# 

**Ajax学习笔记**

**Java相关课程系列笔记之十一**

**笔记内容说明**

Ajax（程祖红老师主讲，占笔记内容100%）；

**目 录**

[一、 Ajax概述](#_Toc878)

[1.1什么是Ajax](#_Toc759)

[1.2 Ajax对象：如何获得Ajax对象](#_Toc22146)

[1.3 Ajax对象的属性](#_Toc12060)

[1.4编程步骤](#_Toc3722)

[1.5编码问题](#_Toc19389)

[1.6 Ajax的优点](#_Toc31776)

[1.7缓存问题（IE浏览器）](#_Toc946)

[1.8案例：简易注册（使用Ajax进行相关验证，get请求）](#_Toc28861)

[1.9案例：修改1.8案例，使用post请求](#_Toc29835)

[1.10案例：使用Ajax实现下拉列表](#_Toc17854)

[二、 JSON](#_Toc17403)

[2.1什么是JSON](#_Toc7350)

[2.2数据交换](#_Toc23910)

[2.3轻量级](#_Toc11042)

[2.4 JSON语法（www.json.org）](#_Toc313)

[2.5如何使用JSON来编写Ajax应用程序](#_Toc13568)

[2.6案例：股票的实时行情](#_Toc23480)

[2.7案例：显示热卖的前3个商品](#_Toc13527)

[2.8同步请求](#_Toc18263)

[2.9案例：修改1.8案例step1中的JS代码（使用同步请求）](#_Toc25898)

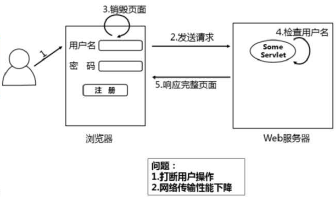
# 一、Ajax概述

## 1.1什么是Ajax

### 1.1.1 Ajax 概述

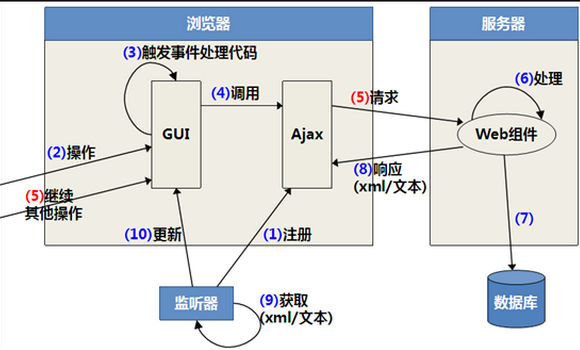
Asynchronous Javascript And Xml 异步的JavaScript和Xml，是一种用来改善用户体验的技术，其实质是利用浏览器内置的一个特殊对象：XMLHttpRequest，一般称之为Ajax对象。

传统请求响应方式，都是先销毁原来的页面，然后发送请求，等待服务器发送响应，再生成新页面。Ajax的工作流程：



### 1.1.2 Ajax原理

Ajax对象在向服务器发送请求时，浏览器并不会销毁当前页面，用户仍然可以对当前页面作其他的操作。异步地向服务器发送请求，服务器送回部分数据，不是一个完整的新的页面，而是文本或者Xml文档。在浏览器端，可以利用这些数据部分更新当前页面。整个过程，页面无刷新，不打断用户的操作。



## 1.2 Ajax对象获取

由于XMLHttpRequest: Ajax对象没有标准化，所以要**区分浏览器**。

function getXhr(){//注意：后面的案例**都将用到**此函数

var xhr=null;

if(window.XMLHttpRequest){

xhr=new XMLHttpRequest();//非IE浏览器

}else{

xhr=new ActiveXObject('Microsoft.XMLHttp');//IE浏览器

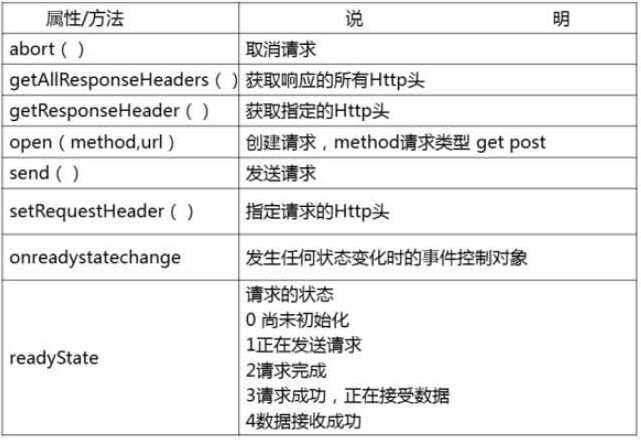
}

return xhr;

