听着音乐 补充知识

Shayimerdan - Sozlerim ba
03:46

Ajax交互

欢迎您到IparhanGeek技术试验室,在这里我会把新的,旧的只要学过的技术点,框架语言还有详细到参数规格详细的剖析出来,因为我听到过一句话和坚信这句话"不会走路的想跑步总会跌倒的稀巴烂"， 不过我特别喜欢这个思维所以我决定吧把之前学过的和将要学习的全部关于编程的知识每天推送文章,这也可能是将是唯一能坚持做到底的事情了。

初探Ajax

说到Ajax没有一个开发人员不知道的吧,起码99% 的开发人员都熟悉也几乎每天都在看和使用(就算是维护),下面我们先来初探一下Ajax：

AJAX = Asynchronous JavaScript and XML（异步的 JavaScript 和 XML）。

可以看出是异步的架构,还有如果研究过底层的话可以看出原生是XML格式。

1

Ajax 工作流程

先看看原生的Ajax和工作流程：

工作流程：

1.创建对象

2.判断版本（老和新版本）

提示：这里主要是指浏览器的内核版本

3.监控跟服务器连接的状况

3.1：onreadystatechange：存储函数（或函数名），每当 readyState 属性改变时，就会调用该函数。

3.2：readyState：

存有 XMLHttpRequest 的状态。从 0 到 4 发生变化。

- 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

3.3：status：

200: "OK"

404: 未找到页面

4.发起请求(GET/POST)

PUT,DELETE,HEAD,OPTIONS,TRACE,CONNECT.

5.发起数据

```
function loadXMLDoc()
{
    //1.创建一个对象
    var xmlhttp;
    //2.如果是新版本的IE7以上
    if (window.XMLHttpRequest)
    {
        //  3.IE7+, Firefox, Chrome, Opera, Safari 浏览器执行代码
        xmlhttp=new XMLHttpRequest();
    }
    else
    {
        // 4.IE6, IE5 浏览器执行代码
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    //5.监控过程
    xmlhttp.onreadystatechange=function()
    {
        //6.判断是否跟服务器连接成功 就像TCP三次握手
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
            //7..服务器相应
            document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
        }
    }
    //8.发起请求
    xmlhttp.open("GET","/try/ajax/ajax_info.txt",true);
    //9.数据发送
    xmlhttp.send();
}
```

Ajax 面试题

1

Ajax 请求方式有几种和区别?

答案：GET,POST,PUT,DELETE,HEAD,OPTIONS,TRACE,CONNECT,

其中GET和POST常用:

GET：

1.数据发生在浏览器URL中

2.数据格式：http://localhost:8080/login?name=kaisar&password=

3.GET请求可以被缓存(cookie)-**安全级别高的场景不适应GET**

4.请求可以被浏览器收藏

5.有长度限制：IE4之前是4KB ， 之后是7-8KB

6.请求和相应速度比POST快

POST:

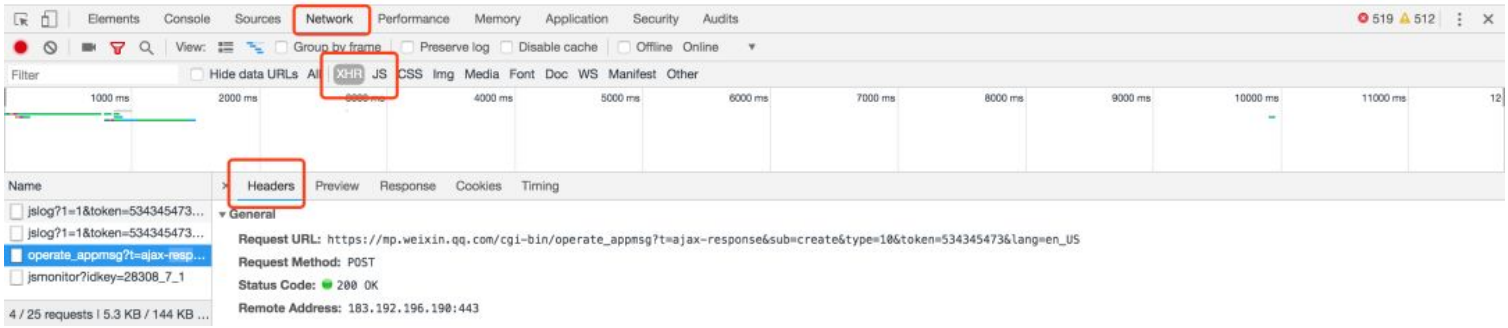
1.请求数据不会被自动缓存-手动可以存到Session

2.请求数据没有长度

3.数据安全级别高

4.交互数据速度慢

HEAD：



1.请求请求头信息

PUT DELETE：

1.请求覆盖(比如说使用API更新数据适合用PUT)

2.使用API删除数据可以使用DELETE

2.

什么情况下回跨域 和 解决方式?

