

Ajax

1.1 Ajax技术简单介绍

Ajax是一门异步的用于发送网络请求的技术。

全称为：Async javascript and XML。

简单说明：Ajax 这个概念是由 Jesse James Garrett 在 2005 年发明的。它本身不是单一技术，是一串技术的集合。

技术特点：通常情况下，每次提交表单的时候，都会刷新整个界面，而使用ajax发送请求可以实现异步发送请求获取数据而不需要刷新整个页面。

Ajax的优点

- (1) 不需要插件支持(一般浏览器且默认开启 JavaScript 即可);
- (2) 用户体验极佳(不刷新页面即可获取可更新的数据);
- (3) 提升 Web 程序的性能(在传递数据的时候不必整体提交);
- (4) 减轻服务器和带宽的负担(将服务器的一些操作转移到客户端);

Ajax 的不足

- (1) 不同版本的浏览器对XMLHttpRequest 对象支持度不同(比如IE5之前);
- (2) 前进、后退的功能被破坏;
- (3) 搜索引擎的支持度不够(引擎爬虫还不能理解JS引起变化数据的内容);
- (4) 开发调试工具缺乏

1.2 Ajax发送GET请求

① 使用Ajax发送GET请求的基本步骤

- 01 创建请求对象 (XMLHttpRequest)
- 02 设置请求对象 (open) 请求方式为GET
- 03 发送请求 (send)
- 04 监听请求的状态 (onreadystatechange)
- 05 处理请求结果

② 使用Ajax发送GET请求的注意点

兼容性问题

```
//javascript创建请求对象的兼容性处理：  
if (window.XMLHttpRequest)  
{
```

```
var request = new XMLHttpRequest();
}else
{
    var request = new ActiveXObject("Microsoft.XMLHTTP");
}
```

缓存问题

问题说明：在IE浏览器中如果发送的是GET请求，只要URL没有发生变化，那么IE浏览器就会认为网页的内容也没有发生变化，因此会优先使用缓存数据。

解决方式：如果在IE浏览器中想让数据实时更新（获取到最新的数据），那么可以让每次请求的URL都不一样（即每次请求的URL地址不一样）。

具体思路：在开发中可以使用 随机数因子 或者是 时间戳 来添加一个额外的参数给url路径。

- 获取时间戳：var date = new Date();date.valueOf()
- 获取随机数因子 Math.random()

补充说明

① 加随机数或者是时间戳的目的在于让每次请求的时候url的内容都不一样

② 参数变化的作用就是让每次发送网络请求的时候URL都不相同，以让服务器总是把最新的数据返回。

监听请求状态

onreadystatechange的几种状态

- (1) 请求未初始化 - 0
- (2) 服务器连接已经建立 - 1
- (3) 请求已经接收 - 2
- (4) 请求处理中 - 3
- (5) 请求已经完成，且响应已经就绪 - 4

判断请求结果

网络请求发送之后，可能成功，也可能失败。在代码中，我们可以通过响应状态码来判断请求成功还是失败，并最终做出相应的处理。

- (1) 当请求完成的时候再进行处理，readyState == 4
- (2) 通过响应码判断请求成功还是失败，status == 200
- (3) 请求成功解析服务器返回的响应体：response.Text

请求超时

在开发中可能需要对请求的时间进行限制，例如要求设置网络请求的超时时间为10秒，如果超过了该时间那么就提示用户检查网络。

在具体的代码实现中，该需求可以使用定时器配合abort方法来实现。

中文转码

在发送GET请求的时候，如果请求路径中存在中文，那么在发送网络请求之前应该先对请求路径进行中文转码处理，使用 `encodeURIComponent` 方法来完成。

③ 使用Ajax发送GET请求的代码示例

```
// 数据处理方法
function json2str(data) {
    var arr = [];
    for(var key in data){
        arr.push(key+"="+data[key]);
    }
    // 不能直接将中文提交给服务器，中文需要编码之后再提交
    return encodeURIComponent(arr.join("&"));
}

// 发送请求方法
function myAjax(url, data, timeout, success, error) {
    // 0.对参数进行处理
    var res = json2str(data);

    // 1.创建异步对象
    if(window.XMLHttpRequest){
        var xhr = new XMLHttpRequest();
    }else{
        var xhr = new ActiveXObject("Microsoft.XMLHTTP");
    }
    // 2.设置URL
    xhr.open("get", url+"?" + res, true);
    // 3.发送请求
    xhr.send();
    // 4.监听状态
    xhr.onreadystatechange = function () {
    // 5.处理返回结果
        if(xhr.readyState == 4){
            // 清空定时器
            clearTimeout(timer);
            if(xhr.status == 200){
                success(xhr.responseText);
            }else{
                error(xhr.status);
            }
        }
    }
    // 6.对超时时间处理
    var timer = setTimeout(function () {
        alert("超时了");
        // 中断请求
        xhr.abort();
    }, timeout);
}
```

1.3 Ajax发送POST请求

① 使用Ajax发送POST请求的基本步骤

01 创建请求对象 (XMLHttpRequest)

02 设置请求对象 (open) 请求方式为POST

03 设置请求头信息

- xhr.setRequestHeader("Content-type","application/x-www-form-urlencoded");

04 发送请求 (send) , 参数以查询字符串的方式传递

05 监听请求的状态 (onreadystatechange)

06 处理请求结果

② 使用Ajax发送POST请求的代码示例

```
//001 创建请求对象
var xhr;
if(window.XMLHttpRequest)
{
    xhr = new XMLHttpRequest();
}else
{
    xhr = new ActiveXObject("Microsoft.XMLHTTP");
}

//02 设置请求方法和请求路径
xhr.open("post", "php/04-post.php", true);

//++ 设置请求头信息
xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
//03 发送请求
//++ 把提交的参数存放在请求体中提交
xhr.send("username=zs&password=123");

//04 监听网络请求的状态
xhr.onreadystatechange = function () {
    if(xhr.readyState == 4)
    {
        //05 解析服务器返回的数据
        if(xhr.status == 200)
        {
            console.log("请求成功" + xhr.responseText);
        }else
        {
            console.log("请求失败" + xhr.status);
        }
    }
}
}
```

- Posted by 博客园·文顶顶 | 花田半亩
- 联系作者 简书·文顶顶 新浪微博·Coder_文顶顶
- 原创文章，版权声明： 自由转载-非商用-非衍生-保持署名 | 文顶顶