}

## 1.3 Ajax对象的属性

Ajax对象的请求端属性和方法如下表所示：



Ajax对象的请求响应端属性和方法如下表所示：



**onreadystatechange**：绑定一个事件处理函数（监听器），该函数用来处理readystatechange事件。Ajax对象的readyState属性发生改变，如从0到1，则会产生onreadystatechange事件。

**readyState**：返回Ajax对象与服务器通讯的状态，返回值是一个number类型的值。一共有5个值，分别是：

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。

3：（数据传送中）已接收部分数据。

4：（响应结束）Ajax对象已经获得了服务器返回的所有的数据。

## 1.4 Ajax发送请求

### 1.4.1 Ajax发送get请求

1. 调用xhr.open()方法：建立与服务器之间的连接，方法签名如下：

xhr.open('get',check\_username.do?username=chang&age=23,true);

三个参数依次为：请求方式、请求资源路径、请求是同步还是异步

true：异步请求Ajax对象发送请求时，用户可以对当前页面作其他的操作，不会销毁页面

false：同步请求Ajax对象发送请求时，浏览器锁定当前页面，用户只能等待，不会销毁页面

2. xhr.onreadystatechange=f1();绑定一个事件处理函数（监听器）

3. xhr.send(null);发送请求参数，因为参数已经写在了请求资源路径中，所以这里为null

4. 编写事件处理函数

function f1(){

if(xhr.readyState==4){

var txt=xhr.responseText;

dom操作……

}

}

### 1.4.2 Ajax发送post请求

1. xhr.open('post','check\_username.do',true);建立连接

2. xhr.setRequestHeader('content-type','application/x-www-form-urlencoded');发送一个消息头

3. xhr.onreadystatechange=func1();绑定一个事件处理函数（监听器）

4. xhr.send('username=chang');发送请求参数

注意事项：

* 与get请求不同，请求参数要放到send方法里面。
* 因为按照HTTP协议的要求，发送post请求时，应该发送一个content-type消息头，而Ajax对象在默认情况下，不会发送这个消息头，所以，需要调用setRequestHeader方法来添加。

## 1.5 Ajax编码

### 1.5.1发get请求编码

**（1）产生原因**

IE浏览器内置的Ajax对象会使用''GBK''或’‘GB2312’‘对**中文参数**进行**编码**，而其他浏览器Chrom、Firefox内置的Ajax对象会使用’‘utf-8’‘对**中文参数**进行**编码**。服务器端，默认用’‘ISO-8859-1’‘去**解码**。因为编码与解码所使用的字符集（编码格式）不一致，所以，会出现乱码问题。

**（2）解决方案**

step1：设置服务器使用指定的字符集去**解码**。比如，可以修改Tomcat的server.xml配置，添加URIEncoding=‘‘utf-8’‘。告诉服务器，对于所有的get请求，默认使用’‘utf-8’‘去解码，修改之后重新启动服务器。

<Connector port=‘‘8080’‘ maxHttpHeaderSize=‘‘8192’‘

maxThreads=‘‘150’‘ minSpareThreads=‘‘25’‘ maxSpareThreads=‘‘75’‘

enableLookups=‘‘false’‘ redirectPort=‘‘8443’‘ acceptCount=‘‘100’‘

connectionTimeout=‘‘20000’‘ disableUploadTimeout=‘‘true’‘ URIEncoding=‘‘utf-8’‘ />

step2：使用encodeURI()函数（JS中内置函数）**对请求地址进行编码**。encodeURI()函数会使用’‘utf-8’‘进行编码

xhr.open('get','check\_username.do?username='+$F('username'),true);

var uir='check\_username.do?username='+$F('username');

xhr.open('get',encodeURI(uri),true);

### 1.5.2发post请求编码

**（1）产生原因**

所有浏览器内置的Ajax对象都会使用’‘utf-8’‘对中文参数进行编码，而服务器默认情况下，会使用’‘ISO-8859-1’‘去解码。

注意事项：Firefox特殊，本应是乱码，但能正常显示。通过截取消息头发现Firefox会在消息头中自动添加’‘charset=utf-8’‘。

**（2）解决方案**

服务器端添加：request.setCharacterEncoding(‘‘utf-8’‘);

## 1.6 Ajax特点

1. 页面无刷新，不打断用户的操作

2. 按需要获取数据，客户端（浏览器）与服务器端之间的数据传输量大大减少

3. 是一种标准化的技术，不需要下载任何插件

## 1.7缓存问题

当发送get请求时，IE浏览器内置的Ajax对象会检查请求地址是否访问过，如果访问过，则不再向服务器发送请求。解决方式如下：

**方式一**：在请求地址后面添加一个随机数，用于欺骗IE，例如：

xhr.open('get','getNumber.do?'+Math.random(),true);

