

AJAXday01

ajax 的异步加载

一.HTTP 协议

1.url

结构: 协议+主机名称+目录结构+文件名

http://www.codeboy.com/admin/user_list.html

url 完整结构

<scheme>://<user>:<pwd>@<host>:<port>/<path>;<params>?<query>#<frag>

scheme 方案、协议, 以哪种方式获取浏览器的资源

不区分大小写, 常见协议, http/https/ftp

常见协议	默认端口号	基本作用
ftp	21	文件上传下载
SSH	22	安全远程登录
TELNet	23	远程登录
SMTP	25	邮件传输协议
DNS	53	域名解析
HTTP	80	超文本传输
POP3	110	邮件接收
HTTPS	443	加密传输 http

<scheme>://<user>:<pwd>@<host>:<port>/<path>;<params>?<query>#<frag>

user: 用户名

pwd: 密码

host: 主机名 localhost 127.0.0.1

port: 端口号

path: 资源在服务器上存放的位置

params: 参数, 服务器端使用。某些方案会使用参数来指定输入的值, 每个参数都采用“名/值”形式

query: 查询字符串, 某些方案会使用查询字符串传递参数。

frag: 锚点

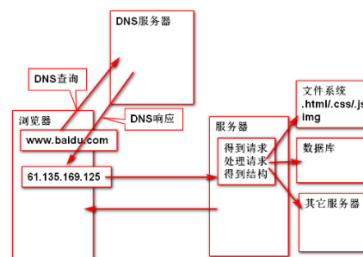
2.HTTP 协议

HTTP:HyperText Transfer Protocol 超文本传输协议, 规范了数据是如何打包及传递的。

http 的历史 现在使用的版本是

http/1.1 不需要设置

3.详解



HTTP 协议规定, 客户端与服务器端传递的数据块 Message (消息/报文)。必须符合特定的格式, 这个 Message 才能被客户端和服务端彼此理解



- 1.请求消息(请求起始行, 请求头, 请求主体), 浏览器要发送给服务器的数据
- 2.响应消息(响应起始行, 响应头, 响应主体), 服务器发送给浏览器的数据

Request 请求消息

客户端要发送给服务器的数据

由三部分组成 请求起始行, 请求头, 请求主体

1.请求起始行

-请求方法

-请求 url

-协议版本

请求方法:

get 客户端向服务器要数据的时候使用
靠地址栏传递查询字符串, 无请求主体

post 客户端向服务器提交数据的时候使用
隐式发送。有请求主体(form data)

delete 客户端删除服务器上的内容(一般禁用)

put 客户端往服务器上放数据(一般禁用)

有请求主体

connect 测试连接

trace 追踪请求路径

option 选项

head 表示客户端只回去响应的消息头

url 访问的地址链接:

协议版本 http/1.1

<p>2.请求头 Request Headers</p> <p>host:告诉服务器，请求的是哪一个主机</p> <p>Connection: keep-alive 告诉服务器进行持久连接</p> <p>User-Agent:告诉服务器自己(浏览器)的类型</p> <p>Accept-Encoding:gzip,deflate,告诉服务器，自己能接收的压缩文件类型</p> <p>Accept-Language:告诉服务器自己能接收的自然语言类型</p> <p>Referer:引用/推荐人，告诉服务器请求来自哪个网页</p>
<p>3.请求主体 Form data</p>

Response 响应消息

<p>服务器端发送给客户端的数据，由三步组成：响应起始行，响应头，响应主体</p>
<p>1. 响应起始行</p> <p>协议版本 http/1.1</p> <p>响应状态码</p> <p>原因短句，对状态码的解释说明</p>
<p>响应状态码</p> <p>告诉客户端浏览器，服务器的响应状态是什么取值：</p> <p>1XX:100-199 提示信息</p> <p>2XX:成功响应 200 ok</p> <p>3xx: 301 永久重定向 www.sun.com</p> <p>302 临时重定向</p> <p>304 请求未被修改，还是原来的请求，命中缓存</p> <p>4XX: 客户端请求的错误</p> <p>404 NOT Found 请求资源不存在</p> <p>403 Forbidden 权限不够</p> <p>405 Method not Allowed 请求方法不被允许</p> <p>5XX: 服务器运行错误</p>
<p>原因短句，就是对状态码的解释</p>

