第 12 章 Ajax

学习要点：

1.Ajax 概述
2.load()方法
3.$.get()和$.post()
4.$.getScript()和$.getJSON()
5.$.ajax()方法
6.表单序列化

主讲教师：李炎恢
官方网站：http://www.ycku.com
合作网站：http://www.ibeifeng.com

Ajax 全称为："Asynchronous JavaScript and XML"（异步 JavaScript 和 XML），它并不是 JavaScript 的一种单一技术，而是利用了一系列交互式网页应用相关的技术所形成的结合体。使用 Ajax，我们可以无刷新状态更新页面，并且实现异步提交，提升了用户体验。

一．Ajax 概述

Ajax 这个概念是由 Jesse James Garrett 在 2005 年发明的。它本身不是单一技术，是一串技术的集合，主要有：

1.JavaScript，通过用户或其他与浏览器相关事件捕获交互行为；
2.XMLHttpRequest 对象，通过这个对象可以在不中断其它浏览器任务的情况下向服务器发送请求；
3.服务器上的文件，以 XML、HTML 或 JSON 格式保存文本数据；
4.其它 JavaScript，解释来自服务器的数据（比如 PHP 从 MySQL 获取的数据）并将其呈现到页面上。

由于 Ajax 包含众多特性，优势与不足也非常明显。优势主要以下几点：
1.不需要插件支持（一般浏览器且默认开启 JavaScript 即可）；
2.用户体验极佳（不刷新页面即可获取可更新的数据）；
3.提升 Web 程序的性能（在传递数据方面做到按需放松，不必整体提交）；
4.减轻服务器和带宽的负担（将服务器的一些操作转移到客户端）；

而 Ajax 的不足由以下几点：
1.不同版本的浏览器度 XMLHttpRequest 对象支持度不足(比如 IE5 之前)；
2.前进、后退的功能被破坏（因为 Ajax 永远在当前页，不会几率前后页面）；
3.搜索引擎的支持度不够（因为搜索引擎爬虫还不能理解 JS 引起变化数据的内容）；
4.开发调试工具缺乏（相对于其他语言的工具集来说，JS 或 Ajax 调试开发少的可怜）。

异步和同步

使用 Ajax 最关键的地方，就是实现异步请求、接受响应及执行回调。那么异步与同步有什么区别呢？我们普通的 Web 程序开发基本都是同步的，意为执行一段程序才能执行下一段，类似电话中的通话，一个电话接完才能接听下个电话；而异步可以同时执行多条任务，感觉有多条线路，类似于短信，不会因为看一条短信而停止接受另一条短信。Ajax 也可以使用同步模式执行，但同步的模式属于阻塞模式，这样会导致多条线路执行时又必须一条一条执行，会让 Web 页面出现假死状态，所以，一般 Ajax 大部分采用异步模式。

二．load()方法

jQuery 对 Ajax 做了大量的封装，我们使用起来也较为方便，不需要去考虑浏览器兼容性。对于封装的方式，jQuery 采用了三层封装：最底层的封装方法为：$.ajax()，而通过这层封装了第二层有三种方法：.load()、$.get()和$.post()，最高层是$.getScript()和$.getJSON()方法。

.load()方法可以参数三个参数：url(必须，请求 html 文件的 url 地址，参数类型为 String)、data(可选，发送的 key/value 数据，参数类型为 Object)、callback(可选，成功或失败的回调函数，参数类型为函数 Function)。

如果想让 Ajax 异步载入一段 HTML 内容，我们只需要一个 HTML 请求的 url 即可。

```
//HTML
<input type="button" value="异步获取数据" />
<div id="box"></div>

//jQuery
$('input').click(function () {
    $('#box').load('test.html');
});
```

如果想对载入的 HTML 进行筛选，那么只要在 url 参数后面跟着一个选择器即可。

```
//带选择器的 url
$('input').click(function () {
    $('#box').load('test.html .my');
});
```

如果是服务器文件，比如.php。一般不仅需要载入数据，还需要向服务器提交数据，那么我们就可以使用第二个可选参数 data。向服务器提交数据有两种方式：get 和 post。

```
//不传递 data，则默认 get 方式
$('input').click(function () {
    $('#box').load('test.php?url=ycku');
});

//get 方式接受的 PHP
<?php
    if ($_GET['url'] == 'ycku') {
        echo '瓢城 Web 俱乐部官网';
    } else {
        echo '其他网站';
```

```
    }
?>

//传递 data，则为 post 方式
$('input').click(function () {
    $('#box').load('test.php', {
        url : 'ycku'
    });
});

//post 方式接受的 PHP
<?php
    if ($_POST['url'] == 'ycku') {
        echo '瓢城 Web 俱乐部官网';
    } else {
        echo '其他网站';
    }
?>
```