**方式二**：使用post方式发请求。

# 

# 二、JSON

## 2.1 JSON概述

JavaScript Object Natation，javascript对象表示法是一种轻量级的数据交换技术规范，因为借鉴了JavaScript对象创建的一种语法结构，故命名为JSON。

JSON将数据转换成一种中间的，与平台无关的数据格式（比如Xml或者JSON字符串）发送给另外一方来处理。

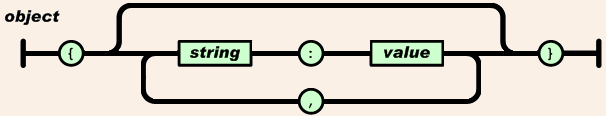
JSON相对于Xml，所需的数据大小要小的多，并且解析的速度更快。因此Xml现在用的少了,Ajax中的x即指用Xml交换数据。

## 2.4 JSON语法（[www.json.org](http://www.json.org)）

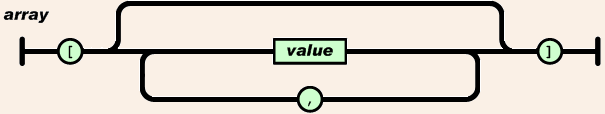
### 2.4.1 JSON 语法规则

JSON可以使用{} [] : , ‘‘‘‘这些字符来描述json规则，JSON规则描述的数据类型如下:

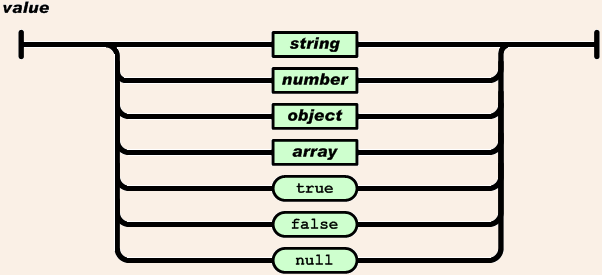
1. 以{}开始时，是在用json描述一个对象



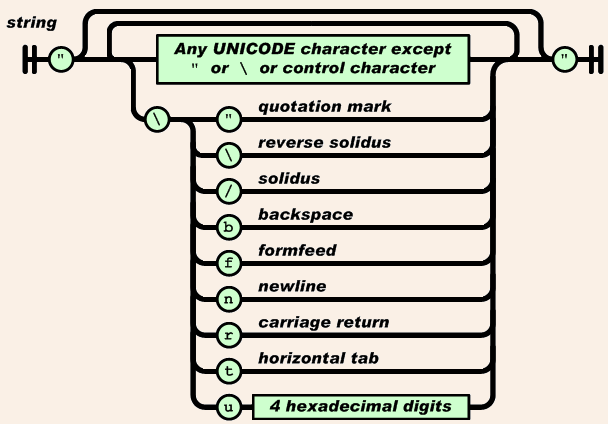
2. 以[]开始时，是在用json描述一个数组



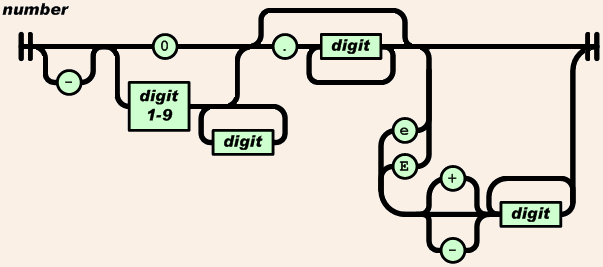
3. JSON 中的value可以是 string ,number, true or false , null, object ,array, 这些元素可以重复出现



4. 以””开始时，是在用json描述一个字符串



5. JSON 中的number类型表示格式如下



### 3.4.2 JSON对象和javascript对象

虽然javascript语言也用{}，[]两种符号来表达对象和数组，但只能说它与json的语法规则比较类似，实质上两者是一个不同的东西。

在JSON中，属性名一定要用双引号隔开引起来，属性值如果是String类型也必须用双引号隔起来。在javascript中，对象就没有这么严格的规定属性名可以不引或单引或双引，属性值如果是String类型可以单引或双引

### 3.4.3 JSON数据传输

虽然json是网络中传输的字符串的格式，但是当符合json格式的字符串到达客户端或服务器时，客户端或服务器端要针对这段符合json格式的字符串进行操作，所以都该完成共json字符串到集体语言对象的转换。

