

Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心



Ajax技术

硕士研究生 喻露

2017年10月29日

内容提要



- 背景简介
- 基本概念
- Ajax原理
- Ajax实现
- 优劣分析
- 应用场景

- 2005年2月，Adaptive Path 的Jesse James Garrett最早创造了这个词(Ajax: A New Approach to Web Application)
- 2006年google公司推出了google maps地图检索系统，引爆了web开发的新浪潮——Ajax





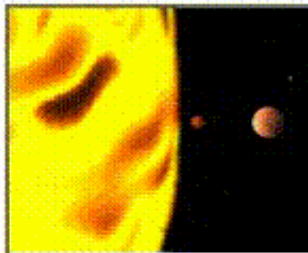
基本概念



- 什么是Ajax
 - Asynchronous JavaScript XML (异步JS+XML)
 - 创建交互式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
- Ajax技术包括
 - HTML CSS
 - DOM
 - XML
 - JavaScript
 - XMLHttpRequest (XHR)

- 同步与异步
 - 同步
 - 发送方发出数据后，等接收方发回响应以后才发下一个数据包的通讯方式
 - 异步
 - 发送方发出数据后，不等接收方发回响应，接着发送下个数据包的通讯方式
 - 例子
 - 传统B/S模式（同步）
 - 提交请求→等待服务器处理→处理完毕返回（在此期间浏览器不能干任何事）
 - Ajax技术（异步）
 - 请求通过事件触发→服务器处理（浏览器仍可做其他事情）→处理完毕

- JavaScript
 - 脚本语言，主要用来向HTML页面添加交互行为
 - 事件驱动，不需要经过Web服务器就可以对用户的行为做出响应
- JavaScript用途
 - 对用户行为作出响应
 - 操作DOM对象
 - 在数据被提交到服务器之前验证数据