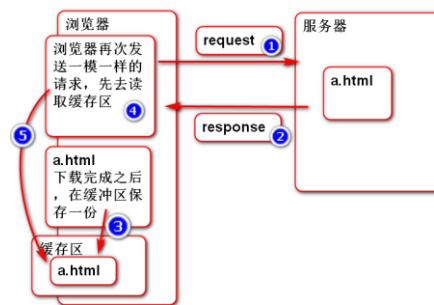
<p>2.响应头</p> <p>Date 告诉浏览器，服务器的响应时间，格林威治时间 +8h 是北京时间</p> <p>Connection:keep-alive 告诉浏览器已经启动了持久连接</p> <p>Content-Type:响应主体的类型：</p> <p>text/html 响应的数据是 html 文本</p> <p>text/plain 普通文本</p> <p>text/css 样式文件</p>

<p>application/javascript js 文件</p> <p>image/png(jpeg.gif) 图片</p> <p>application/json 响应回来的是 json 格式的字符串</p> <p>application/xml 响应回来的是 xml 格式的字符串</p>
<p>3.响应主体 响应回来的数据</p> <p>network 中，在 header 的右边第二个</p>

缓存

1. 什么是缓存

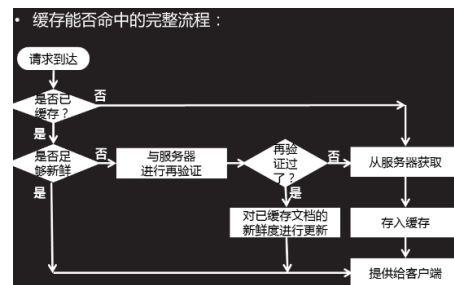
客户端将服务取得响应后自动保存，再次访问时，直接使用保存数据



2.缓存的优点

- 1.减少冗余的数据传输，节省客户端的流量
- 2.节省服务器带宽
- 3.降低了对服务资源的消耗和运行需求
- 4.降低了远距离传输而造成的延迟加载

缓存的原理



- 1.请求--无缓存--访问服务器--存缓存--客户端得到数据
- 2.请求--有缓存--够新鲜--命中缓存--客户端得到数据
- 3.请求--有缓存--不新鲜--访问服务器看是否过期--没过期--更新缓存的新鲜度--客户端得到数据
- 4.请求--有缓存--不新鲜--访问服务器看是否过期--已过期--访问服务器--存缓存--客户端得到数据

3.与缓存相关的消息头

Cache-Control:http1.1 的用法
从服务器将文档传到客户端之时起，
此文档处于新鲜的秒数，这个秒数是一个相对时间
语法：0 不缓存
max-age= 新鲜的秒数
Expires http1.0 用法
指定过期的确切时间，是一个具体的时间点

4.网页如何添加缓存

```
<meta http-equiv="消息头" content="消息头对应的值">
```

例如：

```
<meta http-equiv=" Cache-Control " content="max-age=3600">
```

5.Http 性能优化

1.http 连接的过程
发起请求-->建立连接-->处理请求-->访问资源-->构建响应-->发送响应-->记录日志

2.http 连接性能优化的思路

- 2.1 减少请求发送次数(合理设置缓存)
- 2.2 减少连接创建次数(开启持久连接)
- 2.3 提高服务器端的运行速度
- 2.4 尽可能减少响应数据的长度

3.安全的 HTTP 协议
HTTPS，安全版本的 http
SSL：为数据通信提供了安全支持
a.客户端发送请求消息时，在 SSL 层加密
服务器接收到加密文件，在 SSL 层解密
得到请求明文，对请求进行处理
b.服务器发送响应消息时，在 SSL 层加密
客户端接收到加密文件，在 SSL 层解密
得到响应明文，解析相应内容

二.DOM 操作

1.完整的 javascript 操作的组成

a.js 核心：ECMA Script ECMA--欧洲计算机制造联合会
b.DOM: Document Object Model 文档对象模型
让 js 动态的操作页面上的元素
c.BOM: Browser Object model 浏览器对象模型
让 js 动态的操作浏览器

