Ajax 使用

1. Ajax 概述

Ajax 即“**A**synchronous **J**avascript **A**nd **X**ML”（异步 JavaScript 和 XML），是指一种创建交互式、快速动态网页应用的网页开发技术，无需重新加载整个网页的情况下，能够更新部分网页的技术。

2. 传统网站中存在的问题

- 网速慢的情况下，页面加载时间长，用户只能等待
- 表单提交后，如果一项内容不合格，需要重新填写所有表单内容
- 页面跳转，重新加载页面，造成资源浪费，增加用户等待时间

上面的解决方案：Ajax 它是浏览提供的一套方法，可以实现页面无刷新更新数据，提高用户浏览网站应用的体验

3. Ajax 的应用场景

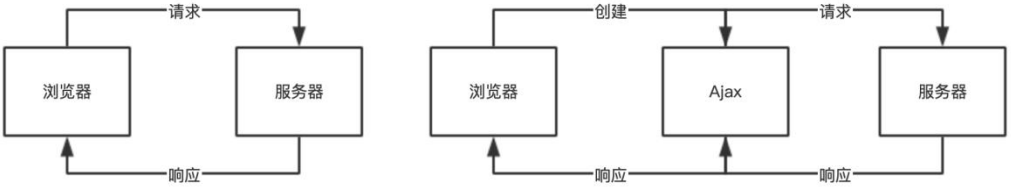
- 页面上拉加载更多数据
- 列表数据无刷新分页
- 表单页离开焦点数据验证
- 搜索框提示文字下拉列表
- 级连显示

4. Ajax 的运行环境

Ajax 技术需要运行在网站环境中才能生效

5. Ajax 运行原理

Ajax 相当于浏览器发送请求与接收响应的代理人，以实现在不影响浏览页面的情况下，局部更新页面数据，从而提高用户体验



6. Ajax 的实现步骤

1. 创建 Ajax 对象

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

2. 告诉 Ajax 请求地址以及请求方式

   ```
   xhr.open('GET','http://www.example.com')
   ```

3. 发送请求

   ```
   xhr.send()
   ```

4. 获取服务器端给客户端的响应数据

   ```
   xhr.onload = function(){
     console.log(xhr.responseText)
   }
   ```

样例：

```
var xhr = new XMLHttpRequest()
xhr.open('GET','http://httpbin.org/get')
xhr.send()
xhr.onreadystatechange = function(){

 if (xhr.readyState==4 && xhr.status==200){
  console.log(xhr.responseText)
}
```

7. Ajax 的 get 请求参数

在 get 请求中，参数是拼接在 url 中的，所以此时可以获取到参数，拼接到 url 即可

```
uname = document.getElementByID('uname')
pwd = document.getElementByID('pwd')
params = 'name='+uname+'&pwd='+pwd
url = '/getInfo?'+params
xhr.open('GET',url)
```

8. Ajax 的 post 的使用

Ajax 使用 post 的请求基本一样。但在传参时，有些不同，参数应该放在 body 中

8.1 str 的请求参数方式传递

```
uname = document.getElementByID('uname')
pwd = document.getElementByID('pwd')
params = 'name='+uname+'&pwd='+pwd
url = 'http://httpbin.org/post'
xhr.open('POST',url)
xhr.setRequestHeader('Content-Type','application/x-www-form-urlencoded')
xhr.send(params)
```

8.2 以下为 json 的方式传递

```
uname = document.getElementByID('uname')
pwd = document.getElementByID('pwd')
params = {'uname':name,'pwd':pwd}  //json 参数
args = JSON.stringify(params) //将 json 参数转成 string
url = 'http://httpbin.org/post?'
xhr.open('POST',url)
xhr.setRequestHeader('Content-Type','application/json')
xhr.send(args)
```

9. 获取服务器端的响应

9.1 Ajax 状态码

在创建 ajax 对象，配置 ajax 对象，发送请求，以及接收完服务器端响应数据，这个过程中的每一个步骤都会对应一个数值，这个数据就是 ajax，并且会触发 onreadystatechange 事件

- 0: 请求未初始化
- 1: 服务器连接已建立
- 2: 请求已接收
- 3: 请求处理中
- 4: 请求已完成，且响应已就绪

xhr.readyState *//获取 Ajax 状态码*

在 onreadystatechange 事件中，我们规定当服务器响应已做好被处理的准备时所执行的任务。

当 readyState 等于 4 且状态为 200 时，表示响应已就绪：

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

9.2 Ajax 的响应数据

服务器端响应的数据格式

在真实的项目中，服务器端大多数情况下会以 JSON 对象作为响应数据的格式。当客户端拿到响应数据时，要将 JSON 数据和 HTML 字符串进行拼接，然后将拼接的结果展示在页面中

在 http 请求与响应的过程中，无论是请求参数还是响应内容，如果是对象类型，最终都会被转换为对象字符串进行传输

JSON.parse() *//将 json 字符串转换为 json 对象*

10. Ajax 错误处理

网络畅通，服务器端能接收到请求，服务器端返回的结果不是预期的结果。**可以判断服务器端返回的状态码，分别进行处理。xhr.status 获取 http 状态码**

1. 网络畅通，服务器端没有接收到请求，返回 404 状态。**检查请求地址是否错误**
2. 网络畅通，服务器端能接收到请求，服务器端返回 500 状态码
3. 网络中断，请求无法发送到服务器端

```
xhr.onerror = function(){
  consolo.log('网络中断，无法发送 Ajax')
}
```

11. 同步与异步

同步，就是调用某个东西是，调用方得等待这个调用返回结果才能继续往后执行。

异步，和同步相反 调用方不会理解得到结果，而是在调用发出后调用者可用继续执行后续操作，被调用者通过状体来通知调用者，或者通过回掉函数来处理这个调用

比方说：

你去商城买东西，你看上了一款手机，能和店家说你一个这款手机，他就去仓库拿货，你得在店里等着，不能离开，这叫做同步。

现在你买手机赶时髦直接去京东下单，下单完成后你就可用做其他时间（追剧、打王者、lol）等货到了去签收就 ok 了.这就叫异步。

xhr.open("GET","test1.txt",**true**);