JQUERY - AJAX

http://www.tutorialspoint.com/jquery/jquery-ajax.htm

Copyright © tutorialspoint.com

AJAX is an acronym standing for Asynchronous JavaScript and XML and this technology help us to load data from the server without a browser page refresh.

If you are new with AJAX, I would recommend you go through our Ajax Tutorial before proceeding further.

JQuery is a great tool which provides a rich set of AJAX methods to develop next generation web application.

Loading simple data

This is very easy to load any static or dynamic data using JQuery AJAX. JQuery provides **load** method to do the job −

Syntax

Here is the simple syntax for **load** method −

```
[selector].load( URL, [data], [callback] );
```

Here is the description of all the parameters −

- **URL −** The URL of the server-side resource to which the request is sent. It could be a CGI, ASP, JSP, or PHP script which generates data dynamically or out of a database.

- **data −** This optional parameter represents an object whose properties are serialized into properly encoded parameters to be passed to the request. If specified, the request is made using the **POST** method. If omitted, the **GET** method is used.

- **callback −** A callback function invoked after the response data has been loaded into the elements of the matched set. The first parameter passed to this function is the response text recieved from the server and second parameter is the status code.

Example

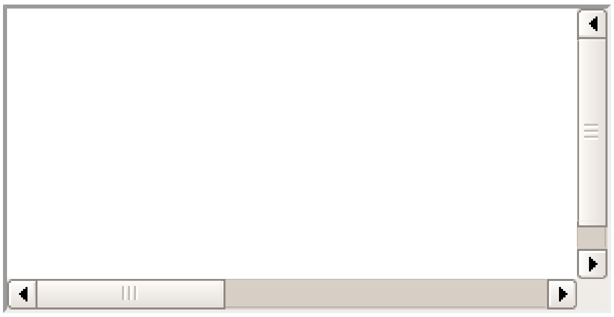
Consider the following HTML file with a small JQuery coding −

```
<html>
   <head>
      <title>The jQuery Example</title>
      <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>

      <script type="text/javascript" language="javascript">
         $(document).ready(function() {
            $("#driver").click(function(event){
               $('#stage').load('/jquery/result.html');
            });
         });
      </script>

   </head>

   <body>

      <p>Click on the button to load /jquery/result.html file −</p>

      <div >
         STAGE
      </div>
      <input type="button"  />

   </body>

</html>
```

Here **load** initiates an Ajax request to the specified URL **/jquery/result.html** file. After loading this file, all the content would be populated inside <div> tagged with ID *stage*. Assuming, our /jquery/result.html file has just one HTML line −

```
<h1>THIS IS RESULT...</h1>
```

When you click the given button, then result.html file gets loaded.



Getting JSON data

There would be a situation when server would return JSON string against your request. JQuery utility function **getJSON** parses the returned JSON string and makes the resulting string available to the callback function as first parameter to take further action.

Syntax

Here is the simple syntax for **getJSON** method −

```
[selector].getJSON( URL, [data], [callback] );
```

Here is the description of all the parameters −

- **URL −** The URL of the server-side resource contacted via the GET method.

- **data −** An object whose properties serve as the name/value pairs used to construct a query string to be appended to the URL, or a preformatted and encoded query string.

- **callback −** A function invoked when the request completes. The data value resulting from digesting the response body as a JSON string is passed as the first parameter to this callback, and the status as the second.

Example

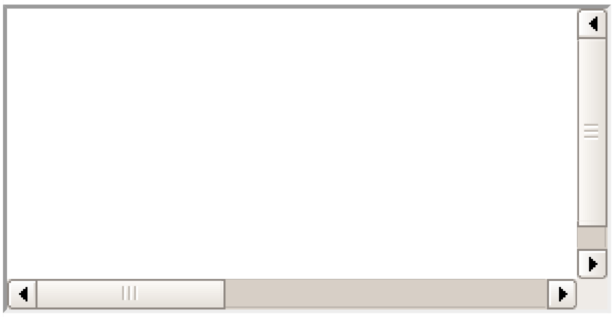
Consider the following HTML file with a small JQuery coding −

```
<html>
   <head>
      <title>The jQuery Example</title>
      <script type="text/javascript" 
src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
		
      <script type="text/javascript" language="javascript">
         $(document).ready(function() {
            $("#driver").click(function(event){
               $.getJSON('/jquery/result.json', function(jd) {
                  $('#stage').html('<p> Name: ' + jd.name + '</p>');
                  $('#stage').append('<p>Age : ' + jd.age+ '</p>');
                  $('#stage').append('<p> Sex: ' + jd.sex+ '</p>');
               });
            });
         });
      </script>

   </head>

   <body>

      <p>Click on the button to load result.json file –</p>

      <div >
         STAGE
      </div>

      <input type="button"  />

   </body>

</html>
```