鼠标在太阳和星星上移动，可以看到不同的描述。

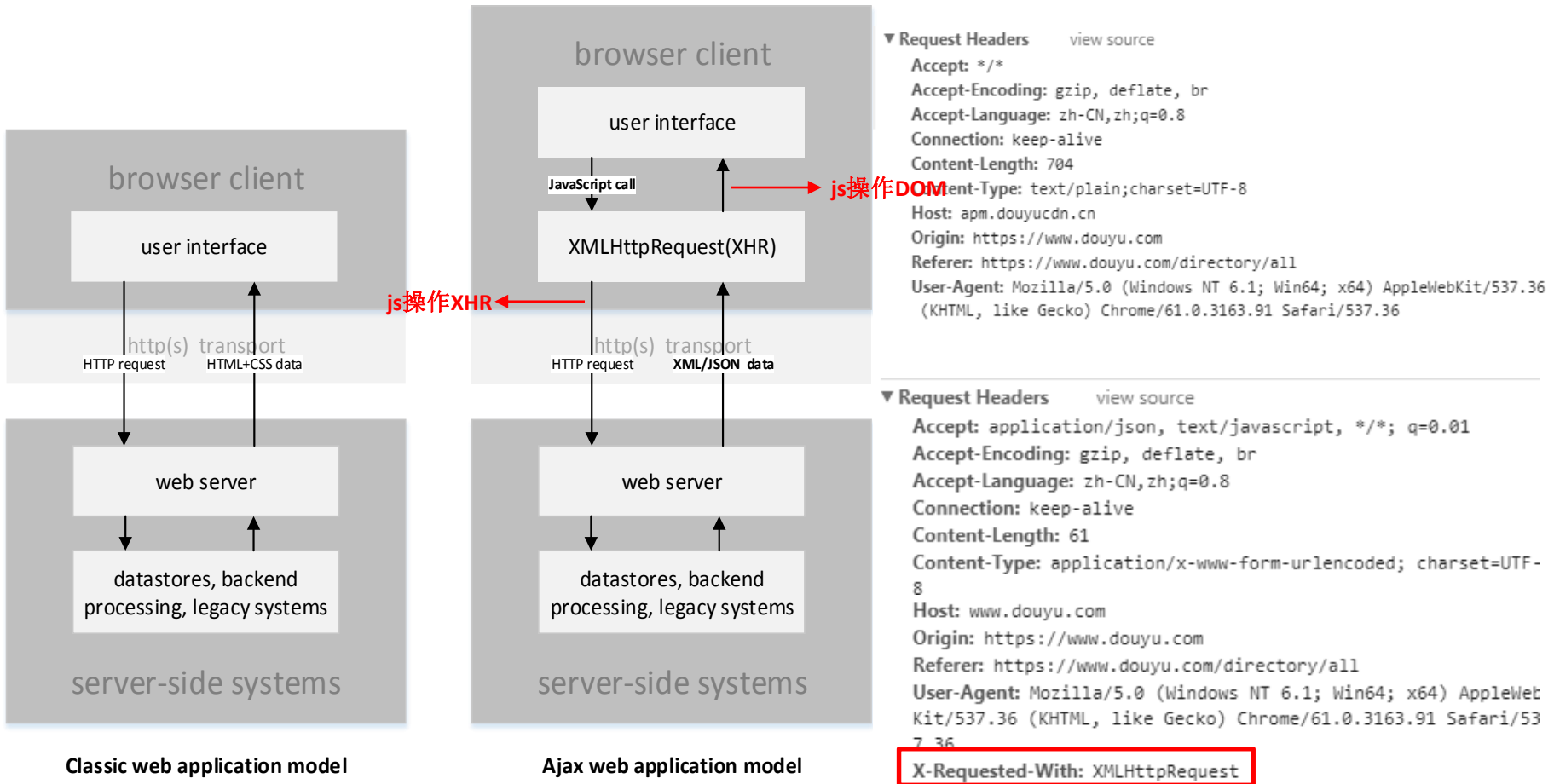
Ajax原理



Ajax原理



- web应用程序模型对比



Ajax实现



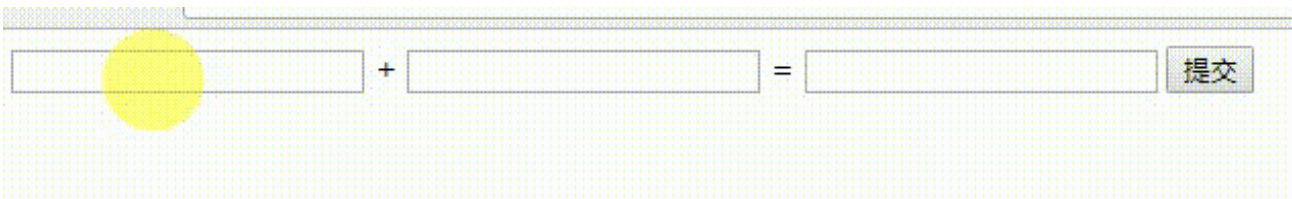
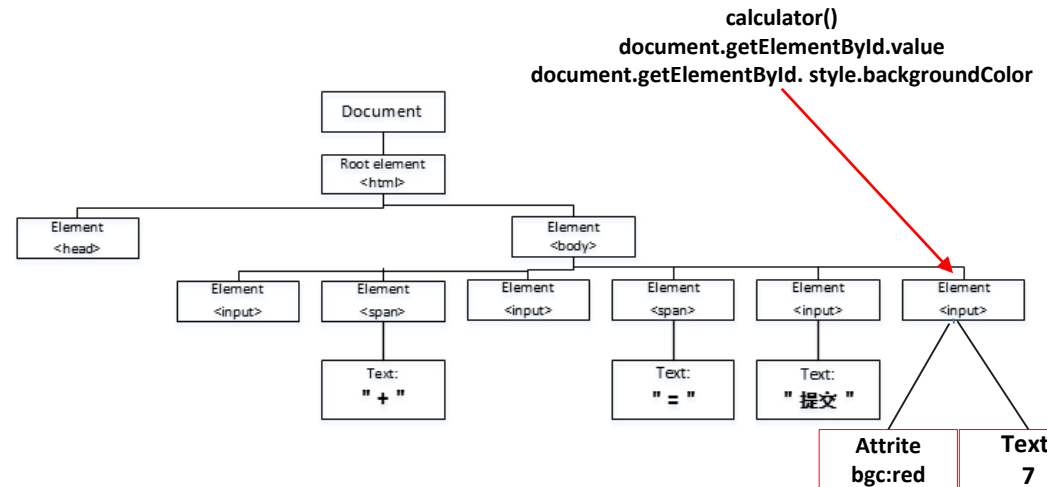
Ajax实现