在 Ajax 数据载入完毕之后，就能执行回调函数 callback，也就是第三个参数。回调函数也可以传递三个可选参数：responseText（请求返回）、textStatus（请求状态）、XMLHttpRequest（XMLHttpRequest 对象）。

```
$('input').click(function () {
    $('#box').load('test.php', {
        url : 'ycku'
    }, function (response, status, xhr) {
        alert('返回的值为：' + response + '，状态为：' + status + '，
                                                状态是：' + xhr.statusText);
    });
});
```

注意：status 得到的值，如果成功返回数据则为：success，否则为：error。XMLHttpRequest 对象属于 JavaScript 范畴，可以调用一些属性如下：

属性名	说明
responseText	作为响应主体被返回的文本
responseXML	如果响应主体内容类型是"text/xml"或"application/xml"，则返回包含响应数据的 XML DOM 文档
status	响应的 HTTP 状态
statusText	HTTP 状态的说明

如果成功返回数据，那么 xhr 对象的 statusText 属性则返回'OK'字符串。除了'OK'的状态字符串，statusText 属性还提供了一系列其他的值，如下：

HTTP 状态码	状态字符串	说明
200	OK	服务器成功返回了页面
400	Bad Request	语法错误导致服务器不识别
401	Unauthorized	请求需要用户认证
404	Not found	指定的 URL 在服务器上找不到
500	Internal Server Error	服务器遇到意外错误，无法完成请求
503	ServiceUnavailable	由于服务器过载或维护导致无法完成请求

三．$.get()和$.post()

.load()方法是局部方法，因为他需要一个包含元素的 jQuery 对象作为前缀。而$.get()和$.post()是全局方法，无须指定某个元素。对于用途而言，.load()适合做静态文件的异步获取，而对于需要传递参数到服务器页面的，$.get()和$.post()更加合适。

$.get()方法有四个参数，前面三个参数和.load()一样，多了一个第四参数 type，即服务器返回的内容格式：包括 xml、html、script、json、jsonp 和 text。第一个参数为必选参数，后面三个为可选参数。

```
//使用$.get()异步返回 html 类型
$('input').click(function () {
    $.get('test.php', {
        url : 'ycku'
    }, function (response, status, xhr) {
        if (status == 'success') {
            $('#box').html(response);
        }
    })                                          //type 自动转为 html
});
```

注意：第四参数 type 是指定异步返回的类型。一般情况下 type 参数是智能判断，并不需要我们主动设置，如果主动设置，则会强行按照指定类型格式返回。

```
//使用$.get()异步返回 xml
$('input').click(function () {
    $.get('test.xml', function (response, status, xhr) {
        $('#box').html($(response).find('root').find('url').text());
    });                                         //type 自动转为 xml
});
```

注意：如果载入的是 xml 文件，type 会智能判断。如果强行设置 html 类型返回，则会把 xml 文件当成普通数据全部返回，而不会按照 xml 格式解析数据。

```
//使用$.get()异步返回 json
$.get('test.json', function (response, status, xhr) {
```

```
        alert(response[0].url);
    });
```

$.post()方法的使用和$.get()基本上一致，他们之间的区别也比较隐晦，基本都是背后的不同，在用户使用上体现不出。具体区别如下：

1.GET 请求是通过 URL 提交的，而 POST 请求则是 HTTP 消息实体提交的；

2.GET 提交有大小限制（2KB），而 POST 方式不受限制；

3.GET 方式会被缓存下来，可能有安全性问题，而 POST 没有这个问题；

4.GET 方式通过$_GET[]获取，POST 方式通过$_POST[]获取。

```
    //使用$.post()异步返回 html
    $.post('test.php', {
        url : 'ycku'
    }, function (response, status, xhr) {
        $('#box').html(response);
    });
```

四．$.getScript()和$.getJSON()

jQuery 提供了一组用于特定异步加载的方法：$.getScript()，用于加载特定的 JS 文件；$.getJSON()，用于专门加载 JSON 文件。

有时我们希望能够特定的情况再加载 JS 文件，而不是一开始把所有 JS 文件都加载了，这时课时使用$.getScript()方法。

```
    //点击按钮后再加载 JS 文件
    $('input').click(function () {
        $.getScript('test.js');
    });
```

$.getJSON()方法是专门用于加载 JSON 文件的，使用方法和之前的类似。

```
    $('input').click(function () {
        $.getJSON('test.json', function (response, status, xhr) {
            alert(response[0].url);
        });
    });
```

五．$.ajax()

