李南江亲授-jQuery+Ajax从放弃到知根知底

<https://www.bilibili.com/video/av22807707?p=1>

# 五个步骤 / 基本使用

button.onclick = function (event) {

// 1.创建一个异步对象

var xmlhttp = new XMLHttpRequest();

// 2.设置请求方式和请求地址

/\*

第一个参数method: 请求的类型: GET或POST

第二个参数url: 文件在服务器上的位置

第三个参数async: true(异步) 或 false (同步)，一般都设置为true

\*/

xmlhttp.open(GET, ”04-ajax-get.php”, true);

// 3.发送请求

xmlhttp.send();

// 4.监听状态的变化

xmlhttp.onreadystatechange = function (ev2) {

/\*

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

\*/

if (xmlhttp.readyState === 4) {

/\*判断是否请求成功，根据服务器返回的状态码，一般来说，大于200小于300和304都是成功的状态码\*/

if (xmlhttp.status >= 200 && xmlhttp.status < 300 || xmlhttp. status === 304) {

// 5.处理返回的结果

console.log(xmlhttp.responseText); // 成功的处理结果

// responseText获得字符串形式的响应数据

// responseXML获得XML形式的响应数据

}else{

console.log(“没有接收到服务器返回的数据”); // 失败的处理结果

}

}

# IE兼容

1. 所有现代浏览器均支持XMLHttpRequest对象(IE5和IE6使用ActiveXObject)。

var xmlhttp;

if (window.XMLHttpRequest)

{

// code for IE7+, Firefox, Chrome, opera, Safari

xmlhttp=new XMLHttpRequest();

}

else

{

// code for IE6, IE5

xmlhttp=new ActiveXOjject("Microsoft.XMLHTTP");

}

1. Ajax在IE中的缓存问题

在IE浏览器中，通过Ajax发送GET请求，IE浏览器认为同一个URL只有一个结果。即，IE不能拿到实时变化的数据，如果已经请求过一个URL，则该地址的返回的内容发生变化后，IE得到的结果不会发生变化。

解决方法：使每一次请求的URL都不一样，例如在URL后添加时间戳。

xmlhttp.open(”GET", 05-ajax-get.txt+ ”?t=" + (new Date().getTime()), true);

# 封装Ajax

## GET请求

function obj2str (obj) { // 处理参数

/\*

userName":"1nj",

”userPwd" :”123456"

\*/

obj.t = new Date().getTime(); // 增加时间戳，兼容IE

var res = [];

for(var key in obj) { // URL中不能出现中文，用encodeURIComponent进行转换

res.push (encodeURIComponent(key)+ ”=" + encodeURIComponent(obj[key]));

// [userName=lnj, userPwd=123456] ;

}

return res.join(“&”) ; // userName= ln j&userPwd= 123456

function ajax(url, obj, timeout, success, error) {

// url为别人传入的服务器地址

// obj为传入的对象

// timeout为超时时间

// success、error为两个回调函数，用于处理请求成功/失败

// 0.将对象转换为字符串

var str = obj2str (obj); // 将对象转换为指定格式：key=value&key=value;

// 1.创建一个异步对象

var xmlhttp;

if (window.XMLHttpRequest) // 兼容IE

{

// code for IE7+, Firefox, Chrome, opera, Safari

xmlhttp=new XMLHttpRequest();

}

else

{

// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

// 2.设置请求方式和请求地址

/\*

第一个参数method: 请求的类型: GET或POST

第二个参数url: 文件在服务器上的位置

第三个参数async: true(异步) 或 false (同步)，一般都设置为true

\*/

xmlhttp.open(GET, url+ ”?t=" + str, true);

// 3.发送请求

xmlhttp.send();

// 4.监听状态的变化

xmlhttp.onreadystatechange = function (ev2) {

/\*

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

\*/

if (xmlhttp.readyState === 4) {

clearInterval(timer); // 成功则清除超时时间定时器

/\*判断是否请求成功，根据服务器返回的状态码，一般来说，大于200小于300和304都是成功的状态码\*/

if (xmlhttp.status >= 200 && xmlhttp.status < 300 || xmlhttp. status === 304) {

// 5.处理返回的结果

success(xmlhttp); // 成功的处理结果

}else{

error(xmlhttp); // 失败的处理结果

}

// 判断外界是否传入了超时时间

if (timeout) {

timer = set Interval (function () {

xmlhttp.abort(); // 中断请求

clearInterval(timer); // 清除定时器

}, timeout);

}

使用：

var oBtn = document. querySe lector(" button' ) ;

oBtn.onclick = function (ev) {

ajax("04-ajax-get2.php", {

“userName”: “lnj”,

“userPwd”: “123456”