如果当前json字符串到达的是客户端，那么就要完成从json字符串到javascript对象的转换。如果是服务端，就是json字符串到java对象的一个转换。

可以使用浏览器内的原生的JSON对象的stringfy()方法和parse方法，来完成javascript对象和JSON字符串的转换

stringify()方法：javascript对象—>json字符串

parse()方法：json字符串—>javascript对象

### 2.4.4 Javascript对象创建方式

JavaScript中创建对象的三种方式

方式一：使用Json语法来创建

var p = {'name':'zs','age':22};

方式二：使用Object来创建

var obj = new Object();

obj.name = 'zs';

obj.age = 22;

方式三：利用JavaScript函数来创建（对象模版）

function Person(name,age){

this.name = name; this.age = age;

}

var person1 = new Person(‘‘zs’‘,22)

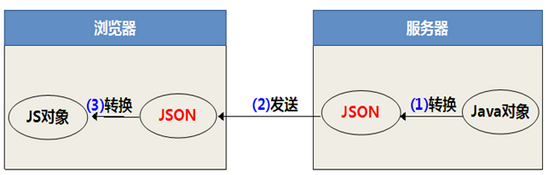
JavaScript表示一个对象组成的数组，[ {},{},{},……]

function f3(){//表示一个对象组成的数组

var arr=[{'name':'chang','age':22},{'name':'bo','age':23}];

alert(arr[1].name); }

## 2.5 JSON来编写Ajax应用



### 2.5.1 Java对象转换为JSON字符串

Java对象转换成JSON字符串,就是普通字符串用了JSON语法而已的字符串。一般使用JSON官方提供的API（json-lib）来实现转换（6个包：1主5副）。也可用谷歌提供的API。

情形一：Java对象转换成一个JSON字符串，使用JSONObject.fromObject()

/\*\* 效果：{'name':'chang','code':'10086','price':12.8} \*/

public static void test1(){

Stock s=new Stock(); s.setName(‘‘chang’‘); s.setCode(‘‘10086’‘); s.setPrice(12.8);

JSONObject obj=JSONObject.fromObject(s);//得到一个json对象

String jsonStr = obj.toString(); System.out.println(jsonStr);

}

情形二：Java对象组成的集合转换成一个JSON字符串，使用JSONArray.fromObject() [{'name':'chang','code':'10086','price':12.8},{'name':'chang','code':'100861','price':12.81},{'name':'chang','code':'1008611','price':12.82}]

情形三：Java对象组成的数组转换成一个JSON字符串，使用JSONArray.fromObject()

### 2.5.2 JSON字符串转换成JavaScript对象

可以使用一些工具提供的方法，比如prototype提供了evalJSON()函数，prototype是一个js文件，里面提供了很多常用的函数，比如：

$(id)：document.getElementBuId(id);

$F(id)：$(id.value);

$(id1,id2,id3...)：分别依据id1,id2...查找对应的节点，然后放到一个数据里面返回。

strip()：除掉字符串两端的空格。trim()函数也有，但有的浏览器不支持。

evalJSON()：将JSON字符串转换成对应的JavaScript对象或者JavaScript对象组成的数组。

//将一个JSON字符串转成一个JavaScript对象

function f4(){

var str=‘‘{'name':'chang','age':24}’‘;

//alert(typeof str);

//使用prototype框架提供的evalJSON函数，将字符串转成一个JavaScript对象

var obj=str.evalJSON();

alert(obj.name);

}

//将JSON字符串转换成一个JacaScript对象组成的数组

function f5(){

var str=‘‘[{'name':'chang','age':22},{'name':'bo','age':23}]’‘;

var arr=str.evalJSON(); alert(arr[1].name);

}

## 2.8同步请求

**同步请求**：Ajax对象在向服务器发送请求时，浏览器会锁定当前页面，用户不能够对当前页面做任何的操作。发送同步请求的方式如下：open(请求方式，请求地址，false)

优先使用异步，因为同步会影响性能，当服务器端处理比较慢的时候，浏览器会锁定当前页面（’‘假死’‘）。只有当客户端需要等待服务器的响应之后，才能继续向下执行时，应该使用同步。

Firefox的某些版本（低版本，如3、4）对于同步的支持比较特殊：不能使用xhr.onreadystatechange来绑定一个事件处理函数，而应该在send方法执行之后，才调用xhr.responseText方法来获得服务器返回的数据。例如：

function getType(){//返回浏览器类型

if(navigator.userAgent.indexOf('Firefox')!=-1){

return 'firefox';

}else{

return 'other';

}//其他浏览器

}

function quoto(){

if(getType() != 'firefox'){

xhr.onreadystatechange=function(){

if(xhr.readyState == 4){ var txt = xhr.responseText; …… } }; }

xhr.send('username=' + $F('username'));

if(getType() == 'firefox'){ var txt = xhr.responseText; …… } }

## 2.9案例：使用同步请求