

jQuery 之 form 插件

jQuery 表单插件 jQuery.form

这是一个全面支持表单的 jQuery 插件，支持文件上传，包含以下一些方法：

- ajaxForm
- ajaxSubmit
- formToArray
- formSerialize
- fieldSerialize
- fieldValue
- clearForm
- clearFields
- resetForm

使用时你必须依次引入 jquery.js、jquery.form.js，并且顺序不能颠倒

示例代码：

```
// wait for the DOM to be loaded
$(document).ready(function() {
    // bind 'myForm' and provide a simple callback function
    $('#myForm').ajaxForm(function() {
        alert("Thank you for your comment!");
    });
});
```

项目主页：<http://malsup.com/jquery/form/>

下载地址：<http://malsup.com/jquery/form/jquery.form.js>

表单插件 API

英文原文：<http://www.malsup.com/jquery/form/#api>

表单插件 API 提供了几个方法，让你轻松管理表单数据和进行表单提交。

ajaxForm ()

增加所有需要的事件监听器，为 AJAX 提交表单做好准备。ajaxForm 不能提交表单。在 document 的 ready 函数中，使用 ajaxForm 来为 AJAX 提交表单进行准备。ajaxForm 接受 0 个或 1 个参数。这个单个的参数既可以是一个回调函数，也可以是一个 Options 对象。

可链接 (Chainable)：可以。

实例：

```
$('#myFormId').ajaxForm();
```

ajaxSubmit ()

马上由 AJAX 来提交表单。大多数情况下，都是调用 ajaxSubmit 来对用户提交表单进行响应。ajaxSubmit 接受 0 个或 1 个参数。这个单个的参数既可以是一个回调函数，也可以是一个 Options 对象。

可链接 (Chainable)：可以。

实例：

```
// 绑定表单提交事件处理器
```

```
$('#myFormId').submit(function() {
    // 提交表单
```

```
$(this).ajaxSubmit();
// 为了防止普通浏览器进行表单提交和产生页面导航（防止页面刷新？）返回 false
return false;
});
```

formSerialize ()

将表单串行化（或序列化）成一个查询字符串。这个方法将返回以下格式的字符串：
name1=value1&name2=value2。

可链接（Chainable）：不能，这个方法返回一个字符串。

实例：

```
var queryString = $('#myFormId').formSerialize();
```

```
// 现在可以使用$.get、$.post、$.ajax 等来提交数据
```

```
$.post('myscript.php', queryString);
```

fieldSerialize ()

将表单的字段元素串行化（或序列化）成一个查询字符串。当只有部分表单字段需要进行串行化（或序列化）时，这个就方便了。这个方法将返回以下格式的字符串：
name1=value1&name2=value2。

可链接（Chainable）：不能，这个方法返回一个字符串。

实例：

```
var queryString = $('#myFormId .specialFields').fieldSerialize();
```

fieldValue ()

返回匹配插入数组中的表单元素值。从 0.91 版起，该方法将总是以数组的形式返回数据。如果元素值被判定可能无效，则数组为空，否则它将包含一个或多于一个的元素值。

可链接（Chainable）：不能，该方法返回数组。

实例：

```
// 取得密码输入值
```

```
var value = $('#myFormId :password').fieldValue();
```

```
alert('The password is: ' + value[0]);
```

resetForm ()

通过调用表单元素原有的 DOM 方法，将表单恢复到初始状态。

可链接（Chainable）：可以。

实例：

```
$('#myFormId').resetForm();
```

clearForm ()

清除表单元素。该方法将所有的文本（text）输入字段、密码（password）输入字段和文本区域（textarea）字段置空，清除任何 select 元素中的选定，以及将所有的单选（radio）按钮和多选（checkbox）按钮重置为非选定状态。

可链接（Chainable）：可以。

```
$('#myFormId').clearForm();
```

clearFields ()

清除字段元素。只有部分表单元素需要清除时才方便使用。

可链接 (Chainable): 可以。

`$('#myFormId .specialFields').clearFields();`

Options 对象

ajaxForm 和 ajaxSubmit 都支持众多的选项参数, 这些选项参数可以使用一个 Options 对象来提供。Options 只是一个 JavaScript 对象, 它包含了如下一些属性与值的集合:

target

指明页面中由服务器响应进行更新的元素。元素的值可能被指定为一个 jQuery 选择器字符串, 一个 jQuery 对象, 或者一个 DOM 元素。

默认值: null。

url

指定提交表单数据的 URL。

默认值: 表单的 action 属性值

type

指定提交表单数据的方法 (method): “GET”或“POST”。

默认值: 表单的 method 属性值 (如果没有找到默认为“GET”)。

beforeSubmit

表单提交前被调用的回调函数。“beforeSubmit”回调函数作为一个钩子 (hook), 被提供来运行预提交逻辑或者校验表单数据。如果“beforeSubmit”回调函数返回 false, 那么表单将不被提交。“beforeSubmit”回调函数带三个调用参数: 数组形式的表单数据, jQuery 表单对象, 以及传入 ajaxForm/ajaxSubmit 中的 Options 对象。表单数组接受以下方式的数据:

