

jQuery and AJAX

Kayartaya Vinod

\$.ajax – jQuery's Ajax support

- **\$.ajax(optionsObject)**
 - Minimal form: `$.ajax({url: "address", success: funct});`
 - Don't forget the ".". It is `$.ajax(...)`, not `$ajax(...)`.
 - The handler function gets the response text, not the response object. jQuery guesses if it should be plain text, XML, or JSON from the response type. If you want to enforce that handler gets given type, use `dataType` option
- **Options for \$.ajax({...})**
 - Almost-always used
 - `url`, `success`
 - Other common options
 - `cache`, `data`, `dataType`, `error`, `type`, `username`, `password`

\$.ajax – jQuery's Ajax support

- **Simplest form**

- function someHandler(text) { ... }
- Note that it gets the response *text*, not the response object
- And, “text” can really be XML or JSON, depending on the `dataType` option

- **Full form**

- function someHandler(text, status, request) { ... }
- **text**
 - Response data from server
- **status**
 - String describing the status: "success" or "notmodified". Rarely useful. In error handlers, the status string is more meaningful.
- **request**
 - The raw XMLHttpRequest object.

Common options

- **Options (almost) always used: url, success**

- \$.ajax({url: "some-address", success: someFunction});
 - success is not strictly required; you might want to just fire off some data to the server and not display anything

- **Common options: example**

```
$.ajax({  
  url: "address",  
  success: successHandlerFunction,  
  data: { param1: "foo bar", param2: "baz"},  
  error: errorHandlerFunction,  
  cache: false,  
  dataType: "json",  
  username: "resig",  
  password: "scriptaculous-fan" });
```

Options - details

Name	Description	Default
async	Should the request be asynchronous? Use synchronous requests with caution since they lock up the browser.	true
beforeSend	Function to modify XMLHttpRequest object before it is sent (e.g., to set custom headers). The XHR is automatically passed to function.	None
cache	Is browser permitted to cache the page? Set to false if you use GET and you could get different responses back from the same data. Equivalent to having the server send Cache-Control: no-cache and Pragma: no-cache.	true (except false if dataType is script or json)
complete	Function to be called after error or success function is finished.	None
contentType	Content-Type of data sent to server. Rarely needed.	application/x-www-form-urlencoded
data	Data to send to server (possibly after conversion). Sent in the appropriate place depending on whether it is GET or POST. Can be a String or an object. If a String, sent unchanged. If an object, property names become param names and property values get URL-encoded and become param values. & and = inserted automatically. If a value is an array, it is serialized with repeated param names.	Empty

Options - details

Name	Description	Default
dataFilter	Response-data sanitizing function. Rarely used.	<i>None</i>
dataType	The format in which to pass the response to the handler function. Legal values are text, html (same as text except embedded scripts are run), xml, json, jsonp (JSON with Padding), and script (evaluates the response as JavaScript and returns it as plain text).	html , xml, or json <i>(makes intelligent guess)</i>
error	Function to be called if request fails. Function is passed 3 args: the XHR object, a string describing the error type, and an optional exception object. Possible values for the second argument are null, "timeout", "error", "notmodified" and "parsererror".	<i>None</i>
global	jQuery lets you set global defaults for various handlers: should they be used if set?	true
ifModified	Should the request be considered successful only if the response has changed since the last request (based on the Last-Modified header)?	false
jsonp	Override the callback function name in a jsonp request. JSONP is a JSON extension in which the name of a callback function is specified as an input argument of the call itself.	callback
password username	Username and password to be used in response to HTTP authentication request.	<i>None</i>

Options - details

Name	Description	Default
processData	Should the value of the "data" property, if an object, be turned into a URL-encoded query string?	true
scriptCharset	Forces the request to be interpreted as a certain charset. Only for requests with dataType of "jsonp" or "script" and type of "GET".	None
success	Function to be called if request succeeds. Function passed 3 args: the data returned from the server (formatted according to the dataType property), a string describing the status, and the XHR.	None
timeout	Timeout in milliseconds. If request takes longer, the error handler will be called instead of the success handler.	Global timeout, if set via \$.ajaxSetup
traditional	Should data with arrays be serialized in traditional manner (shallow), or recursively (deep).	false
type	The HTTP method to use for the request. "get" and "post" are main options, but some browsers support other methods.	get
url	The address to request. Should be a relative URL.	None
xhr	Callback for creating your own custom XMLHttpRequest object.	ActiveXObject if available (IE), XMLHttpRequest otherwise

Shortcuts for \$.ajax

- **\$.get**
 - \$.get("url", dataObj, someFuncnt)
 - \$.ajax({url: "url", data: dataObj, success: someFuncnt});
- **\$.post**
 - \$.post("url", dataObj, someFuncnt)
 - \$.ajax({url: "url", data: dataObj, success: someFuncnt, type: "post"});
- **\$.getJSON**
 - \$.getJSON("url", dataObj, someFuncnt)
 - \$.ajax({url: "url", data: dataObj, success: someFuncnt, dataType: "json"});
- **Note**
 - get and post take the type as an optional fourth argument