$.ajax()是所有 ajax 方法中最底层的方法，所有其他方法都是基于$.ajax()方法的封装。这个方法只有一个参数，传递一个各个功能键值对的对象。

$.ajax()方法对象参数表

参数	类型	说明
url	String	发送请求的地址
type	String	请求方式：POST 或 GET，默认 GET

timeout	Number	设置请求超时的时间（毫秒）
data	Object 或 String	发送到服务器的数据，键值对字符串或对象
dataType	String	返回的数据类型，比如 html、xml、json 等
beforeSend	Function	发送请求前可修改 XMLHttpRequest 对象的函数
complete	Function	请求完成后调用的回调函数
success	Function	请求成功后调用的回调函数
error	Function	请求失败时调用的回调函数
global	Boolean	默认为 true，表示是否触发全局 Ajax
cache	Boolean	设置浏览器缓存响应，默认为 true。如果 dataType 类型为 script 或 jsonp 则为 false。
content	DOM	指定某个元素为与这个请求相关的所有回调函数的上下文。
contentType	String	指定请求内容的类型。默认为 application/x-www-form-urlencoded。
async	Boolean	是否异步处理。默认为 true，false 为同步处理
processData	Boolean	默认为 true，数据被处理为 URL 编码格式。如果为 false，则阻止将传入的数据处理为 URL 编码的格式。
dataFilter	Function	用来筛选响应数据的回调函数。
ifModified	Boolean	默认为 false，不进行头检测。如果为 true，进行头检测，当相应内容与上次请求改变时，请求被认为是成功的。
jsonp	String	指定一个查询参数名称来覆盖默认的 jsonp 回调参数名 callback。
username	String	在 HTTP 认证请求中使用的用户名
password	String	在 HTTP 认证请求中使用的密码
scriptCharset	String	当远程和本地内容使用不同的字符集时，用来设置 script 和 jsonp 请求所使用的字符集。
xhr	Function	用来提供 XHR 实例自定义实现的回调函数
traditional	Boolean	默认为 false，不使用传统风格的参数序列化。如为 true，则使用。

```
//$.ajax 使用
$('input').click(function () {
    $.ajax({
        type : 'POST',                          //这里可以换成 GET
```

```
            url : 'test.php',
            data : {
                url : 'ycku'
            },
            success : function (response, stutas, xhr) {
                $('#box').html(response);
            }
        });
    });
```

注意：对于 data 属性，如果是 GET 模式，可以使用三种之前说所的三种形式。如果是 POST 模式可以使用之前的两种形式。

六．表单序列化

Ajax 用的最多的地方莫过于表单操作，而传统的表单操作是通过 submit 提交将数据传输到服务器端。如果使用 Ajax 异步处理的话，我们需要将每个表单元素逐个获取才方能提交。这样工作效率就大大降低。

```
    //常规形式的表单提交
    $('form input[type=button]').click(function () {
        $.ajax({
            type : 'POST',
            url : 'test.php',
            data : {
                user : $('form input[name=user]').val(),
                email : $('form input[name=email]').val()
            },
            success : function (response, status, xhr) {
                alert(response);
            }
        });
    });
```

使用表单序列化方法.serialize()，会智能的获取指定表单内的所有元素。这样，在面对大量表单元素时，会把表单元素内容序列化为字符串，然后再使用 Ajax 请求。

```
    //使用.serialize()序列化表单内容
    $('form input[type=button]').click(function () {
        $.ajax({
            type : 'POST',
            url : 'test.php',
            data : $('form').serialize(),
            success : function (response, status, xhr) {
                alert(response);
            }
        });
```

```
});
```

.serialize()方法不但可以序列化表单内的元素，还可以直接获取单选框、复选框和下拉列表框等内容。

```
//使用序列化得到选中的元素内容
$(':radio').click(function () {
    $('#box').html(decodeURIComponent($(this).serialize()));
});
```

除了.serialize()方法，还有一个可以返回 JSON 数据的方法：.serializeArray()。这个方法可以直接把数据整合成键值对的 JSON 对象。

```
$(':radio').click(function () {
    console.log($(this).serializeArray());
    var json = $(this).serializeArray();
    $('#box').html(json[0].value);
});
```

有时，我们可能会在同一个程序中多次调用$.ajax()方法。而它们很多参数都相同，这个时候我们课时使用 jQuery 提供的$.ajaxSetup()请求默认值来初始化参数。

```
$('form input[type=button]').click(function () {
    $.ajaxSetup({
        type : 'POST',
        url : 'test.php',
        data : $('form').serialize()
    });
    $.ajax({
        success : function (response, status, xhr) {
            alert(response);
        }
    });
});
```

在使用 data 属性传递的时候，如果是以对象形式传递键值对，可以使用$.param()方法将对象转换为字符串键值对格式。

```
var obj = {a : 1, b : 2, c : 3};
var form = $.param(obj);
alert(form);
```

注意：使用$.param()将对象形式的键值对转为 URL 地址的字符串键值对，可以更加稳定准确的传递表单内容。因为有时程序对于复杂的序列化解析能力有限，所以直接传递 obj 对象要谨慎。

北风网 IBEIFENG.COM

Ycku.com 瓢城Web俱乐部，做最酷的教学

感谢收看本次教程！

本课程是由北风网(ibeifeng.com)

瓢城 Web 俱乐部(ycku.com)联合提供：

本次主讲老师：李炎恢

谢谢大家，再见！