`[{ name: 'username', value: 'jresig' }, { name: 'password', value: 'secret' }]`

默认值: null

success

表单成功提交后调用的回调函数。如果提供“success”回调函数, 当从服务器返回响应后它被调用。然后由 dataType 选项值决定传回.responseText 还是 responseXML 的值。

默认值: null

dataType

期望返回的数据类型。null、“xml”、“script”或者“json”其中之一。dataType 提供一种方法, 它规定了怎样处理服务器的响应。这个被直接地反映到 jQuery.httpData 方法中去。下面的值被支持:

'xml': 如果 dataType == 'xml', 将把服务器响应作为 XML 来对待。同时, 如果“success”回调方法被指定, 将传回 responseXML 值。

'json': 如果 dataType == 'json', 服务器响应将被求值, 并传递到“success”回调方法, 如果它被指定的话。

'script': 如果 dataType == 'script', 服务器响应将求值成纯文本。

默认值: null (服务器返回.responseText 值)

semantic

Boolean flag indicating whether data must be submitted in strict semantic order (slower). Note that the normal form serialization is done in semantic order with the exception of input elements of type="image". You should only set the semantic option to true if your server has strict semantic

requirements and your form contains an input element of type="image".

布尔标志，表示数据是否必须严格按照语义顺序（sloper?）来进行提交。注意：一般来说，表单已经按照语义顺序来进行了串行化（或序列化），除了 type="image" 的 input 元素。如果你的服务器有严格的语义要求，以及表单中包含有一个 type="image" 的 input 元素，就应该将 semantic 设置为 true。（译注：这一段由于无法理解，翻译出来可能语不达意，但请达人指正。）

默认值：false

resetForm

布尔标志，表示如果表单提交成功是否进行重置。

Default value: null

clearForm

布尔标志，表示如果表单提交成功是否清除表单数据。

默认值：null

实例：

// 准备好 Options 对象

```
var options = {
    target:  '#divToUpdate',
    url:      'comment.php',
    success: function() {
        alert('Thanks for your comment!');
    }
};
```

// 将 options 传给 ajaxForm

```
$('#myForm').ajaxForm(options);
```

注意：Options 对象还可以用来将值传递给 jQuery 的 \$.ajax 方法。如果你熟悉 \$.ajax 所支持的 options，你可以利用它们来将 Options 对象传递给 ajaxForm 和 ajaxSubmit。

ajaxForm() 适用于以表单提交方式处理 ajax 技术（需要提供表单的 action、id、method，最好在表单中提供 submit 按钮）它大大简化了使用 ajax 技术提交表单时的数据传递问题，使用 ajaxForm() 你不需要逐个的以 JavaScript 的方式获取每个表单属性的值，并且也不需要请求路径后面通过 url 重写的方式传递数据。ajaxForm() 会自动收集当前表单中每个属性的值，然后将其以表单提交的方式提交到目标 url。这种方式提交数据较安全，并且使用起来更简单，不必写过冗余的 JavaScript 代码

```
$(document).ready(function(){
    registerForm'表单id
    data回调数据
    $('#registerForm').ajaxForm(function(data){
        alert(data); //弹出ajax请求后的回调结果
    });
});
```

ajaxSubmit() 适用于以事件的机制以 ajax 提交 form 表单（超链接、图片的 click 事件），

该方法作用与 `ajaxForm()` 类似，但它更为灵活，因为他依赖于事件机制，只要有事件存在就能使用该方法。你只需指定该 form 的 `action` 属性即可，不需要提供 `submit` 按钮。

```
$(document).ready(function(){
    $('#btn').click(function(){
        $('#registerForm').ajaxSubmit(function(data){
            alert(data);
        });
        return false;
    });
});
```

该段代码作用是在表单中 id 为 `btn` 的按钮 `click` 事件触发时通过 `ajaxSubmit()` 方法以 `ajax` 技术提交表单到表单的 `action` 所指路径

formSerialize() 是将一个 form 中所有的表单元素以 `name` 作为 key，`value` 作为值进行序列化操作，这就需要你必须为每一个表单元素设置表单元素 `name` 属性及填充表单元素 `value` 的值，最好也设置 `id` 方便 `jquery` 定位表单元素。若要使用此方法你必须设置表单元素 `name` 属性及填充表单元素 `value` 的值，我在初次使用时就忘了设置 `name` 属性，最后在同事的帮助下找了好久才发现此错误。

```
var str=$('#registerForm').formSerialize(); // registerForm为form id
alert(str);
```

fieldSerialize() 是将 form 中表单元素进行序列化以 `name` 作为 key，`value` 作为值进行序列化操作，这就需要你必须为每一个表单元素设置表单元素 `name` 属性及填充表单元素 `value` 的值。

```
var str=$('#username').fieldSerialize();
alert(str);
```

其他方法不做总结了，用的话大家在 `google` 上 `baidu` 一下相信会有很多令你满意的答案的。