

# AJAX课程



## 第1章：原生AJAX

### 1.1 AJAX 简介

AJAX 全称为Asynchronous Javascript And XML，就是异步的 JS 和 XML。

通过AJAX可以在浏览器中向服务器发送异步请求，最大的优势：无刷新获取数据。

AJAX 不是新的编程语言，而是一种将现有的标准组合在一起使用的新方式。

### 1.2 XML简介

XML 可扩展标记语言。

XML 被设计用来传输和存储数据。

XML和HTML类似，不同的是HTML中都是预定义标签，而XML中没有预定义标签，全都是自定义标签，用来表示一些数据。

```
<people>
  <name>孙悟空</name>
  <age>18</age>
  <gender>男</gender>
</people>
```

在老式的金融系统当中可能还能看到，现在主流的方式已被JSON取代了。

```
{
  "name": "孙悟空",
  "age": 18,
  "gender": "男"
}
```

## 1.3 AJAX的特点

### 1.3.1 AJAX的优点

- 1.可以无需刷新页面而与服务器端进行通信。
- 2.允许根据用户事件来更新部分页面内容。

### 1.3.2 AJAX的缺点

- 1.没有浏览历史，不能回退
- 2.存在跨域问题
- 3.SEO不友好

## 1.4 AJAX的使用

### 1.4.1 核心对象

XMLHttpRequest, AJAX的所有操作都是通过该对象进行的。

### 1.4.2 使用步骤

//1. 创建XMLHttpRequest对象

```
var xhr = new XMLHttpRequest();
```

//2. 设置请求信息

```
xhr.open(method, url);
```

//可以设置请求头, 一般不设置

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

//3. 发送请求

```
xhr.send(body) //get请求不传body参数, 只有post请求使用
```

//4. 接收响应

//xhr.responseXML 接收xml格式的响应数据

//xhr.responseText 接收文本格式的响应数据

```
xhr.onreadystatechange = function (){  
    if(xhr.readyState == 4 && xhr.status == 200){  
        var text = xhr.responseText;  
        console.log(text);  
    }  
}
```

```
}
```

### 1.4.3 解决IE缓存问题

问题：在一些浏览器中(IE),由于缓存机制的存在，ajax只会发送的第一次请求，剩余多次请求不会在发送给浏览器而是直接加载缓存中的数据。

解决方式：浏览器的缓存是根据url地址来记录的，所以我们只需要修改url地址即可避免缓存问题

```
xhr.open("get", "/testAJAX?t="+Date.now());
```

### 1.4.4 AJAX请求状态

xhr.readyState 可以用来查看请求当前的状态

<https://developer.mozilla.org/zh-CN/docs/Web/API/XMLHttpRequest/readyState>

- 0: 表示XMLHttpRequest实例已经生成，但是open()方法还没有被调用。
- 1: 表示send()方法还没有被调用，仍然可以使用setRequestHeader()，设定HTTP请求的头信息。
- 2: 表示send()方法已经执行，并且头信息和状态码已经收到。
- 3: 表示正在接收服务器传来的body部分的数据。
- 4: 表示服务器数据已经完全接收，或者本次接收已经失败了

## 第2章：跨域

### 2.1同源策略

同源策略(Same-Origin Policy)最早由 Netscape 公司提出，是浏览器的一种安全策略。

同源： 协议、域名、端口号 必须完全相同。 两个资源必须来自同一个服务。

违背同源策略就是跨域。

## 2.2如何解决跨域

### 2.2.1 JSONP

- 1) JSONP是什么

JSONP(JSON with Padding), 是一个非官方的跨域解决方案, 纯粹凭借程序员的聪明才智开发出来, 只支持get请求。

- 2) JSONP怎么工作的?

在网页有一些标签天生具有跨域能力, 比如: img link iframe script。

JSONP就是利用script标签的跨域能力来发送请求的。

- 3) JSONP的使用

#### 1.动态的创建一个script标签

```
var script = document.createElement("script");
```

#### 2.设置script的src, 设置回调函数

```
script.src = "http://localhost:3000/testAJAX?callback=abc";

function abc(data) {

    alert(data.name);

};
```

#### 3.将script添加到body中

```
document.body.appendChild(script);
```

#### 4. 服务器中路由的处理

```
router.get("/testAJAX" , function (req , res) {  
  
    console.log("收到请求");  
  
    var callback = req.query.callback;  
  
    var obj = {  
  
        name:"孙悟空",  
  
        age:18  
  
    }  
  
    res.send(callback+"("+JSON.stringify(obj)+")");  
  
});
```

## 2.2 CORS

### ■ 1) CORS是什么？

CORS（Cross-Origin Resource Sharing），跨域资源共享。CORS是官方的跨域解决方案，它的特点是不需要在客户端做任何特殊的操作，完全在服务器中进行处理，支持get和post请求。

### ■ 2) CORS怎么工作的？

CORS是通过设置一个响应头来告诉浏览器，该请求允许跨域，浏览器收到该响应以后就会对响应放行。