- JS操作DOM

- DOM

- 文档对象模型 (Document Object Model)
 - DOM定义了表示和修改文档所需的对象、这些对象的行为和属性以及这些对象之间的关系
 - 通过JS直接修改DOM对象

```
<html>
<head>
  <meta charset="UTF-8">
  <title>计数器</title>
</head>
<body>
  <input type="text" name="text" id="pre" onblur="validate(this.value);">
  <span id="operator"> + </span>
  <input type="text" name="text" id="next" onblur="validate(this.value);">
  <span></span>
  <input type="text" id="result" readonly="true">
  <input type="button" id="btn" value="提交" onclick="calculator();">
</body>
</html>
```



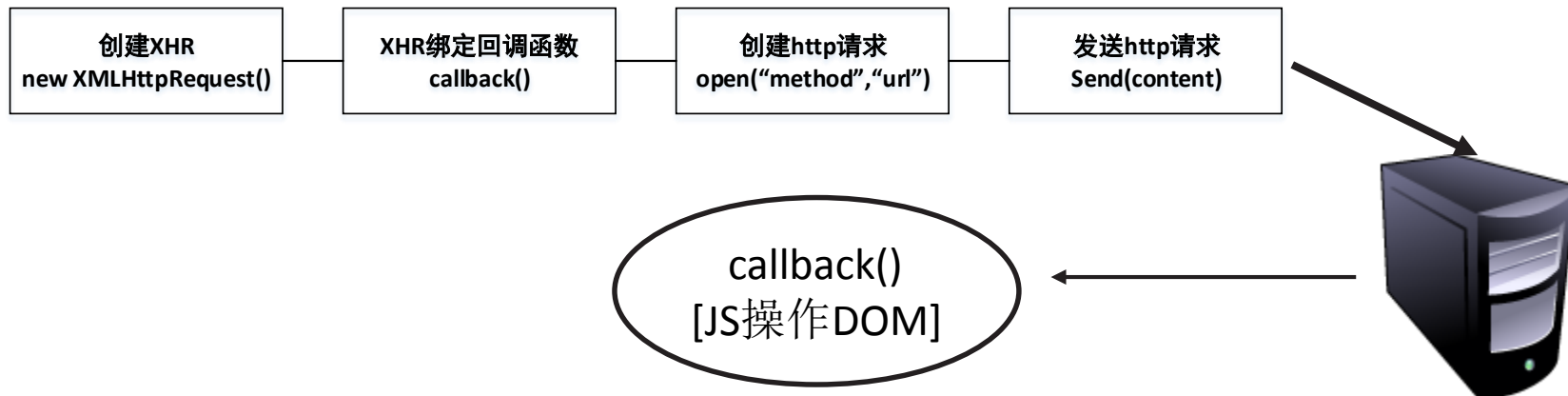
- JS操作XHR
 - XHR (XMLHttpRequest)
 - 用于在后台与服务器交换数据，这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

方法	open("method","url", "async")	建立对服务器的调用。Method参数可以是GET、POST。url参数可以是相对URL或绝对URL。async可选参数可以是true、false。这个方法还包括2个可选的参数
	send(content)	向服务器发送请求
属性	responseText	服务器的响应，表示为一个串
	responseXML	服务器的响应，表示为XML

- JS操作XHR

- 过程

- 创建XHR对象
 - 绑定http请求的事件响应函数（回调函数）
 - 创建http请求
 - 发送http请求



- 基本步骤

- 要完整实现一个AJAX异步调用和局部刷新, 通常需要以下几个步骤

1. 创建XMLHttpRequest对象, 也就是创建一个异步调用对象.
2. 设置响应HTTP请求状态变化的函数.
3. 创建一个新的HTTP请求, 并指定该HTTP请求的方法、URL及验证信息
4. 发送HTTP请求.
5. 获取异步调用返回的数据.
6. 使用JavaScript和DOM实现局部刷新.

