据给客户端, 提供接口:

<http://60.205.181.47/myPHPCode2/weibo.php> , 使用该接口获取数据并显示.



作业

2. 使用Ajax实现聊天室

实现聊天功能, 显示效果图如下:

1, 在输入框中输入内容后, 点击发送按钮可以发送出去;

2, 可以定时从服务器获取数据;

发送数据的URL:

<http://60.205.181.47/myPHPCode2/chat>,

参数: type=send

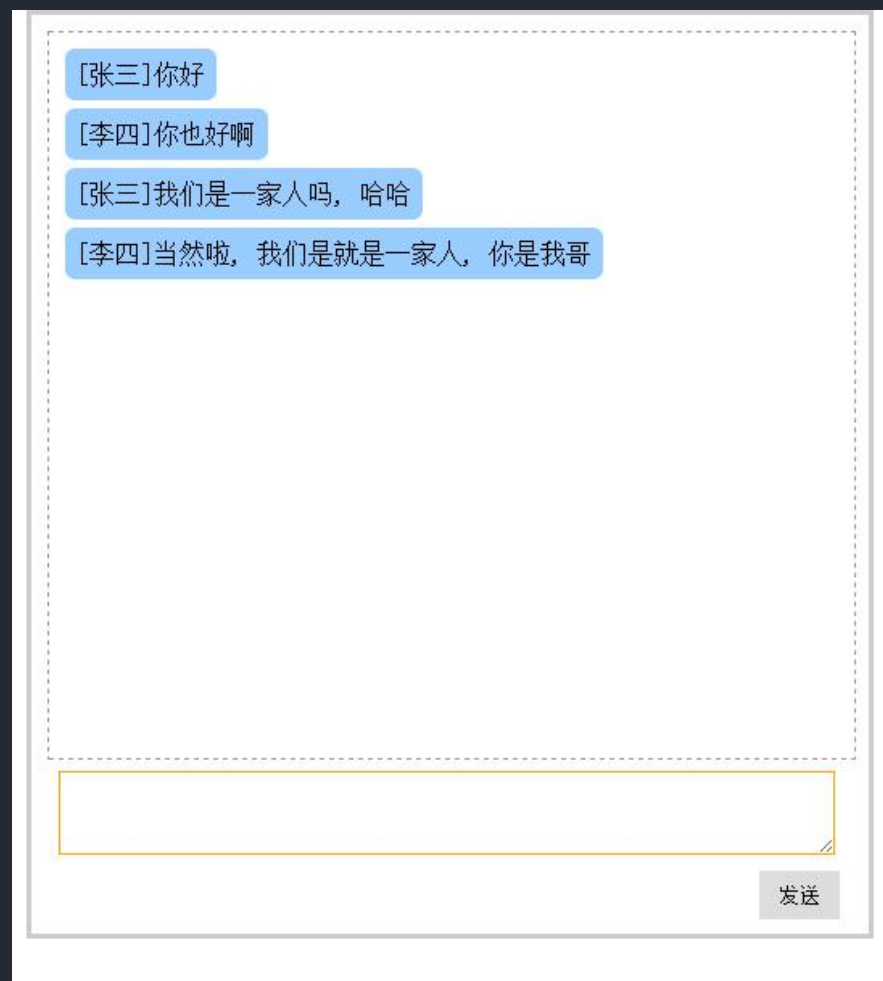
sender=发送者填入的用户名

msg=发送者的发送消息

获取数据的URL:

<http://60.205.181.47/myPHPCode2/chat>,

参数: type=query



THANK YOU



做真实的自己，用良心做教育