}, 3000

, function (xhr) {

alert (xhr. responseText) ;

}, function (xhr) {

alert("请求失败”);

});

}

## POST请求

button.onclick = function (event) {

// 1.创建一个异步对象

var xmlhttp = new XMLHttpRequest();

// 2.设置请求方式和请求地址

/\*

第一个参数method: 请求的类型: GET或POST

第二个参数url: 文件在服务器上的位置

第三个参数async: true(异步) 或 false (同步)，一般都设置为true

\*/

xmlhttp.open(POST, ”04-ajax-get.php”, true);

//注意点:以下代码必须放到open和send之间

xmlhttp.setRequestHeader(“Content-type", “application/x-wwwform-urlencoded”);

// 3.发送请求

xmlhttp.send(“userName=zs&userPwd=321”); // 在send中传参

// 4.监听状态的变化

xmlhttp.onreadystatechange = function (ev2) {

/\*

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

\*/

if (xmlhttp.readyState === 4) {

/\*判断是否请求成功，根据服务器返回的状态码，一般来说，大于200小于300和304都是成功的状态码\*/

if (xmlhttp.status >= 200 && xmlhttp.status < 300 || xmlhttp. status === 304) {

// 5.处理返回的结果

console.log(xmlhttp.responseText); // 成功的处理结果

// responseText获得字符串形式的响应数据

// responseXML获得XML形式的响应数据

}else{

console.log(“没有接收到服务器返回的数据”); // 失败的处理结果

}

}

# 真正的封装

function obj2str (data) { // 处理参数

/\*

userName":"1nj",

”userPwd" :”123456"

\*/

data.t = new Date().getTime(); // 增加时间戳，兼容IE

var res = [];

for(var key in data) { // URL中不能出现中文，用encodeURIComponent进行转换

res.push (encodeURIComponent(key)+ ”=" + encodeURIComponent(data[key]));

// [userName=lnj, userPwd=123456] ;

}

return res.join(“&”) ; // userName= ln j&userPwd= 123456

function ajax(option) { // 传入对象，这样与形参顺序无关

// type为请求类型

// url为别人传入的服务器地址

// obj为传入的对象

// timeout为超时时间

// success、error为两个回调函数，用于处理请求成功/失败

// 0.将对象转换为字符串

var str = obj2str (option.data); // 将对象转换为指定格式：key=value&key=value;

// 1.创建一个异步对象

var xmlhttp;

if (window.XMLHttpRequest) // 兼容IE

{

// code for IE7+, Firefox, Chrome, opera, Safari

xmlhttp=new XMLHttpRequest();

}

else

{

// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

// 2.设置请求方式和请求地址

/\*

第一个参数method: 请求的类型: GET或POST

第二个参数url: 文件在服务器上的位置

第三个参数async: true(异步) 或 false (同步)，一般都设置为true

\*/

// 判断请求类型

if(option.type.toUpperCase() === “GET”) { // GET请求，同时转换为大写，因此外面可以是小写

xmlhttp.open(option.type, option.url+ ”?t=" + str, true);

// 3.发送请求

xmlhttp.send();

}else{ // POST请求

xmlhttp.open(option.type, option.url, true);

//注意点:以下代码必须放到open和send之间

xmlhttp.setRequestHeader(“Content-type", “application/x-wwwform-urlencoded”);

// 3.发送请求

xmlhttp.send(str); // 在send中传参

}

// 4.监听状态的变化

xmlhttp.onreadystatechange = function (ev2) {

/\*

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

\*/

if (xmlhttp.readyState === 4) {

clearInterval(timer); // 成功则清除超时时间定时器

/\*判断是否请求成功，根据服务器返回的状态码，一般来说，大于200小于300和304都是成功的状态码\*/

if (xmlhttp.status >= 200 && xmlhttp.status < 300 || xmlhttp. status === 304) {

// 5.处理返回的结果

option.success(xmlhttp); // 成功的处理结果

}else{

option.error(xmlhttp); // 失败的处理结果

}

// 判断外界是否传入了超时时间

if (option.timeout) {

timer = set Interval (function () {

xmlhttp.abort(); // 中断请求

clearInterval(timer); // 清除定时器

}, option.timeout);

}

使用：

var oBtn = document.querySelector("button”);

oBtn.onclick = function (ev) {

ajax({

url:"04-ajax-get2.php",

data:{

“userName”: “lnj”,

“userPwd”: “123456”

},

type:”GET”,

timeout:3000,

success:function (xhr) {

alert (xhr. responseText);

},

error:function (xhr) {

alert("请求失败”);

}

});

}