DOM，这个阶段，获取页面上的元素，
改变这个元素的内容

2.简单 DOM 操作

a.要获取的元素一定要有 id
<div id="d1">这是测试的文本!! </div>

b.在 js 中，用 id 获取对象
document.getElementById("d1");
H5 支持简写方式
d1 就是 div 对象

c.获取/设置元素的内容 innerHTML
d1.innerHTML=...
var str=d1.innerHTML;
innerHTML 指的是，双标签，开关标签之间的所有内容。
如果开关标签之间有其它元素,会得到该元素的文本字符串
如果设置 innerHTML 时，添加了新标签，在语法正确的情况下，新标签也会在页面上渲染

d.事件 用户与页面交互行为，调用什么 js 代码

作业：

在网页中创建一个 div，内容随意
添加一个按钮，点击是获取 div 中文本并打印在控制台
再添加一个按钮，点击是将 div 中文本内容修改

在网页中创建一个 div。内容随意
添加一个按钮，在 div 中动态生成一个 3*3 的，内容随意的表格

AJAXday02

一.DOM 操作

1.获取元素的对象

直接使用这个元素的 id 值

2.获取/修改元素的内容(只有双标签叫内容)

d1.innerHTML
可以获取/修改内容文本
还可以动态的添加元素

3.获取/设置表单控件的数据

1.使用 input 控件的 id 代表控件对象
2.通过对象的 value 属性，获取/设置控件的值
<input id="uname" type="text">
uname.value="1000";

4.HTML 元素的事件

允许通过用户的行为来激发的操作就是事件
常用的事件 onclick 用户点击元素, 激发某项操作

文本框/密码框, 失去焦点事件 onblur
文本框/密码框, 获得焦点事件 onfocus

body 的事件 onload
当页面加载完成, 马上要激发的操作

二.AJAX

同步: Synchronous

在一个任务进行过程中, 不能开启其它任务。

同步访问:

浏览器在向服务器发送请求时, 浏览器只能等待服务器的响应, 不能做其它事情。

出现场合:

- 1.地址栏输入网址并跳转
- 2.a 标记跳转
- 3.submit 表单的提交

异步 Asynchronous

在一个任务开启的同时, 可以开启其它任务。

异步的访问: 浏览器在向服务器发送请求时, 不耽误用户在网页上的其它操作

出现的场合:

- 1.用户名的重复验证
- 2.聊天室
- 3.搜索建议(百度, 京东, 淘宝)

1.什么是 AJAX

Asynchronous JavaScript And Xml

异步的 js 和 xml

本质: 使用 js 提供的异步对象

(XMLHttpRequest---xhr)

异步的向服务器发送请求,
并接收响应回来的数据。

原理: ajax 中, 服务器响应回来的是部分数据, 不是完整的页面, 所以可以无刷新效果, 去更改页面的布局结构和内容。相当于在用户端和服务器间加了一个中间层, 利用客户端闲置的处理能力来处理数据。

2.创建异步对象

①标准创建 var xhr=new XMLHttpRequest()

②IE8 以下的创建 (了解)

var xhr=new ActiveXObject("Microsoft.XMLHttp");

如何判断用户的浏览器版本?

使用 window.XMLHttpRequest 属性

如果是高版本, 这个属性有值

如果浏览器不支持标准创建, 这个属性为 null。

```
if(window.XMLHttpRequest){
    var xhr=new XMLHttpRequest();
}else{
    var xhr=new ActiveXObject("Microsoft.XMLHttp");
}
```

3.异步对象的常用属性和方法

abort()取消请求

getAllResponseHeaders()获取所有响应头

getResponseHeader()获取指定响应头

open()创建请求

send()发送请求

get 方法 xhr.send(null);

post 方法 xhr.send(formdata);

xhr.responseText 服务器返回文本

xhr.responseXML 服务器返回 xml 文本

readyState 表示 xhr 对象的请求状态

取值 0-4 表示着 5 个状态:

0: 请求尚未初始化

1: 打开服务器连接, 正在发送请求

2: 接收响应头

3: 接收响应主体

4: 接收响应数据成功

当 readyState 的值为 4 的时候, 才表示所有的数据接收完毕

onreadystatechange 当 xhr 的 readyState 属性发生改变时触发

status 表示服务器的响应状态码

只有在 status=200, 表示服务器已经正确处理了请求

4.使用 ajax 的步骤

1.创建 xhr 对象

var xhr=new XMLHttpRequest();

2.绑定监听

```
xhr.onreadystatechange=function(){
    if(xhr.readyState==4 &&
    xhr.status==200){
        var result=xhr.responseText;
    }
}
```

```

3.打开链接, 创建请求
xhr.open(method,url,true);
method 必须是字符串 "get" "post"
url="127.0.0.1:3000/list?uname="+name+"&upwd="+pwd;
4.xhr.send();