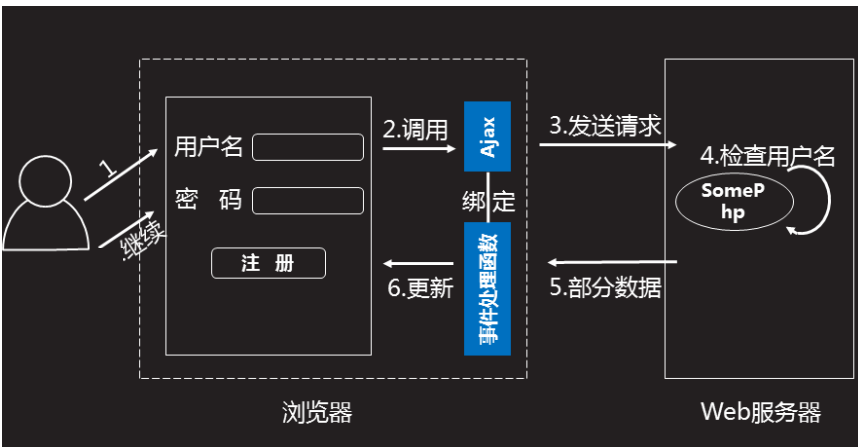
- 基本步骤
- 举例

```
<script type="text/javascript">
function httpStateChange() //响应XMLHttpRequest对象状态变化的函数
{
    if (xmlHttp.readyState==4 && xmlHttp.status==200)
    {
        var userNames = xmlHttp.responseText; ⑤
        var arrUserName = userNames.split(";"); //把获取到的一个字符串分割成字符串数组。
        var bFlag = false; //定义一个变量，用于判断用户名是否已经存在
        for(i=0;i<arrUserName.length;i++)
        {
            if (arrUserName[i]==myForm.myName.value)
            {
                bFlag = true; //用户名存在
                break;
            }
        }
        var node = document.getElementById("myDiv"); //查找用于显示提示信息的节点
        if (bFlag) //更新数据
        {
            node.firstChild.nodeValue = "此用户名太受欢迎,请更换一个"; ⑥
            myForm.submitButton.disabled = true; //提交按钮不可用
        }
        else
        {
            node.firstChild.nodeValue = "用户名不存在,可以使用";
            myForm.submitButton.disabled = false; //提交按钮可用
        }
    }
}
//校验用户名是否有效
function checkName()
{
    //创建XMLHttpRequest对象 | ①
    var xmlHttp = null;
    xmlHttp = new XMLHttpRequest();
    //创建响应XMLHttpRequest对象状态变化的函数
    xmlHttp.onreadystatechange = httpStateChange; ②
    //创建http请求
    xmlHttp.open("get","userName.txt", true); ③
    //发送http请求
    xmlHttp.send(null); ④
}
}</script>
```



用户名

设置后不可更改
中英文均可，最长14个英文或7个汉：





优劣分析

- 优势
 - 无刷新更新数据
 - 异步与服务器通信
 - 前端和后端负载平衡
 - 基于标准被广泛支持
- 劣势
 - 无Back和History功能，破坏浏览器机制
 - 客户端过肥，太多客户端代码造成开发上的成本
 - 违背URL和资源定位的初衷
 - 对搜索引擎支持较弱



应用场景

- Ajax 的特点在于异步交互，动态更新 web 页面，因此它的适用范围是交互较多，频繁读取数据的 web 应用。
 - 数据验证
 - 按需取数据
 - 自动更新

谢谢!

大成若缺，其用不弊。大盈若冲，其用不穷。大直若屈。大巧若拙。大辩若讷。静胜躁，寒胜热。清静为天下正。

