# ajax()方法

ajax() 方法通过 HTTP 请求加载远程数据，该方法是 jQuery 底层 AJAX 实现。简单易用的高层实现见 $.get, $.post 等。$.ajax() 返回其创建的 XMLHttpRequest 对象。大多数情况下你无需直接操作该函数，除非你需要操作不常用的选项，以获得更多的灵活性。最简单的情况下，$.ajax() 可以不带任何参数直接使用。

## 语法

jQuery.ajax([settings])

参数：

settings：可选。用于配置 Ajax 请求的键值对集合。

## 参数

### options

类型：Object

可选。AJAX 请求设置。所有选项都是可选的。

### url

类型：String

默认值: 当前页地址。发送请求的地址（可以是接口地址）

### method

类型：String

可选值：GET POST PUT DELETE

### contentType

类型：String

默认值: "application/x-www-form-urlencoded"。发送信息至服务器时内容编码类型。默认值适合大多数情况。如果你明确地传递了一个 content-type 给 $.ajax() 那么它必定会发送给服务器（即使没有数据要发送）。

### data

类型：String

发送到服务器的数据。将自动转换为请求字符串格式。GET 请求中将附加在 URL 后。查看 processData 选项说明以禁止此自动转换。必须为 Key/Value 格式。如果为数组，jQuery 将自动为不同值对应同一个名称。如 {foo:["bar1", "bar2"]} 转换为 '&foo=bar1&foo=bar2'。

### dataType

类型：String

预期服务器返回的数据类型。如果不指定，jQuery 将自动根据 HTTP 包 MIME 信息来智能判断, 比如 XML MIME 类型就被识别为 XML。在 1.4 中，JSON 就会生成一个 JavaScript 对象，而 script 则会执行这个脚本。随后服务器端返回的数据会根据这个值解析后，传递给回调函数。可用值:

**"xml"**: 返回 XML 文档，可用 jQuery 处理。

**"html"**: 返回纯文本 HTML 信息；包含的 script 标签会在插入 dom 时执行。

**"script"**: 返回纯文本 JavaScript 代码。不会自动缓存结果。除非设置了 "cache" 参数。注意：在远程请求时(不在同一个域下)，所有 POST 请求都将转为 GET 请求。（因为将使用 DOM 的 script标签来加载）

**"json"**: 返回 JSON 数据 。

**"jsonp"**: JSONP 格式。使用 JSONP 形式调用函数时，如 "myurl?callback=?" jQuery 将自动替换 ? 为正确的函数名，以执行回调函数。

**"text"**: 返回纯文本字符串

### cache

类型：Boolean

默认值: true，dataType 为 script 和 jsonp 时默认为 false。设置为 false 将不缓存此页面。

jQuery 1.2 新功能。

### type

类型：String

默认值: "GET"。请求方式 ("POST" 或 "GET")， 默认为 "GET"。注意：其它 HTTP 请求方法，如 PUT 和 DELETE 也可以使用，但仅部分浏览器支持。

### username

类型：String

用于响应 HTTP 访问认证请求的用户名

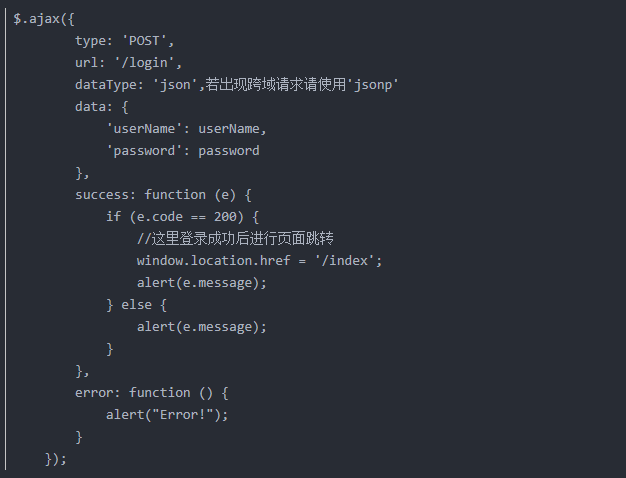
### xhr

类型：Function

需要返回一个 XMLHttpRequest 对象。默认在 IE 下是 ActiveXObject 而其他情况下是 XMLHttpRequest 。用于重写或者提供一个增强的 XMLHttpRequest 对象。这个参数在 jQuery 1.3 以前不可用。

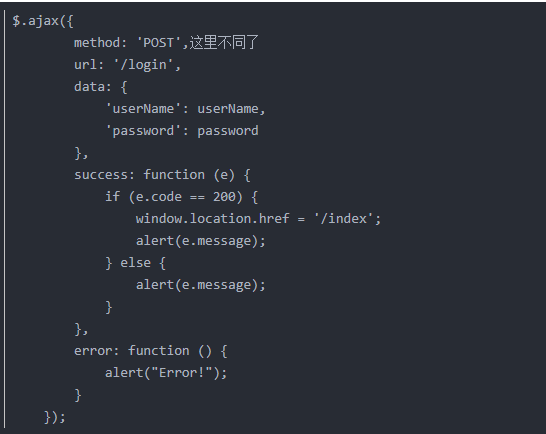
# jQuery中ajax配置项中的使用type与method的区别

本质上两个配置项是没有区别的，区别在于两者出现的时间不同，type对于目前jQuery的版本全部兼容，也就是说



这种方式书写，type类型是GET就是GET请求，是POST请求就是POST请求。

在jQuery1.9以后的版本中使用ajax时，可以使用如下方式：



实现的功能是一样的，但是如果此时你引入的jQuery的js文件版本太旧，就会出现请求全是以默认方式GET进行的，所以会出现一些无厘头的bug。

# JSON.stringify()

JSON.stringify() 方法用于将 JavaScript 值转换为 JSON 字符串。

## 语法

JSON.stringify(value[, replacer[, space]])

## 参数说明

### value

必需， 要转换的 JavaScript 值（通常为对象或数组）

### replacer

可选。用于转换结果的函数或数组。

### space

可选，文本添加缩进、空格和换行符，如果 space 是一个数字，则返回值文本在每个级别缩进指定数目的空格，如果 space 大于 10，则文本缩进 10 个空格。space 也可以使用非数字，如：\t。

## 返回值

返回包含 JSON 文本的字符串

## 示例



# Ajax调用接口示例

<script>

$('#btn522925407').on('click',function(){

var myDate = new Date();

var exportjson = {

"country": "sg",

"language": "en",

"list": {

"date":myDate,

"filename":"registerinfo"

}

}

$.ajax({

url: 'http://localhost:8080/mypage/exportGet',

method: 'GET',

contentType: "application/x-www-form-urlencoded",

data: JSON.stringify(exportjson)

}).done(function(data){

alert(data.error\_message);

}).fail(function(data){

alert("export fail,error in background interface");

});

})

</script>