Here JQuery utility method **getJSON** initiates an Ajax request to the specified URL **result.json** file. After loading this file, all the content would be passed to the callback function which finally would be populated inside <div> tagged with ID *stage*. Assuming, our result.json file has following json formatted content –

```
{
"name": "Zara Ali",
"age" : "67",
"sex": "female"
}
```

When you click the given button, then result.json file gets loaded.



Passing data to the Server

Many times you collect input from the user and you pass that input to the server for further processing. JQuery AJAX made it easy enough to pass collected data to the server using **data** parameter of any available Ajax method.

Example

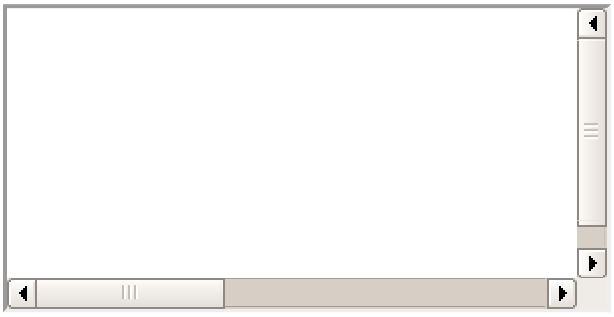
This example demonstrate how can pass user input to a web server script which would send the same result back and we would print it –

```
<html>
   <head>
      <title>The jQuery Example</title>
      <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>

      <script type="text/javascript" language="javascript">
         $(document).ready(function() {
            $("#driver").click(function(event){
               var name = $("#name").val();
               $("#stage").load('/jquery/result.php', {"name":name} );
            });
         });
      </script>

   </head>

   <body>

      <p>Enter your name and click on the button:</p>
      <input type="input"  /><br />

      <div >
         STAGE
      </div>

      <input type="button"  />

   </body>

</html>
```

Here is the code written in **result.php** script −

```
<?php
if( $_REQUEST["name"] )
{
   $name = $_REQUEST['name'];
   echo "Welcome ". $name;
}
?>
```

Now you can enter any text in the given input box and then click "Show Result" button to see what you have entered in the input box.



JQuery AJAX Methods

You have seen basic concept of AJAX using JQuery. Following table lists down all important JQuery AJAX methods which you can use based your programming need −

S.No.	Methods & Description
1	jQuery.ajax*options* Load a remote page using an HTTP request.
2	jQuery.ajaxSetup*options* Setup global settings for AJAX requests.
3	jQuery.get*url, [data], [callback], [type]* Load a remote page using an HTTP GET request.

4	jQuery.getJSON*url, [data], [callback]* Load JSON data using an HTTP GET request.
5	jQuery.getScript*url, [callback]* Loads and executes a JavaScript file using an HTTP GET request.
6	jQuery.post*url, [data], [callback], [type]* Load a remote page using an HTTP POST request.
7	load*url, [data], [callback]* Load HTML from a remote file and inject it into the DOM.
8	serialize Serializes a set of input elements into a string of data.
9	serializeArray Serializes all forms and form elements like the .serialize method but returns a JSON data structure for you to work with.

JQuery AJAX Events

You can call various JQuery methods during the life cycle of AJAX call progress. Based on different events/stages following methods are available −

You can go through all the AJAX Events.

S.No.	Methods & Description
1	ajaxComplete*callback* Attach a function to be executed whenever an AJAX request completes.
2	ajaxStart*callback* Attach a function to be executed whenever an AJAX request begins and there is none already active.
3	ajaxError*callback* Attach a function to be executed whenever an AJAX request fails.
4	ajaxSend*callback* Attach a function to be executed before an AJAX request is sent.
5	ajaxStop*callback* Attach a function to be executed whenever all AJAX requests have ended.

6 ajaxSuccess*callback*

Attach a function to be executed whenever an AJAX request completes successfully.

Loading [MathJax]/jax/output/HTML-CSS/jax.js