

Session 18

jQuery - Ajax

1

Reference

■ Tutorials

<http://learn.jquery.com/ajax/>

http://www.w3schools.com/jquery/jquery_ajax_intro.asp

■ jQuery Methods

http://www.w3schools.com/jquery/jquery_ref_ajax.asp

© Robert Kelly, 2012-2018

2

AJAX Recap

- Asynchronous JavaScript Technology and XML
- Allows incremental update of Web pages within the browser
- Not dependent on any given language or data exchange format, but works well with JavaScript
- Uses the XMLHttpRequest object implementations

The XML part of the name is no longer important in Ajax

© Robert Kelly, 2012-2018

3

XMLHttpRequest Object

- Transport object for communication between client and server
- Methods allow
 - Specify request details
 - Extract response data
- Subject to some cross-domain limitations

© Robert Kelly, 2012-2018

4

Common XMLHttpRequest Properties

- `onreadystatechange` - Event handler (function) for an event that fires at every state change
- `readyState` - Object status integer:
 - 0 = uninitialized
 - 1 = loading
 - 2 = loaded
 - 3 = interactive
 - 4 = complete
- `responseText` - String version of data returned from server process
- `responseXML` - DOM-compatible document object of data returned from server process
- `status` - Numeric code returned by server (e.g., 404)
- `statusText` - String message accompanying the status code

Depends on the
content-type of your
response

© Robert Kelly, 2012-2018

5

Common XMLHttpRequest Methods

- `open("method", "URL"[, asyncFlag])` - Initializes the request parameters (destination URL, method, and asynchronous flag)
- `send(content)` - Transmits the request, optionally with postable string or DOM object data
- `setRequestHeader("label", "value")` - Assigns a label/value pair to the header to be sent with a request
- `abort()` - Stops the current request
- `getAllResponseHeaders()` - Returns complete set of headers (labels and values) as a string
- `getResponseHeader("headerLabel")` - Returns the string value of a single header label

Interaction using XMLHttpRequest is
very low-level and complicated

© Robert Kelly, 2012-2018

6

Ajax - jQuery Load

■ jQuery allows easy, browser-consistent use of Ajax

■ Example

```
$('#result').load('test.html');
```

Load uses http
GET method

- Fetches data from a server
- Sets the html contents of the matched element to the return data
- Optional parameters for 1) data sent to server and 2) callback function

Note that the call is asynchronous, and in this case there is an implicit callback function that inserts the result in the selected elements

© Robert Kelly, 2012-2018

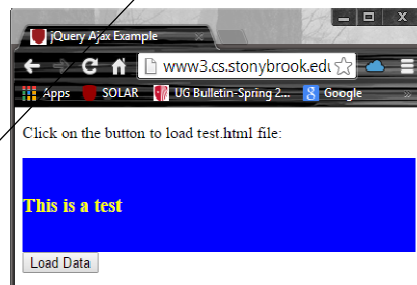
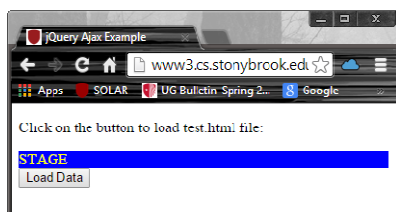
7

Example ...

■ Example inserts the html below (test.html) into the page

```
<br /><h3>This is a test</h3><br />
```

Access the page and try
the example (and look at
test.html)



Example at:

<http://www.cs.stonybrook.edu/~cse336/JQuery-LoadExample.html>

© Robert Kelly, 2012-2018

8

... Example

```
<script>
  $(document).ready(function() {
    $("#driver").click(function(event) {
      $("#stage").load("test.html");
    });
  });
</script>
</head>
<body>
<p id="p1">Click on the button to load test.html file:</p>
<div id="stage" style="background-color:blue;">STAGE</div>
<input type="button" id="driver" value="Load Data" />
</body>
```

test.html contains
`
<h3>This is a test</h3>
`

© Robert Kelly, 2012-2018

9

jQuery load() Method

- Loads data from a server and puts the returned data into the selected element
- Syntax: `$(selector).load(URL, [data], [callback]);`
 - URL - the URL you wish to load (required)
 - data - a set of query string key/value pairs to send with the request (optional)
 - callback - name of a function to be executed after the load() method is completed (optional)

You may not need the *callback* function if you are just loading an error message into a component

Some filtering of the returned data might be done (e.g., remove <html> tag

© Robert Kelly, 2012-2018

10

Callback Function

- The optional *callback* parameter to the *load* function specifies a callback function to run when the *load()* method is completed
- **Callback function parameters:**
 - *responseText* - contains the resulting content if the call succeeds
 - *statusText* - contains the status of the call
 - *xhr* - contains the XMLHttpRequest object

© Robert Kelly, 2012-2018

11

jQuery get() Method

- The jQuery *load* method is just a shortcut when you need to insert the response into an html element
- Use the *get* or *post* methods when you need more control
- **Example - compare with previous code in red**

```
$(document).ready(function() {  
  $("#driver").click(function(event) {  
    $.get("test.html", function(data, status, jqXHR){  
      document.getElementById("stage").innerHTML=jqXHR.responseText;  
    }); }); });
```

XMLHttpRequest
object

Data (if any) sent to
server

```
$(document).ready(function() {  
  $("#driver").click(function(event) {  
    $("#stage").load("test.html");  
  }); });
```

© Robert Kelly, 2012-2018

12

Let's Try That Example Again

```
<script>
  $(document).ready(function() {
    $("#driver").click(function(event) {
      $("#stage").load("http://www.stonybrook.edu/");
    });
  });
</script>
</head>
<body>
<p id="p1">Click on the button to load the html file:</p>
<div id="stage" style="background-color:blue;">STAGE</div>
<input type="button" id="driver" value="Load Data" />
</body>
```

The load failed

© Robert Kelly, 2012-2018

13

Same Origin Policy

- The same origin policy is an important security concept for browser-side programming languages, such as JavaScript
- The policy
 - permits scripts running on pages originating from the same site to access each other's methods and properties with no specific restrictions
 - prevents access to most methods and properties across pages on different sites

Be sure to set the http header in your server to instruct the container/protocol to ignore the policy

Reference: http://en.wikipedia.org/wiki/Same_origin_policy

© Robert Kelly, 2012-2018

14

Sample jQuery Ajax Functions

Method	Description
<code>\$.get(...)</code>	Loads data from a server using an AJAX HTTP GET request
<code>\$.post(...)</code>	Loads data from a server using an AJAX HTTP POST request
<code>serialize()</code>	Encodes a set of form elements as a string for submission (method of the jQuery form object)
<code>\$.getJSON()</code>	Loads JSON-encoded data from a server using an AJAX HTTP GET request

Note the approach to JSON responses

get and post functions include an optional data parameter (2nd parameter) for data to be sent to the server