答案：

1.不同域名之间传递数据

A: www.baidu.com B:www.wexin.com

2.子父级域名之间和子级域名之间跨域

A:www.blog.baidu.com B:www.blog.weixin.com

3.不同协议之间的传递数据

A: http://www.baidu.com B:https://www.weixin.com （虽然都是OSI应用层

4.IP和域名之间传递数据

A:47.254.33.173 B:iparhan.top

5.端口不同

A:iparhan.top:81 B:iparhan.top.82

跨域解决方案：

1.jsonp：添加响应头解决跨域

addHeader('Access-Control-Allow-Origin:*');//允许所有来源访问

addHeader('Access-Control-Allow-Method:POST,GET');//允许访问的方式

2.代理：

例如www.123.com/index.html需要调用www.456.com/server.php，可以写一个接口www.123.com/server.php，由这个接口在后端去调用www.456.com/server.php并拿到返回值，然后再返回给index.html，这就是一个代理的模式。相当于绕过了浏览器端，自然就不存在跨域问题。

3.业务里面解决：

3.1： 代码里解决

```
@Configuration
public class CorsConfig {
    private CorsConfiguration buildConfig() {
        CorsConfiguration corsConfiguration = new CorsConfiguration();
        corsConfiguration.addAllowedOrigin("*"); //允许任何域名
        corsConfiguration.addAllowedHeader("*"); //允许任何头
        corsConfiguration.addAllowedMethod("*"); //允许任何方法
        return corsConfiguration;
    }

    @Bean
    public CorsFilter corsFilter() {
        UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();
        source.registerCorsConfiguration("/**", buildConfig()); //注册
        return new CorsFilter(source);
    }
```

3.2：注解方式

```
//实现跨域注解
//origin="*"代表所有域名都可访问
//maxAge飞行前响应的缓存持续时间的最大年龄，简单来说就是Cookie的有效期 单位为秒
//若maxAge是负数,则代表为临时Cookie,不会被持久化,Cookie信息保存在浏览器内存中,浏览器关闭Cookie就消失
@CrossOrigin(origins = "*",maxAge = 3600)
public class UserController {
```

4.document.domain(JS)

```
//解决跨域
// URL http://a.com/foo
var ifr = document.createElement('iframe');
ifr.src = 'http://b.a.com/bar';
ifr.onload = function(){
    var ifrdoc = ifr.contentDocument || ifr.contentWindow.document;
    ifrdoc.getElementsById("foo").innerHTML);
};

ifr.style.display = 'none';
document.body.appendChild(ifr);
```

5.window.name(JS)

```
//window.name
// 这里是要传输的数据，大小一般为2M，IE和firefox下可以大至32M左右
// 数据格式可以自定义，如json、字符串
window.name = "这是a页面的内容";
setTimeout(function(){
    window.location.href= b.html;
    console.log(window.name);  //"这是a页面的内容"
},2000);
```

3

HTTP 常见状态码

1. 2XX： 2开头的说明请求成功
2. 4XX： 4开头说明请求错误,这个时候问题主要在客户端
3. 3XX： 3开头的大部分是重定向或者是数据空
4. 5XX： 5开头的客户端发请求成功但是服务器出问题

4

Ajax 我们为什么要使用?

答案:

1.页面无数新

2.节省宽带

3.Ajax是本身在客户端所以大部分对服务器压力较小

5

Ajax Callback函数

- onSuccess
- onFailure
- onUninitialized
- onLoading
- onLoaded
- onInteractive

- onComplete
- onException

6
XMLRequest 介绍

Ajax的核心是JavaScript对象XmlHttpRequest。该对象在Internet Explorer 5中首次引入，它是一种支持异步请求的技术。简而言之，**XmlHttpRequest使您可以使用JavaScript向服务器提出请求并处理响应，而不阻塞用户。通过XMLHttpRequest对象，Web开发人员可以在页面加载以后进行页面的局部更新**

- **open()**(**String method,String url,boolean asynch**,String username,String password)
- **send(content)**
- **setRequestHeader(String header,String value)**
- getAllResponseHeaders()
- getResponseHeader(String header)
- abort()

- open()：该方法创建http请求
  - 第一个参数是指定提交方式(post、get)
  - 第二个参数是指定要提交的地址是哪
  - 第三个参数是指定是异步还是同步(true表示异步，false表示同步)
  - 第四和第五参数在http认证的时候会用到。是可选的
- setRequestHeader(String header,String value)：设置消息头（使用post方式才会使用到，get方法并不需要调用该方法）
  - xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");
- send(content)：发送请求给服务器
  - 如果是get方式，并不需要填写参数，或填写null
  - 如果是post方式，把要提交的参数写上去

FORM 表单提交

表单提交参考图：

```
<div id="small-dialog" class="mfp-hide book-form">
    <h3>注册表单 </h3>
        <form id="data" action="#" method="post">
            <input type="text" name="Name" placeholder="请输入用户名" required=""/>
            <input type="text" name="Email" class="email" placeholder="请输入邮件" required=""/>
            <input type="password" name="Password" class="password" placeholder="请输入密码" required=""/>
            <input type="password" name="Password" class="password" placeholder="请重复密码" required=""/>
            <input type="submit" submit="register()" value="立即注册">
        </form>
</div>
```

提示：

1.form表单提交现在用的比较少

2.form没法无刷新

3.适合小量数据传递和测试

END

IparhanGeek

海赛尔

提示：

每篇文章里面的每一个字都是我经过实战或者实际项目研究的结果,希望如果觉得能帮到您可以赞赏一下起码关注也可以，谢谢！

Scan the QR code to add me on WeChat

IparhanGeek