# 2018.03.28

## Ajax

asynchronous javascript and xml异步js和xml

xml: extensible markup language 可扩展标记语言，标准通用标记语言的子集，是一种用于标记电子文件技术使其具有结构性的标记语言。

## Ajax作用

在浏览器与服务器之间进行少量数据交换，是可以使网页实现异步更新。不重新加载整个网页的情况下，对网页的某部分进行更新。

优点：

减轻服务器负担，按需要获得数据

无需刷新页面，减少用户的等待时间

更好的用户体验

减轻带宽的负担

主流浏览器支持

缺点：

Ajax大量使用了JavaScript和ajax引擎，而这个取决于浏览器的支持。所以使用Ajax程序必须测试针对各个浏览器的兼容性。

Ajax更新页面没有刷新整个页面，因此网页的后退功能失效。最好在醒目的位置提醒用户‘数据已更新’。

对搜索引擎支持不好。

## Ajax应用

用户注册时验证用户名是否存在

新闻或者产品评论功能，评论后立即显示

分页时如果不想刷新整个页面，可是实现分页部分页面的更新

## Ajax引擎

Ajax引擎就是XMLHttpRequest对象,所有现代浏览器均支持 XMLHttpRequest 对象（IE5 和 IE6 使用 ActiveXObject）。它同时也是一个Javascript对象.

createXHR.js（解决兼容性。在body下面script中请求）

通常是Javascript监听浏览器网页事件(点击,提交,更改等),由Javascript创建Ajax引擎对象,通过Ajax引擎对象发出请求.Ajax引擎等待并且接收服务器的响应内容,Javascript再从Ajax引擎对象中获取响应内容并且改变网页界面显示效果.

## 使用Ajax

### 创建对象

var xhr = new XMLHttpRequest(); //W3C标准

var xhr = createXHR(); //解决兼容性问题

### 根据实际需要确定get post发送请求open!!

xhr.open(‘GET’, url,true)

para1:get \ post para2:url para3: asynchronous(default---true)

### 监听Ajax引擎对象的状态改变的事件

xhr.onreadyStatechang = function(){

//根据状态处理返回的结果

}

### 发送请求到http服务器并接受响应

xhr.send();

## ajax引擎对象常见的方法和属性

### 方法

abort 取消当前请求

getAllResponseHeaders 获取响应的所有http头

getResponseHeader 从响应信息中获取指定的http头

open(‘GET’, url,true) 发起http请求，第三个参数是默认true

send() 发送http请求到服务器并接收响应

setRequestHeader() 单独指定请求的http头协议信息。post方式才会需要设置。

### 属性

onreadstatechange 监听状态改变时的事件处理句柄。（事件注册）

xhr.onreadystatechange=function(){....}

readState 返回请求的状态。

1. 请求未初始化
2. 服务器连接已建立
3. 请求已接收
4. 请求处理中
5. 请求已完成，且响应就绪

status 返回当前http状态码，只读

1. OK
2. 缓存，未修改

403 没有权限

1. not found

501 服务器级别出错

responseText 返回响应信息--以字符串的形式（只读）

## 同步和异步

同步：发出功能调用时，在没有得到结果之前，不能够继续调用其他功能。

异步：异步过程调用发出后，调用者不能立刻得到结果，程序继续向下执行。实际处理这个调用的部件在完成后，通过状态，通知和回调通知调用者

同步是阻塞模式，异步是非阻塞模式。

Ajax中的异步请求是异步的真正实现，用户页面（HTML）没有直接和服务器打交道而是通过Ajax引擎间接和服务器打交道，用户页面和Ajax引擎打交道时并不会等待服务器返回的响应内容，而压面中JavaScript继续用执行，也可以继续发出新的Ajax请求。

## JSON

JSON（JavaScript Object Notation ）JS对象标记，是一种轻量级的数据存储和交换格式。JSON天生就是JavaScript对象。

### json优点

JSON比XML更小，更快，更易解析

### json对象语法

{“keyname”:”value”}

### json重要方法

JSON.parse(); 字符串转JSON对象

JSON.stringfy(); JSON对象转字符串

# 2018.03.29

## 回顾原生Ajax

### 创建对象

var xhr = new XMLHttpRequest(); //W3C标准

var xhr = createXHR(); //解决兼容性问题

### 根据实际需要确定get post发送请求open!!

xhr.open(‘GET’, url,true)

para1:get \ post para2:url para3: asynchronous(default---true)

### 监听Ajax引擎对象的状态改变的事件

xhr.onreadyStatechang = function(){

//根据状态处理返回的结果

}

### 发送请求到http服务器并接受响应

xhr.send();

## JQuery Ajax

### get()

$(sceletor).get(url,[data],[callback],[type])

url: 待载入页面的url地址

data: 待发送key/value参数。json结构的数据

callback: 载入成功的回调函数，处理返回的结构。回调函数传入的参数就是res返回的结果

type：返回内容格式，默认json,也可以是XML，HTML，script，text，\_default

### post()

$(sceletor).post(url,[data],[callback],[type])

url: 待载入页面的url地址

data: 待发送key/value参数。json结构的数据

callback: 载入成功的回调函数，处理返回的结构。回调函数传入的参数就是res返回的结果

type：返回内容格式，默认json,也可以是XML，HTML，script，text，\_default

### ajax()

$ajax(setting);

setting:语法

{

type: “post”, //可以是post，可以是get

url: ”some.php”, //请求的路由或远程url地址

data: ”name=John&location=Boston”, //传送数据，可以由serialize()方法

dataType: “jsonp”, //返回的数据类型---jsonp是一种跨域方法

success: function(msg){ //成功的回调函数

.....

}

}

## 同域与跨域

同域：协议相同http / https，域名相同，端口相同（同时满足）

跨域：三者中有一个不同，就是跨域

解决方案：

1. jsonp

2.在自己的服务器里添加加应头：res.header(“Access-Control-Allow-Origin”,”\*”)

## 模板入门

模板是web前端模板，将HTML网页划分为前端的固定结构与后端提供数据进行绑定的过程。

why前端模板引擎？

因为HTML结构和数据要拼接成一个动态网页，如果没有模板，最终的HTML的拼接就要我们自己做字符串拼接，模板引擎帮我们自动完成。

### 模板引擎有：

ejs express框架首选模板

artTemplate 效率最高

jQueryTemplate 操作最简单

### 语法对比

#### 绑定数据：

ejs <%=value%>

artTemplate {{content}} 表达式

jQueryTemplate {{=value}} 或者${value}

#### 条件语句

ejs

<% if (condition) { %>

content

<%}%>

artTemplate

{{if condition}}

content

{{/if}}

jqueryTemplate

{{if condition}}

content

{{/if}}

### 循环语句

ejs

<% for(var i = 0; i<arr.length; i++) { %>

content

<% } %>

artTemplate

{{each lists as value index}}

content

{{/each}}

参数说明：lists是数组，value是数组的元素，index是元素的索引

jQueryTemplate

{{each(i,v) lists}}

content

{{/each}}

参数说明：lists是数组，v是数组的元素，i是元素的索引

artTemplates模板引擎使用步骤

1. 下载并解压artTemplate的库文件
2. 引入库文件
3. 定义模板结构（注意 script 的type属性必须是”text/html”）
4. 发起Ajax请求获取数据（使用简单的jQuery）
5. 使用模板引擎渲染前端模板template（‘模板名’，json格式的数据）
6. 使用DOM操作把渲染后的结果更新到页面