```

01_ajax.html 练习



服务器部分步骤

- 1.在 xz 文件夹中, 把 routes 文件夹中的 user.js 复制。改名为 demo.js.
- 2.创建接口 ajaxDemo
//1.ajaxDemo
router.get("/ajaxDemo",(req,res)=>{
 res.send("这是我的第一个 ajax");
});
- 3.在 app.js 中把 demo.js 导入
const demo=require("./routes/demo.js");
server.use("/demo",demo);
- 4.启动服务器, 在地址栏输入网址, 验证
http://127.0.0.1:3000/demo/ajaxdemo
- 5.在 xz 中创建文件夹 myex.并在 app.js 中托管静态资源
server.use(express.static('myex'));
- 6.在 myex 中创建文件 01_ajax.html 中编写 ajax 代码。

前端代码

```

<input type="button" value="ajax"
onclick="getMsg()">
<script>
function getMsg(){//尝试传递参数
//1.创建 xhr 对象
var xhr=new XMLHttpRequest();
//2.绑定监听器
xhr.onreadystatechange=function(){
if(xhr.readyState==4&&xhr.status==200){
var result=xhr.responseText;

```

```

console.log(result);
}
}
//3.打开链接, 创建请求
xhr.open("get","/demo/ajaxdemo",true);
//4.发送请求
xhr.send();
}
</script>

```

5.使用 get 发送带参数的请求

```
url="demo/login?uname=value1&upwd=value2"
```

练习:

- 把 01_ajax.html 以及对应接口/ajaxDemo, 删除, 整体重新敲一遍
- 把 02_login.html 以及对应接口/login, 删除, 整体重新敲一遍

作业

- 1.在 demo.js 中, 创建一个接口 userList
在接口中查询出用户表所有的数据, 并响应给前端页面
select * from xz_user
创建一个网页 03_getList.html.
网页中有一个按钮, 点击后, 获取 userList 接口的数据
- 2.在 demo.js 中, 创建接口 checkUname.模拟学子商城的注册需求。
文本框失去焦点时, 使用 ajax 连接 checkUname.查询数据库是否存在该用户名
如果存在, 响应给页面"用户名被占用"
如果不存在, 响应给页面"用户名可用"
创建网页 04_register.html。一个文本框, 一个密码框, 文本框失去焦点的时候调用连接服务器的 ajax 方法

AJAXday03

一.ajax

1.post 请求

```

1.post 请求的数据放在请求主体中
var
formdata="uname="+uname.value+"&upwd="+upwd.value;
xhr.send(formdata);

```

2.发送请求之前, 要手动修改请求消息头
`xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");`
默认的消息头 `Content-Type:text/plain`,传递的数据是普通的, 无特殊符号的字符, 主体中有特殊符号"&" 必须要把值设置成可以有特殊符号的字符类型 `application/x-www-form-urlencoded`
注意: 设置请求消息头, 放在 `xhr.open` 和 `xhr.send` 之间

二.JSON 数据格式

1.js 对象的数据格式

```
var stu={name:'lilei',age:20,gender:'男'}
```

练习

创建一个笔记本对象 `computer`, 包含以下三个属性

- 1.name 的值为 MacBook Air
- 2.price 7800
- 3.left 200

在控制台按照以下格式打印出结果

笔记本名称 MacBook Air
笔记本的价格 7800 元

```
var computer={
  name:'MacBook Air',
  price:7800,
  left:200
}
```

2.JSON 的数据格式

1.什么是 Json
JavaScript Object Notation
js 对象 表现方式
js 对象表示法, 以 js 对象的数据格式表现出来的字符串

2.Json 的语法

- 1.json 中用一对{}来表示一个对象
- 2.对象中的属性名称必须用"引起来(不要用单引号)
如果值是字符串, 也必须用"引起来
2. json 表现出来是一个字符串, 所以最外层添加单引号'

js	json
<pre>var computer={ name:'MacBook Air', price:7800, left:200 }</pre>	<pre>var computer='{ "name":"MacBook Air", "price":7800, "left":200 }'</pre>
<pre>["tom","lilei","hanmeimei"]</pre>	<pre>'[{"name":"tom","age":18},{"name":"lilei","age":20},{"name":"hanmeimei","age":19}]'</pre>

