Ajax（异步javascript和xml）

服务器软件：

apache：开源的web服务器软件，免费，目前使用人数最多

IIS：微软的web服务器，之前不开源，目前开源

tomcat：web服务器软件

Nginx：web服务器软件

ajax也分get和post

后端决定ajax提交方式（get/post）、数据变量名

前端写请求报文

ajax前端部分

1. 注册事件函数
2. 创建XMLHttpRequest对象
3. 设置请求行
4. 设置请求头
5. 注册回调函数
6. 发送请求

ajax后端部分

1. 接受提交的数据
2. 处理请求
3. 返回响应报文

ajax本质：使用js和xml异步发送http请求和处理返回的数据

通过触发js事件函数，创建xmlhttprequest对象（该对象模拟http协议的get和post请求）发送该模拟请求报文到服务器，服务器处理完数据发送响应报文回浏览器，再通过js回调函数处理返回的数据展示到浏览器上

XMLHttpRequest对象

//创建xmlhttprequest对象

var xhr=XMLHttpRequest();

//设置请求行，参数1：提交方式：get/post，参数2：url，参数3：是否为异步，默认为true异步

xhr.open(‘get/post’, ‘url?key=value/url&key2=value2’, true/false);

//回调函数用onreadystatechange**①**所有浏览器都支持

xhr.onreadystatechange = function() {

if (xhr.readyState==4&&xhr.statu==200) {

数据处理代码;

}

}

//发送请求报文，若方式为post则写入键值对，否则为空或null

xhr.send(null/ ’key=value&key2=value2’);

注：①onreadystatechange事件所有浏览器都支持，但是需要写判断，readyState状态码分为五种：0/1/2/3/4，其中4为完成状态，所以判断if xhr.readyState==4，但是只判断readyState还不够，还需要判断xhr.status==200（页面存在且成功连接）再执行数据处理代码

ajax发送get请求的前端部分

document.querySelector(‘xxx’).onclick = function() { //请求触发事件函数

var xhr=new XMLHttpRequest(); //创建异步对象

xhr.open(‘get’, ’url?name=xxx&age=20’); //设置请求行，get，get请 求将数据拼接在url中

xhr.setRequestHeader(‘cookie’, ‘xxxxxxxx’); //请求头setRequestHeader， ‘参数1’，‘值1’，get请求可以省略设置请求头

xhr.onload=function() { //注册回调函数，请求响应回来后触发，处理 返回的数据

console.log(‘请求响应回来了’);

xhr.responseText; //取用返回的数据，字符串格式

}

xhr.send(null) //请求主体发送，不发送数据直接空或者null

}

ajax发送post请求的前端部分

document.querySelector(‘xxx’).onclick = function() { //请求触发事件函数

var xhr=new XMLHttpRequest(); //创建异步对象

xhr.open(‘post’, ‘url’); //设置请求行，post，url

xhr.setRequestHeader(‘Content-type’, ‘application/x-www-form-urlencode’);

//设置请求头，让ajax像html表单那样发送数据

xhr.onload = function() { //注册回调函数，处理返回的数据

console.log(‘xxx’);

}

xhr.send(‘key=value&key2=value2’); //请求主体发送，将数据写在send 中，格式为字符串‘key=value&key2=value2’内容不要加空格

}

服务器端返回复杂数据

1. xml格式：xxx.xml，相当于一个document

语法：

<?xml version=’1.0’ encoding=’UTF-8’> //声明，可不写

<root> //根标签，必须有，名字自定义

<name>xxx</name>

<age>20</age>

...

</root>

注：

1. xml内标签必须成对
2. 标签名随意，但是最好有意义，类似于编程语言内的变量名

前端解析xml文件：

1. xhr.responseXML.querySelector(‘xxx’)
2. var xxx=标签.innerHTML //对于大量数据需要使用循环进行解析
3. 创建标签来装解析出来的数据

XML缺点：

传输的数据量大，但有效数据比不高

解析复杂

1. json格式：