Json 是一个以 js 对象为格式字符串, 不能以 对象.属性 的方式获取值。需要先把 json 字符串转化成 js 对象, 才能使用

使用 `JSON.parse(json 串)` 将 json 字符串解析为 js 对象/js 数组。
可以在前端页面中, 使用 js 语法操作对象/数组, 再使用 dom 把内容动态的加载到 html 中

提高练习

把数据库中所有的 xz_user 中所有的数据, 都动态放到 html 的 table 中

三.xml 数据格式

什么是 XML
eXtensible Markup Language
可扩展 标记 语言
xml 的标记是没有被预定义过的, 需要自行定义
xml 的宗旨, 是做数据的传递, 而不是数据展示

语法规范

1.xml 的最顶端做声明
`<?xml version="1.0" encoding="utf-8"?>`

2.XML 标记的语法

- 必须成对出现
- xml 中, 标签严格区分大小写
- xml 允许嵌套, 但注意嵌套顺序
- 每个标记允许自定义属性, 格式与 html 一致, 但是值必须带""
- 每个 xml 文档, 有且只有一个跟标签

练习:

创建一个 student.xml 的文件
在文件中声明 xml 的类型, 然后 添加一对跟标记 `<studentlist>`
在根标记中添加 3 个学生信息(student)
每个学生有 name age gender 内容自拟

ajax 获取 xml 数据
使用 var result=xhr.responseXML;获取 xml 的对象
var
students=result.getElementsByTagName("student");
var
names=students[0].getElementsByTagName("name");
names[0].innerHTML
这种 dom 操作获取的数组, 叫类数组
类数组可以使用下标查找元素, 但是不能使用数组的 API

四.项目阶段

1.登录

接收前端传递过来的用户名和密码
将用户名和密码取数据库查询
并给出提示 登录成功/用户名密码错误
1.myPro.js 中写 login 接口
2.myPro 文件夹中创建文件 01_login.html
在网页中添加用户名和密码输入框以及按钮。
点击按钮是异步的向后端发送请求, 完成登录 (post 方法)

2.用户列表

在 myPro.js 定义一个接口 userlist
查询所有的用户数据, 并响应给前端
在 myPro 文件夹中创建网页 02_list.html
在 body 的 onload 事件中绑定方法 get_list()
16:50 <tbody id="content">

登录名称	电子邮箱	联系方式	用户姓名	用户性别	操作
dingding	ding@qq.com	13511011000	丁春秋	女	删除 修改
dangdang	dang@qq.com	13501234568	当当啵	男	删除 修改
doudou	dou@qq.com	13501234569	姜志强	男	删除 修改
yaya	ya@qq.com	13501234560	秦小雅	女	删除 修改
1111	441977193@qq.com	18357100796	null	不详	删除 修改
ABCD	123@qq.com	13538894495	null	不详	删除 修改
mohk	11@qq.com	13512312312	null	不详	删除 修改
121123	491000888@qq.com	13213389258	null	不详	删除 修改
555555	55555555@163.com	13400000000	null	不详	删除 修改
xuyong	123456789@qq.com	15525623622	null	不详	删除 修改
admin	mail@xingyu1993.cn	13580510164	null	不详	删除 修改
siyongbo	616188545@qq.com	18447103998	null	不详	删除 修改
qwerty	1091256014@qq.com	15617152367	null	不详	删除 修改
dingziqiang	996534706@qq.com	15567502520	null	不详	删除 修改
hdb2264068	471062503@qq.com	18898405759	null	不详	删除 修改
wenhua	liwenhua@tedu.cn	15012345678	null	不详	删除 修改

作业:

uid

删除

注册

ajax 错误总结

404: 检查 url 是否正确, 要访问的资源是不是在 url 指向的位置(html 位置, 和路由)
5XX: 检查接口中的代码是否正确
show is not defined show()方法未定义
Uncaught SyntaxError: Unexpected token { 标点符号, 或者括号错误
Cross origin requests are only supported for protocol schemes: http, data, chrome, chrome-extension, https.跨域错误, 说明你的 html 是本地打开
Failed to execute 'setRequestHeader' on 'XMLHttpRequest': The object's state must be OPENED 修改请求头放在 xhr.open 之前或者 xhr.send 之后